DECEMBER 1985 LIDS-R-1519

PROCEEDINGS OF THE

8TH MIT/ONR WORKSHOP ON C^3 SYSTEMS

HELD AT

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS
JUNE 24 TO 28, 1985

EDITED BY

MICHAEL ATHANS
ALEXANDER H. LEVIS

SPONSORED BY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LABORATORY FOR INFORMATION AND DECISION SYSTEMS
CAMBRIDGE, MASSACHUSETTS

WITH SUPPORT FROM

OFFICE OF NAVAL RESEARCH
CONTRACT ONR/N00014-77-C-0532(NRO41-519)

AND IN COOPERATION WITH
IEEE CONTROL SYSTEMS SOCIETY
TECHNICAL COMMITTEE ON C^3

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* 8TH MIT/ONR WORKSHOP ON C^3 SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED INTERIM 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Editors Michael Athans Alexander H. Levis		8. CONTRACT OR GRANT NUMBER(s) ONR-N00014-77-C-0532
9. PERFORMING ORGANIZATION NAME AND ADDRESS Laboratory for Information and Decision Systems Massachusetts Institute of Technology Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-041-519
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE December 1985 13. NUMBER OF PAGES x + 213
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release: Distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

COMMAND, CONTROL, COMMUNICATIONS

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report contains printed manuscripts of papers presented at the Workshop by several authors

DD FORM 1473

FOREWORD

The Eighth MIT/ONR Workshop on C^3 Systems was held from June 24 to 28, 1985 at the Massachusetts Institute of Technology in Cambridge, Massachusetts. These Proceedings constitute the written record of the research presented in the regular sessions held during the first four days. The fifth day's classified session, sponsored by the Office of Naval Research, was held at the MITRE Corporation in Bedford, Massachusetts; the subject matter was C^3I Battle Management associated with the Space Defense Initiative (SDI).

Attendance at the workshop this year was lower than in the recent past. A change in the schedule of the MORS meeting brought the two conferences in conflict, with the result that a number of persons usually active in both meetings had to make a choice. The total attendance was close to 100 persons including students, primarily from MIT. One new feature this year was the introduction of catered lunches for the workshop attendees. This gave additional opportunity for close interaction between speakers and attendees and led to useful discussions. The program had fewer parallel sessions and more time for discussion after each paper and at plenary sessions. Consequently, the atmosphere that prevailed was much more that of a workshop, as in the early years, and less of a conference, as in the recent past. We intend to organize common lunches in future workshops and try to provide adequate time for discussions.

The number of papers in this volume is smaller than in the past. There are three reasons for that. (1) The lower attendance at the workshop this year. (2) A number of authors indicated some difficulty in obtaining the appropriate approvals for submitting a written version of their presentation. (3) There were quite a few briefings by representatives of 6.2 programs that could not be translated easily in the format of a technical paper.

There was a general discussion at the last plenary session about the future and direction of the workshop. The consensus was that the workshop has fulfilled its original mission quite successfully: there is a substantial number of researchers in academic institutions, defense laboratories, and in industry who are addressing issues that can be identified clearly as "C^3 problems". The workshop has been the first forum for the presentation of such research. Eight years later, a substantial portion of the results is being presented at regular meetings of the major professional societies and in archival journals, although in a highly distributed form. So, with aspects of C^3 research becoming part of the mainstream research in several disciplines, is there a unique role for the workshop in the future? The lively comments of the participants indicated that the workshop should be continued - such a forum is still needed. To reflect the broad participation by many research programs in universities and the Department of Defense, it has been decided that the title of the workshop be changed to "The Nth Workshop on C^3 Systems", i.e., to omit the MIT/ONR designation. However, for the foreseeable future, the workshop will continue to be organized by MIT and be sponsored by the Office of Naval Research.

Therefore, we are pleased to announce that the Ninth Workshop on C^3 Systems will be held on June 2 to June 4 at the Naval Postgraduate School and at the Hilton Resort Inn in Monterey, California. The duration of the workshop has been reduced from five to four days; the classified sessions, to be organized by ONR, will be for half a day. The first announcement and call for papers was distributed in November to give ample time to prospective authors to secure the appropriate approvals for their presentations.

The editors sincerely thank the authors and the participants for their contributions to the 1985 Workshop. Special thanks are due to Mr. J. Randolph Simpson of ONR; to Ms. Lisa M. Babine of MIT for her superior handling of the administrative aspects of the Workshop and these Proceedings; and to Ms. Fifa Monserate for assisting in the smooth operation of the workshop at MIT.

Michael Athans
Alexander H. Levis

Cambridge, Massachusetts
December 1985

TABLE OF CONTENTS

		Page
FOREWORD		v
1.	PROBABILISTIC LOGIC MODELS OF MILITARY CONFLICT AND C^3I DEVELOPMENT R. W. Anthony, The MITRE Corporation	1
2.	AN OVERVIEW OF HOW EXPERT SYSTEMS MAY BE APPLIED IN NAVAL COMMAND AND CONTROL SYSTEMS R. E. Wright, Ferranti Computer Systems, Ltd.	7
3.	MARKOVIAN MODELING OF CANONICAL C^3 SYSTEMS COMPONENTS I. Rubin, IRI Corporation I. Mayk, US Army CECOM	15
4.	ARTILLERY CONTROL ENVIRONMENT S. C. Chamberlain and V. A. Kaste, U.S. Army Ballistic Research Laboratory	25
5.	ANALYSIS OF THE FIRE SUPPORT TEAM FORCE DEVELOPMENT TESTING AND EXPERIMENTATION II V. A. Kaste and S. C. Chamberlain, U.S. Army Ballistic Research Laboratory	31
6.	ASSESSMENT OF TIMELINESS IN COMMAND AND CONTROL P. H. Cothier, Service Technique des Télécommunications et Equipements Aéronautiques A. H. Levis, LIDS/MIT	39
7.	ASSESSING THE ORGANIZATIONAL RESPONSIBILITY OF HEADQUARTERS UNDER DIFFERING LEVEL OF STRESS M. G. Sovereign and J. S. Stewart, Naval Postgraduate School	49
8.	EFFECTIVENESS ANALYSIS OF EVOLVING SYSTEMS J. G. Karam, ICF, Inc. A. H. Levis, LIDS/MIT	53
9.	AN EXPERIMENTAL PLAN FOR STUDYING DISTRIBUTED TACTICAL DECISIONMAKING D. Serfaty and D. L. Kleinman, University of Connecticut	65
10.	ON THE ANALYSIS AND DESIGN OF HUMAN INFORMATION PROCESSING ORGANIZATIONS K. L. Boettcher and R. R. Tenny, LIDS/MIT	69
11.	DISTRIBUTED DECISIONMAKING WITH CONSTRAINED DECISION MAKERS - A CASE STUDY K. L. Boettcher and R. R. Tenney, LIDS/MIT	75
12.	CALCULATING TIME-RELATED PERFORMANCE MEASURES OF A DISTRIBUTED TACTICAL DECISIONMAKING ORGANIZATION USING STOCHASTIC TIMED PETRI NETS R. P. Wiley and R. R. Tenney, LIDS/MIT	81

13. COMPUTATION OF DELAYS IN ACYCLICAL DISTRIBUTED DECISIONMAKING ORGANIZATIONS 87

V. Y-Y. Jin and A. H. Levis, LIDS/MIT

14. OPTIMAL TASK ALLOCATION FOR A TEAM OF TWO DECISION MAKERS WITH THREE CLASSES OF IMPATIENT TASKS 95

Z-J. Wu, P. B. Luh, S-C. Chang, University of Connecticut
D. A. Castanon, ALPHATECH, Inc.

15. INFORMATION STORAGE AND ACCESS IN DECISIONMAKING ORGANIZATIONS 101

G. J. Bejjani, Morgan Stanley
A. H. Levis, LIDS/MIT

16. RESPONSE DEPICTION FOR AUTOMATED COMBAT SYSTEMS 111

G. E. Frishkorn and J. R. Gersh, The Johns Hopkins University

17. A THEORY TO GUIDE THE DESIGN OF SITUATION ASSESSMENT AIDS FOR DECISION MAKING 115

D. Noble, Engineering Research Associates, Inc.

18. C^3 SYSTEMS MODELLING 121

S. Rosenstark and J. Frank, New Jersey Institute of Technology

19. A MODEL FOR ONE-DIMENSIONAL IDENTIFICATION/COUNTER DYNAMICS 127

A. U. Meyer, New Jersey Institute of Technology
I. Mayk, U.S. Army CECOM

20. A STOCHASTIC MODEL OF LANCHESTER'S EQUATIONS 135

J. Tavantzis, S. Rosenstark and J. Frank, New Jersey Institute of Technology

21. AN ASSOCIATIVE MEMORY PARADIGM TO SOLVE C^3I PROBLEMS 141

H. Szu, S. Gardner, and L. Sweet, Naval Research Laboratory

22. AN ALGORITHM FOR THE SOLUTION OF A CLASS OF DECENTRALIZED STOCHASTIC CONTROL PROBLEMS 145

M. P. Kastner and D. A. Castanon, ALPHATECH, Inc.

23. DECISION PROCESSES FOR LARGE SCALE RESOURCE ALLOCATION PROBLEMS 153

M. D. Diamond, FMC Corporation
O. M. Carducci, Carnegie-Mellon University

24. COMBINATION OF EVIDENCE IN C^3 SYSTEMS 161

I. R. Goodman, Naval Ocean Systems Center

25. ON COMBINING UNCERTAIN MESSAGES USING BELIEF FUNCTIONS 167

A. P. Dempster and A. Kong, Harvard University

26. MAXIMUM LIKELIHOOD DETECTION AND ESTIMATION OF JUMPS IN DYNAMIC SYSTEMS 173

R. C. Morgan, M. S. Asher, D. W. Porter, and W. S. Levine
Business and Technological Systems, Inc.

27. A MARKOV TWO SENSOR CORRELATION PERFORMANCE MODEL 179

M. A. Kovacich, COMPTEK Research, Inc.

28. THE NEED FOR MULTIPLE HYPOTHESIS DATA CORRELATION METHODS IN MULTI-TARGET TRACKING 185

B. Belkin and W. R. Stromquist, Daniel H. Wagner, Associates

29. LIMITED SENSING ALGORITHMS FOR COMMUNICATION NETWORKS USING CARRIER-SENSE MULTIPLE ACCESS CHANNELS 189

L. Merakos and P. Papantoni-Kazakos, University of Connecticut

30. BROADCAST COMMUNICATION POLICIES FOR DISTRIBUTED AEROACOUSTIC TRACKING 195

J. R. Delaney, MIT Lincoln Laboratory
R. R. Tenney, ALPHATECH, Inc.

LIST OF ATTENDEES 201

C^3 WORKSHOP PROGRAM 207

AUTHOR INDEX 213

Probabilistic Logic Models of Military Conflict and C^3I Development

Robert W. Anthony

The MITRE Corporation

ABSTRACT

Probabilistic logic provides a way to represent mathematically what people think are the logical conditions for success in military conflict. This representation can include propositions about doctrine, peacetime posturing, readiness, system development, and political climate as well as numbers of weapons and their effectiveness.

More specifically, I introduce a special type of probabilistic logic model called the Sequential Reliability Model (SRM). It estimates the reliability of one antagonist's planning sequence in light of the countermeasures that their opponent could use. The SRM can be solved graphically as well as analytically. It also admits enough realism to show why many planning function details do not contribute to conflict outcomes; instead, systems principles embodied within the basic planning approach shape outcomes.

INTRODUCTION

Models of military conflict explicitly or implicitly address four major questions. On each question, my approach differs from more traditional ones.

1. How to Represent the Situation?

Traditional approaches usually represent the situation by mechanistic simulation that plays out one scenario at a time. My alternative approach is to assess a multitude of scenarios and contingencies in a holistic manner. This can be done by using probabilistic logic, that is, expressing the logical alternatives for achieving success in probabilistic terms. Probabilistic logic is already the foundation for reliability theory, fault tree analysis, risk analysis, decision theory as well as some techniques used in expert systems.

2. How to Model Military Conflict?

Traditional approaches to modeling military conflict usually emphasize the efficient use of available assets to coerce or destroy an enemy. However, the most important military modeling problem appears to be that of estimating the dependability of plans and capabilities based upon a holistic view of the conflict situation (Anthony, 1984). My approach is to model military conflict as each side attempting to render the other's plans and systems unreliable.

3. How to Include the Enemy's Actions?

Traditionally, the enemy is included by building two models, one for each side, and coupling these models at a single engagement interface. Whenever C^3I processes are included, they only indirectly shape the physical engagement interface. In contrast, my approach logically enumerates the enemy's array of countermeasure opportunities and integrates their likely responses into the model.

4. How to Deal with Complexity?

Traditional military models incorporate sufficient detail to present a plausible picture of what reviewers and users consider important. Most often, this produces quite complex models with confusing webs of assumptions and interdependencies. However, my approach begins with understandable overview models expressed as general logical propositions. Thereafter, a sequence of embedded and progressively more detailed models articulates relevant, distinct logical branches. The probabilities assigned to these branches are combined to estimate the likelihood that the general propositions are true. Each successive level of embedded models in the sequence is a self-contained representation of the entire conflict situation.

One can use any of my four different approaches to traditional military modeling questions independently. If they are combined with four other simplifying assumptions, as explained in the next section, they produce the Sequential Reliability Model (SRM).

Additional sections show examples of likely outcomes of conflict between forces that emphasize different planning principles. The final section discusses the utility of the Sequential Reliability Model.

THE SEQUENTIAL RELIABILITY MODEL

To explain the Sequential Reliability Model (SRM) approach, I will introduce a greatly simplified three-task planning process representing the command and control cycle. This simplified process can be expanded through the first five steps of its SRM to illustrate all of the concepts necessary to attack more complex and realistic situations. Graphical as well as analytical solutions to the SRM show how one can obtain an intuitive understanding of the features that determine conflict outcomes.

Step 1: Figure 1a shows the simplified three-task planning process for RED. Its tasks of "Seeing, Deciding, and Acting," are roughly equivalent to gathering intelligence, laying out options, and taking action. The logical condition for RED to succeed using this planning process is that all three serial tasks be completed successfully.

Step 2: Figure 1b is the reliability diagram representing BLUE's situation. In order to thwart RED, BLUE must cause one or more of RED's planning tasks to fail. This is indicated by the "tilda" above each task probability. The resulting diagram is the "logical dual" of RED's planning process; that is, the conditions for RED's plan to fail in terms of failures among RED's tasks.

Because each task element of each of the diagrams shown in Figures 1a through 1e represent logically distinct

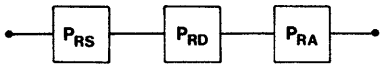


Figure 1a
Step 1: RED Plan, $P_{RP}(P_{RS}, P_{RD}, P_{RA})$

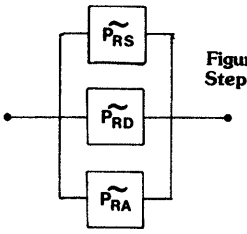


Figure 1b
Step 2: BLUE Situation, $P_{BS}(\widetilde{P_R})$

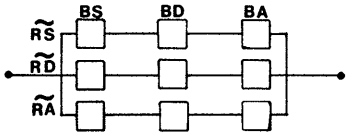


Figure 1c
Step 3: BLUE Countermeasures,
$P_{BCM}(P_B)$

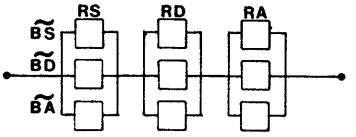


Figure 1d
Step 4: RED Situation,
$P_{RS}(\widetilde{P_B})$

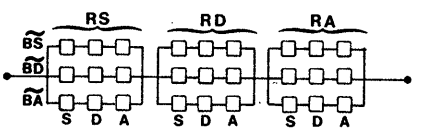


Figure 1e
Step 5: RED Counter-Countermeasures
$P_{RC^2M}(P_R)$

Figure 1
Five Steps of a Sequential Reliability Model

activities, a complete notation would have to distinguish the probability of each element from all others. However, this burdensome detail is unnecessary as will soon be shown. Therefore, the following simplified notation and designation of elements is adopted for these figures.

Consider the lower left hand corner element of the central cluster of nine elements in Figure 1e. As will be described shortly, this element represents RED's counter-countermeasure planning task of "seeing" what action BLUE intends as a countermeasure against RED's orginial decision task. The probability of success for this task element could be designated with three sets of subscripts -- RD, $\widehat{BA}$, and S. The "S" is for the "seeing" task in the counter-countermeasure plan against BLUE's actions. The failure of these actions are designated by the "row" heading "$\widehat{BA}$". Finally, RED's original decision process, the target of BLUE's actions, is represented by the "RD" bracketing the middle nine elements.

All of RED's counter-countermeasure task element probabilities are simply designed by P_R in the caption to Figure 1e. The rationale for this simplification will be given below along with the description of the five figures. Also, note that the subscript "RS" to the system probability for the "RED Situation" is different from the subscript "RS" on task element probabilities standing for "RED Seeing."

Step 3: Figure 1c is the reliability diagram for BLUE's countermeasures to RED's plan. Here, BLUE's planning processes for countering each of RED's three tasks also consist of three similar tasks--seeing, deciding, and acting. (BLUE could have chosen to concentrate on countering only one of RED's tasks, but that would produce a different, simpler example).

Two of the four additional simplifying assumptions necessary to produce the SRM are illustrated by this step. First, all nine of BLUE's countermeasure tasks are assumed to be independent from one another. The rationale is that distinct capabilities, such as intelligence gathering require distinct countermeasures, such as jamming or destroying sensors. If correlations arise, for example, due to air reconnaissance used against both enemy sensors and weapons, these can often be separated into several independent conditional elements. The second assumption is that BLUE's planning processes for each of RED's tasks are logically similar. In the example, each of BLUE's countermeasures to a RED task has a common generic form--a series of three tasks. If planning processes originate in common cultural factors such as political and educational systems, doctrine and policy, technological capability, and nationally accepted military principles, then militray planning processes for seemingly distinct activities could have quite similar forms.

Step 4: Figure 1d is the reliability diagram showing RED's situation taking into account BLUE's countermeasures. This diagram is the logical dual of RED's countermeasures shown in Figure 1c. It completes the first level of the SRM. In this example, the first level could represent a Corps commander's anticipated situation given that he initiated a plan and an opposing Corps commander had reacted to it. An experienced commander would probably consider his counterpart commander's psychology along with enemy doctrine and assets. At this point the original plan could be altered, or if it is deemed acceptable, sent down to Division level for futher articulation.

Note that Figure 1d shows three serial clusters of three parallel tasks. Each serial cluster corresponds to one of RED's original planning tasks shown in Figure 1a. However, in Figure 1d BLUE's countermeasures are embedded within that original structure. This will always be the case since the dual of the dual is itself. Embedding additional processes does not alter the original, or subsequent, task structures. Also, note that the "dual" operation applied to BLUE's serial tasks produced parallel tasks for RED at the counter-countermeasure level.

Step 5: Figure 1e is the first step of the second level of planning. RED uses counter-countermeasures to cause BLUE's counters to fail. In the example, this could correspond to Division level articulation of the Corps plan. Independent C2M plans for air reconnaissance, contingencies, and artillery would be among the articulated logical branches represented by the three-task generic planning process.

Two more simplifying assumptions are illustrated in Figure 1e. These are (1) that the same number of planning steps are taken on all logical branches, and (2) that the likelihoods of success for the lowest level planning task elements are estimated by a single quantity. These assumptions are plausible for two reasons. Wise planning would balance efforts across all tasks--improving the weakest link until all have comparable reliability. This causes the prospects for success of otherwise distinct processes to converge. The

second reason is that only the average task-element probability matters in calculating the overall system probability (Barlow and Porchan, 1975).

Further steps in the SRM would be similar to those already shown, consisting of a sequence of taking the dual and inserting countermeasure plans.

Outcomes of the Conflict

The probability of RED's plan working can now be calculated and plotted in terms of the probability of RED's lowest level planning tasks elements working; see Figure 2. The chance of RED's initial plan working in terms of its three tasks alone is given by the curve designated P_{RP}. Next, the chance of RED's plan working in terms of their counter-countermeasures is the curve designated P_{RC^2M}, where:

$$P_{RC^2M} = (1 - (1 - P_R^3)^3)^3.$$

Successive levels converge toward a vertical threshold. To the right of this threshold, RED wins because their plan works. To the left, BLUE thwarts RED's plan (for BLUE to win, one must consider a BLUE planning initiative).

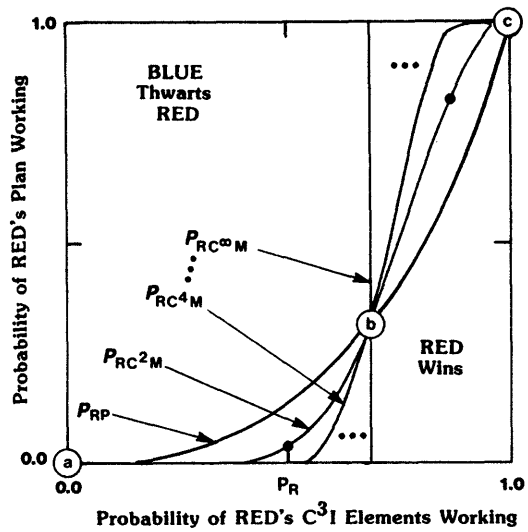


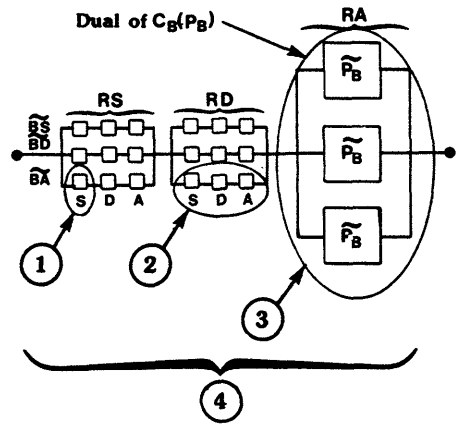
Figure 2
Outcome Curves

Note that all of the successive reliability curves pass through the "fixed point" at "b". This fixed point determines the abscissa of the threshold. There are also two other "fixed points" at "a" and "c". These will be shown to be important for more realistic planning functions.

Derivation of the Outcome Solution

The outcomes for this SRM can be derived graphically as well as analytically. The graphical method provides more insight into the causes and sensitivities of the conflict outcome. Steps of the graphical solution parallel a functional notation that provides a simple analytical solution. These tools generalize to complex situations more easily than do the reliability diagrams of the type shown in Figures 1a through 1e.

Figure 3 explains the graphical derivation of the outcome curve for RED's counter-countermeasures starting with the reliability diagram shown previously in Figure 1e. The explanation begins with the most deeply embedded variable, P_R, and works out to evaluate the overall probability of planning success, P_{RC^2M}. There are four steps going from the inside to the outside of this stack of nested functions. For



(1) **C^2M Planning Element** P_R

(2) **RED's C^2M Plans Foil BLUE CM Planning Element** $C_R(P_R) = \widetilde{P_B}$

(3) **Dual of BLUE's C^3I Functions Represents BLUE's CM Plan Failing to Counter a Step of RED's Plan**

$$C_B^D(C_R(P_R)) = \widetilde{C_B(P_B)} = P_R^1$$

(4) **Chance of RED's Plan Working Due to RED's C^2M Planning Elements**

$$P_{RC^2M}(P_R) = C_R^1(P_R^1)$$
$$= C_R^1(C_B^D(C_R(P_R)))$$

By Definition $C_B^D(\widetilde{P_B}) = 1 - C_B(1 - \widetilde{P_B})$

$$= C_R(1 - C_B(1 - C_R(P_R)))$$

Because $C_R(P_R) = P_R^3$ and $C_B(P_B) = P_B^3$

$$= (1 - (1 - P_R^3)^3)^3$$

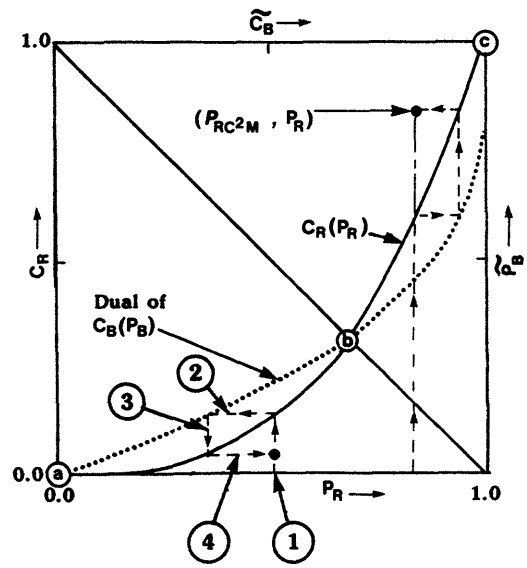


Figure 3
Graphical and Analytical Solution

each step, Figure 3 shows the variables and the functions being evaluated as well as the corresponding graphical interpretation of those functions.

The last two lines of step four show how the nested function notation can be evaluated analytically. Here, the dual is evaluated as the logical complement, and the generic planning functions are then substituted. The result is the same as the direct evaluation of the diagram in Figure 1e given before.

Taken together, all four steps generate one point on the outcome curve. This point is below RED's generic planning curve shown as the solid line. If the initial P_R were greater than the abscissa for point "b", the point would have been above RED's generic planning curve; see the right hand construction on the outcome plot.

Careful examination of the dotted line on the outcome plot shows that it represents the dual of BLUE's generic planning function. In fact, the dotted line is a "reflection" of BLUE's generic planning function about the 45^{o} diagonal shown in Figure 3. Wherever the dotted and solid curves intersect, the cycle of evaluation segments in the graphical solution degenerate to a point. This happens at the three fixed points, "a", "b", and "c". Since these fixed points determine the limiting thresholds and values of SRM, plotting the RED and BLUE generic planning functions in this manner immediately reveals the ultimate conflict outcome. Similar graphical constructions reveal the outcomes for more complex and realistic conflict situations.

OUTCOMES OF CONFLICT

More realistic planning functions used in SRMs generate solutions that mimic some of the less obvious outcomes observed in military conflict. Moreover, the SRM shows why systems principles shape these outcomes.

Consider the effects of complexity, quantity, and quality on generic planning functions. In Figure 4, complexity is pictured as additional serial task clusters that have to work in order for the planning function to work, e.g., five rather than three. Quantity is pictured as multiple backups for executing each task, e.g., nine versus three. Poor quality is pictured as a significant probability of task failure due to factors often outside a commander's control such as weather, terrain, morale, and experience. These uncontrollable factors are represented by series elements inserted into each original task. Unrecognized or uncontrollable correlations across a parallel cluster of backups also arise from such sources as maintenance errors common to all backups, shared logistical constraints, and poor system testing that masks system wide failure modes. Correlations can be represented as a series conditional probability element reducing the quality of the entire planning function.

The plots in the right of Figure 4 show the contrasting generic planning curves for two combatants that adopt different strategies--quality versus quantity. Here it is assumed that investing in more backups sacrifices investment necessary to buy quality and vice versa. (Note that these curves are exaggerated to emphasize planning differences.)

Suppose the RED force took the planning initiative against the BLUE force; that is, the quantity strategy was pitted against the quality strategy. Figure 5 shows the graphical solution and outcome curves. The inverted dual of BLUE's planning function (the dotted line) intersects RED's planning function (the solid line) at fixed points "a", "b" and "c". Again, a steep threshold develops at point "b". The steep counter-countermeasures curve for RED shows that the outcomes are near the limit of convergence already. In a formal mathematical sense, this is due to the steepness of both planning functions at their point of intersection "b". Operationally, this is due to the extensive use of backups by both sides which produces the steep transition in the generic planning functions between rather high and quite low reliabilities. The conflict transition threshold is far to the left for RED because BLUE has many fewer backups. RED's counter-countermeasures would have to be very poor for BLUE to thwart RED's plan.

The fixed point "c" represents a non-trivial conflict outcome and an opportunity for BLUE. Because RED's plans have intrinsic residual chances of failing on their own, even a decimated BLUE force might survive and prevail as RED collapes. For example, the Germans decimated the Russian forces in WWII, but greatly overextended themselves in the process. Weather, distance, and supply shortages eventually led to a collapse of the German offensive.

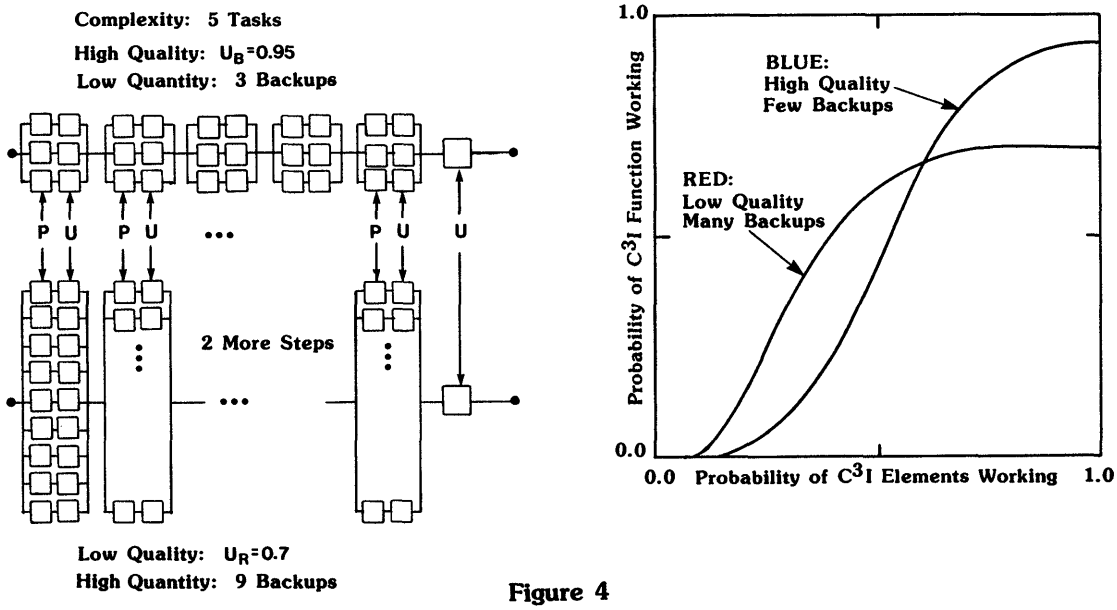


Figure 4
Complexity, Quality, and Quantity

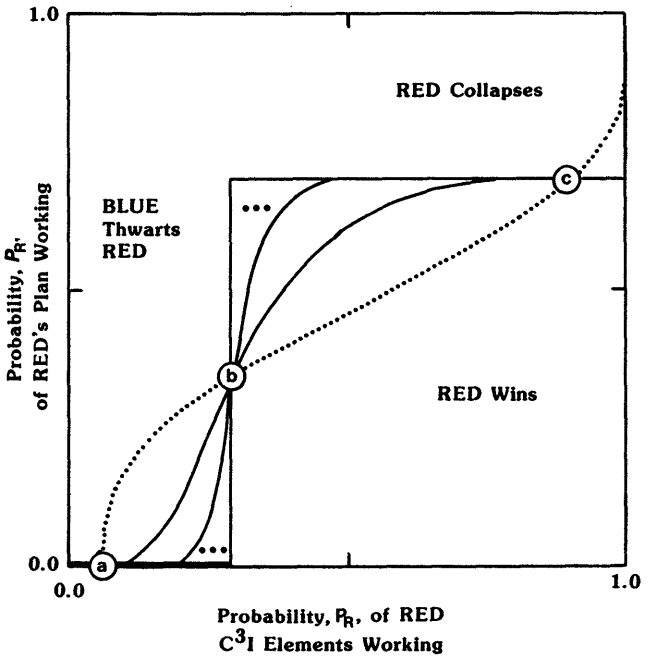


Figure 5
Quality versus Quantity

No matter how likely RED's controllable elements are of working, the "uncontrollable" ones, outside of the immediate influence of either commander, still generate a large chance of collapse. For example, even if RED had a numerically superior force supporting each basic planning element, their prospects of winning could never exceed the cap set by fixed point "c". Thus, the "uncontrollable" parameter in the planning function exerts a leveling effect. This might explain the historical result that across 1,000 battles numerical superiorities as high as 500 percent increased the odds of winning by only 10 percent (Spinney, 1980).

The shape of the generic planning function influences conflict outcomes by determining the location of fixed points and rates convergence. Figure 6 shows how systems principles influence the shape of the planning function. Here, "predictability" refers to a reduction in the uncontrollable factors. Other principles are also illustrated. For example, secrecy is modeled by adding parallel elements not known to the enemy to each step. Secrecy's effect is to bring up the low tail of the planning function where task element reliabilities are poor. All the examples of Figure 4 were "modular and interoperable" in the sense that any means of performing one task could support any of the several ways of performing the other tasks. That is, the planning functions were serial steps consisting of parallel alternatives. This need not be the case. The planning functions could have been three sets of five serial tasks taken in parallel. If so, the system performance could be much worse as shown by the dotted line in Figure 6.

Figure 7 illustrates how these principles provide an explanation for the effects of surprise attack. By choosing the weather, terrain, and state of readiness for the initiation of hostilities, RED improves their "unpredictable factors siutation" while worsening BLUE's. These conditions represent RED surprising BLUE. Consequently, the outcome curves are radically altered because the RED and BLUE planning curves no longer intersect to form fixed points "a" and "b". The zone in which BLUE thwarts RED vanishes in the limit. Only a RED collapse, which is now less likely, can save BLUE.

If one tried to explore these qualitative results using the reliability diagrams or purely analytical methods, one would soon see how much they are obscured by detail. Thus, the graphical solution provides a useful connection from the logically derived planning function, through its underlying systems principles, to the outcome solutions.

THE UTILITY OF SEQUENTIAL RELIABILITY MODELS

The Sequential Reliability Models provide a wide range of tools for exploring military conflict situations.

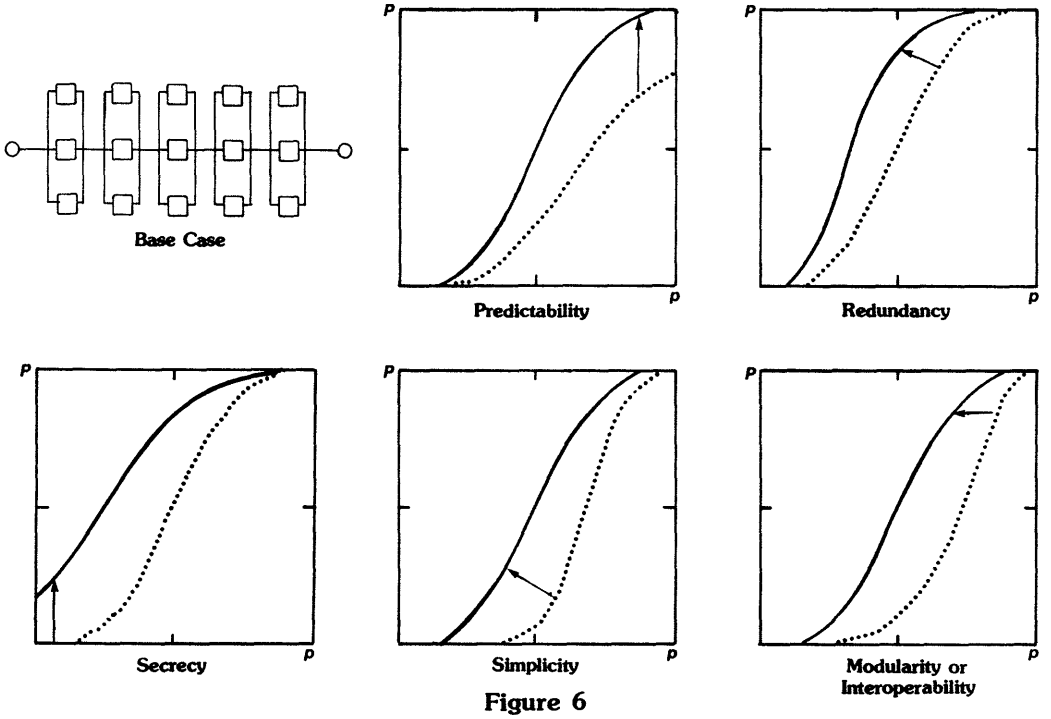


Figure 6
Systems Principles Affecting Planning Functions

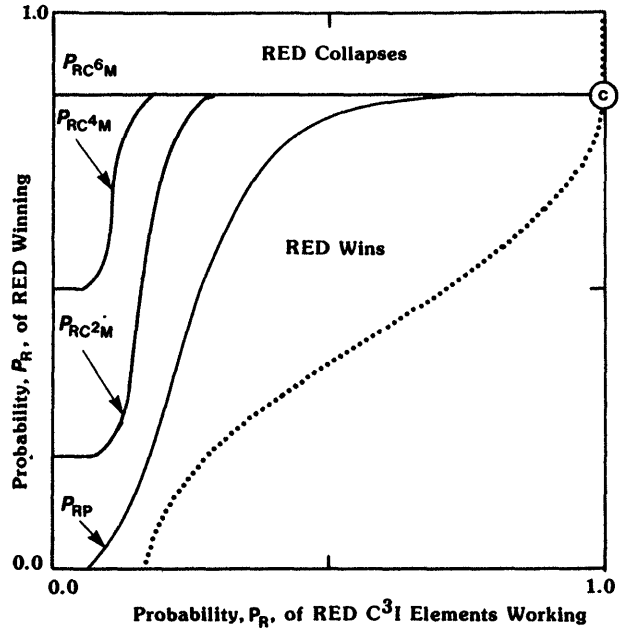


Figure 7
Conflict Outcomes From Surprise

First, SRMs are quite robust in their options for representing conflict situations in terms of probabilistic logic. One can choose a RED or BLUE perspective, an appropriate level of detail, a combination of subjects such as plans, systems, or life-cycles, and a variety of conditions for success. An example of a different condition for success would be that instead of only one element among several parallel elements having to work, two-thirds would have to work. This could model a battlefront in which two of three battalions must win for brigade to win; two of three brigades must win for a division to win; and two divisions must win for corps to win.

Second, SRMs enable one to apply systems principles as tools for exploring alternative outcomes in highly interactive conflict situations. Examples given in the previous section show how system principles such as predictability, redundancy, and simplicity can affect conflict outcomes. Furthermore, the sensitivity of conflict situations to parameters representing the degree of predictability could model military surprise. These examples show how systems principles can be identified as qualitative changes in the generic planning function, and how those changes affect conflict outcomes.

Third, SRMs can support a variety of established analytical methods thereby extending these traditional tools. Because SRMs represent logical conditions for success, they can be formulated to place bounds on plausible outcomes given explicit assumptions. They can explore the sensitivity, thresholds and break points of these assumptions. They can also provide measures of performance for survivability, endurability, supportability, availability, and other "-ilities".

Overall, SRMs enable users to easily build, understand and adjust models of a situation. The reasons for outcomes can be seen in terms of system principles which, in turn, can be extracted from the specifics of the initial logical descriptions. Since the fixed points determine most of the conflict outcome, SRM models naturally suppress non-essential details. Alternative conflict strategies can be explored for points of leverage and sensitivity to things suspected but not known. Moreover, all identified options can be represented simultaneously in the logic of a single model.

REFERENCES

Anthony, Robert W., 1984. "Holistic Patterns in Command, Control, Communications, and Information Systems" Proceedings of the 7th MIT/ONR Workshop on C3 Systems, Laboratory for Information and Decision Systems, MIT.

Barlow, Richard E. and Frank Proschan, 1975. Statistical Theory of Reliability and Life Testing Probability Models, Holt, Rinehart and Winston, Inc.

Spinney, Franklin C., 1980. Defense Facts of Life, Office of the Secretary of Defense (Director, Program Analysis and Evaluation), Pentagon, Washington, D.C.

AN OVERVIEW OF HOW EXPERT SYSTEMS MAY BE APPLIED IN NAVAL COMMAND AND CONTROL SYSTEMS

Ronald E. Wright

Ferranti Computer Systems Limited,
Bracknell Division,
Western Road, Bracknell, Berkshire, England

ABSTRACT

This paper gives some views and experience of the application of A.I. techniques, in particular Expert Systems, to Naval Command and Control systems, and attempts to identify some of the design methodologies, attributes and performance parameters involved.

1. INTRODUCTION

Many areas of human activity require the assessment of a complex situation, the planning of what is to be done about it, and the initiation and control of the subsequent response. Sometimes such activities have a real-time element in that, unless it is possible to make a response in a given time, then it is too late for the response to be effective. Military defence is a human activity of this type and the need to aid the assessment and response process has given rise to the concept of Command and Control systems. Some of these systems are high level, strategic systems designed to enable government authorities to respond on timescales of hours or days; others are low-level, tactical systems used for the timely and effective operation of a local force when responses are often required within minutes or even seconds. One class of Command and Control System is the Action Data Automation (ADA) and Computer Assisted Command Systems (CACS) used in ships of the Royal Navy since the early 1960's. In these the automation provided by the use of digital techniques is largely directed towards establishing a recognised picture of the physical situation from which the combat team can make their assessment of the tactical situation and evoke suitable responses.

The emerging technologies of Artificial Intelligence are very relevant to Command and Control systems. They offer the possibility of developing such systems from their present state to one where there will be considerable automation of the correlation of information and the assessment, planning and control functions. The question is almost not what will happen (autonomous, intelligent weapon systems seem to be inevitable developments) but how it will happen.

2. THE CURRENT STATUS OF NAVAL COMMAND SYSTEMS AND THE NEED FOR AUTOMATED DECISION SUPPORT

The Royal Navy is currently introducing a new range of C-2 systems into its major ships, the CACS series, where CACS stands for Computer Assisted Command System (Ref 1). The main function of these systems is to fuse the data from the ship's own sensors and data received by data links from co-operating ships, aircraft and other sources, to provide the Command Team with a representation of the current tactical situation.

The Command System is one element in the ship's Combat System, shown diagrammatically in Fig. 1. The various systems forming the Combat System comprise a physically distributed system which is linked together by a Combat System Highway or Local Area Network (LAN).

The Command System itself is a distributed system with computing nodes. The number of computing nodes and the distribution of functions is determined in the design of a specific system, a typical example being shown in Fig. 2. Man-Machine Interfaces are provided by display consoles, which have their own processing facilities, and interface to the Command System Highway. The architecture supports the concept of multi-functional displays. The processing nodes are based on two close-coupled processors, which allow a single processor reversionary mode in the event of a failure of either one of the processors in a node. The computers used are Ferranti FM1600E processors which use 24-bit 3-address instructions. The shared stores include bubble memories for essential programs. The FM1600E uses TTL(S) MSI logic and has been the subject of evolutionary development and enhancement over the last twenty years. Data links are buffered from the system by a special node, the Data Link Pre-Processor (DLPP). This is a multi-processor system based on the Ferranti ARGUS M700 micro-processor which has a 16-bit word. The configuration of the DLPP depends on the links being used, but typically would be configured to pre-process NATO Links 11 and 14. The processing capability of the Display Consoles is also provided by a multi-ARGUS M700 system.

The emphasis of the CACS systems is that they are an aid to the ship's Command Team. There is essentially no "intelligence" (in the sense of decision making capability) in the systems. The method used in designing the systems is a top-down approach in which the design of the system is developed in a hierarchical manner, with each successive level of design being of increasing detail, considering hardware, software and operator functions. Experienced operators form part of the design team and, to an extent, their expertise is implicit in the system design. For example in how the operator is lead through pre-planned procedures by a "menu" system of data presentation and associated Alerts and Prompts.

At the Combat System level the trend is increasingly for more processing to take place in the sub-systems (e.g. radars, sonars, etc). This gives an increasing autonomy in the operation of the sub-systems, reduces the traffic demands on the bus, emphasises the role of the Command System and the combat team as being one of combat management.

One important example of this trend is Close In Weapon Systems (CIWS). Such systems provide "last ditch" defence against missiles such as sea skimmers. Typically a CIWS will have to detect a target and engage it in a matter of seconds. A typical flow diagram of the processes involved is given in Fig. 3.

The operator sets up the parameters which determine the status of the system (engagement arcs, alarm ranges, firing rules, etc.) but thereafter the sequence from target detection through to "cease firing" is autonomous. The human operator's task is to supervise the system activities and override automatic decisions if he feels it is necessary to do so. Some systems allow human interventions (for example in track initiation), but this is usually at the expense of system performance. It should be noticed that a CIWS is an autonomous system (as compared with automatic) and involves artificial intelligence in that it simulates the decisions human operators would take in a non-autonomous system. The calculations involved are usually algorithmic, but in some areas (such as track initiation) the calculations imply rules, and hence a potential application of an Expert System approach. However, in general, current CIWS designs adopt a simplistic approach and the number of rules involved in a particular activity is small, so that their implementation does not justify the overhead of an Expert System shell. For example, the threat evaluation in a CIWS is typically something like:

> "if the target is coming towards you and it will arrive before any other target, then it is the major threat."

At the C-2 level the Combat Team are faced with alternatives in responding to a threat to their own ship, and these may not be independent, for example manoeuvring the ship to avoid a missile might put the missile in the blind-fire arc of the CIWS; dispensing chaff might affect the target illuminating radar.

Some mechanisation of the choice of options which has built into it expert experience, would be a valuable aid to the combat team. One technique that might be applicable is the "Influence Diagram", an operational research tool used to model a developing situation to see if factors under control can be manipulated to give a more desirable course of events. The proposed approach (Ref 2) was to define attributes and attribute values which described the situation accurately enough, determine the initial attribute value sets that could be obtained when a missile is first detected, and from these initial sets set up a branching network with branches accounting for all reasonable attribute changes (such as whether the attacking missiles seeker locks on), but excluding attribute changes from own ship's action at this stage.

On every limb of this branching network there will be a set of attribute values defining the state of the situation at that point in its development. Thus every path through the network represents a possible sequence of events, each sequence being defined by a sequence of sets of attribute values (i.e. a sort of chain code defining the system state). These attribute value sets define different event sequences without ambiguity - but unfortunately they are not available to a human or machine decision maker on the spot. He has to make do with the information from his own ship sensors. Because of the limited vision of these sensors there is available only an incomplete definition of the real situation, which will generally indicate only that the situation may be any one of a number of possibilities. Thus for each definitive set of attribute values there is an abbreviated set (those perceived by the sensors) which must be used by the Combat Team. (In this respect the problem is more difficult than designing, say, a chess playing machine, where the situation is fully perceived).

For every path in the branching network the experts can work out a best action sequence, assuming access to full information; and record the perceived attribute value (PAV) at which it should be initiated. The experts also assign to each new action initiation point a figure of merit (FOM) which reflects the estimated penalty to own ship at the end of the old network path if action is not taken.

Where there is a one-to-one relationship between a perceived set and a sequence of actions, or where there is more than one set to an action sequence, there is no problem; but where (because of the poor definition of an abbreviated perceived set) there are several alternative action sequences which could be initiated by one attribute value set, then the alternative new branches are examined to see if perceived sets in their histories can differentiate between them.

The networks described form a knowledge base. When the decision system (Fig. 4) is in operation successive scans of an attribute value are taken from the own ship sensors and data processing outputs as the situation develops. These sets of attribute values are compared with each of the stored action initiation sets in a pattern-matching process. When a "goodness of fit" combined with an initiation set figure of merit exceeds a threshold the corresponding decision is taken.

Such a decision system implies that the risk that a situation will develop which has not been accounted for in the knowledge base, but on the other hand when a situation arises which is described in the knowledge base it will be recognised by the decision system, whereas a human operator may not have recognised the situation.

This conceptual decision system has highlighted two attributes required by decision support systems for own ship defence:

The need to make decisions in the absence of complete information on the tactical situation.

The need of a mechanism to represent and evaluate a changing situation.

Two problems hidden in the above approach are:

How to exercise the expert advisers so that all reasonable scenarios are anticipated (knowledge elicitation).

How to apportion numeric values to subjective parameters, such as figure-of-merit thresholds.

3. THE APPLICATION OF EXPERT SYSTEMS

So far we have considered the CACS systems, autonomous CIWS systems and a possible decision system for ship self-defence. All of these systems imply the use of rules, but not in any formal way. The A.I. community has involved the concept of Expert Systems as a formal methodology for realising rule-based systems with the possibility of using standard

software frameworks and development environments for the design of Expert Systems for specific applications.

In the remainder of this paper we will give examples of the use of Expert Systems in naval applications, and comment on the associated system parameters.

An "Expert System" is built round two basic components - a "Knowledge Base" and an "Inference Engine" (Fig. 5). The Knowledge Base is data base of structured knowledge. Such knowledge can be represented in various ways, but our work has used a Frame and rule-based approach. The "Inference Engine" mechanises methods of plausible reasoning and co-ordinates the interaction of other components of the system. Such other components may include a Man-Machine Interface, for interacting with the user; a DBMS, for holding and accessing background data; an input processor, for inputing new data such as sensory information and possibly rules in developing the Knowledge Base.

There are two basic modes of reasoning used by the Inference Engine : forward chaining, where the reasoning goes from facts to conclusions, and backward chaining where the reasoning goes from an unproved hypothesis backwards in search of facts that would prove it.

The configuration of an Expert System for naval applications is shown in Fig. 6. This was a system developed at the Admiralty Research Establishment (ARE) at Portsdown, England, to demonstrate Expert Systems applied to one of a number of similar planning tasks found in naval Command and Control systems (Refs 3 and 4). Such tasks are characterised by requiring an understanding of the situation and the allocation of resources. The specific task selected for demonstration was EW Tasking as it is a relatively small well-defined activity. The available software only supported backward inferencing, so the design was based on an inference net capable of evaluating a plan already created by the operator. The operator created the plan interactively using the system's colour V.D.U. and the data base, which held details of enemy radars and own force ESM and ECM equipments. This facility in itself proved a great improvement over conventional manual methods. The plan arrived at was then evaluated by the Expert System with the top assertion (goal) being that the proposed tasking has desirable properties. Initially the output gave a certainty value of belief in the top-level assertion and the certainty values of each of the supporting factors. However, this output was of little practical use to a Naval operator and the output was restructured to emphasise the good and bad aspects of the plan. Assertions were combined using Bayes' rule using weightings to represent the importance of the factors as seen by the expert. However, determining the values to be given to these factors involved considerable guesswork and tuning to get reasonable results.

From the point of view of this paper an interesting aspect of this work is the use of rule-based network to represent a plan.

4. FLAG OFFICERS AID AT SEA (FLOATS)

So far we have mainly been concerned with defence of own ship. When ships act in a group the Flag Officer has the task of deploying his available forces in an optimum way. This is an area where the current systems give little help to the Flag Officer, although some aspects of his task are subject to written rules and procedures. We selected the development of a system to assist in the stationing of ships, submarines, helicopters and aircraft (platforms) in a maritime force to provide an optimum defensive screen as being of importance, as it often has to be planned in real-time and involves the assessment of a large quantity and variety of data. We had within our organisation both experienced programmers and ex-service officers with expertise in naval warfare. Usually a command team is divided at some level of responsibility into sub-teams responsible for anti-aircraft, surface, and anti-submarine warfare. Each of these groups will have a different view of the ideal disposition of the force in terms of the threat they perceive and the resources they have in their area of responsibility. So we have a multi-expert system and one of the Flag Officers tasks is to weigh the advice given him by his various advisers. For this project use was made of an existing Expert System Shell "EXPERT 4", which was developed from a program originated at ARE. The initial implementation of FLOATS runs on a VAX11/780. The Expert Shell is written in Pascal and the rest of the package in VAX FORTRAN 77.

The possibilities of stationing a number of ships is infinite and so some strategy is necessary to reduce the task to manageable proportions. The approach we adopted is very much that of the captain of a cricket team or basket ball team. When the team is fielding there are recognised field placings that give a good defensive position. Given these field placings, the Team Captain then allocates his available players taking into account their individual talents, for example, the deep fielders need good running, catching and throwing ability, the close-in fielders need quick reactions.

The positioning of the platforms of a task force depends on the nature of the operation (e.g. defence of a main force, convoy or an amphibious operation). The initial program has addressed the stationing of the platforms of a main body of ships and a defensive screen. The output of the program is to give recommended positions (say as range and bearing from some central reference point) for each platform or defined areas of operation (as would be applicable, for example, to a maritime patrol aircraft).

Data on the scenario is entered by keyboard or, if already archived, recalled from memory. The main body, the central core of the force, consists of the high value units or the ships protected by the remainder. For our initial implementation the ships of the Main Body are given a simple circular disposition.

The screen comprises ships and other platforms whose primary role in the force is to protect others by use of their sensors and weapons. The program allocates the available platforms to either a close screen (which provides short range defence of the main body from air, submarine, or surface attack), an advanced screen (giving defence in depth and freedom to deploy specific capabilities) and a distant screen or picket (deployed at a distance from the main body and in a direction to give the best early warning of an attack).

The system computes the Torpedo Danger Zone (TDZ) around the main body and the arc(s) that include all the individual threat sectors. The platforms allocated to the close screen (usually destroyers and frigates) are stationed on the TDZ limit and spread

uniformly within the threat sectors. The advanced screen (usually comprised of helicopters) is likewise disposed around the limit of the computed Missile Danger Zone (MDZ). The operator is given the opportunity to constrain the solution by imposing special requirements, namely the stationing of a ship as a "goal keeper" over a specified high value unit; the formation on an inner screen between the main body and the TDZ; and the stationing of a specific platform in a specific position. The disposition task uses algorithmetic procedures. The program does not yet position the platforms with advanced warning capabilities (e.g. AEW aircraft) as this computation will need consideration of geographical features and preferably a combined algorithmic and rule based approach. The main goal will be to deploy the appropriate platforms in the direction of the appropriate threat.

The next step is to construct the "ideal" platform for each station, in terms of type of vehicle and its equipment fit (kit). For this an expert system shell is used and a rule base containing some 120 rule clauses. When the ideal platform for each station has been determined the data is passed to a matching programme which determines a figure of merit (FOM) for each platform in each station. This gives a matrix of FOM's of platforms (columns) against stations (rows) and allows a sort to allocate an optimum allocation giving an acceptable total FOM and a close match between each platform and its corresponding ideal. It is then possible to calculate the actual sector each platform is going to control from its sensor performance, and display the result in plan position form to the user. Access is possible to radar and sonar propagation models (e.g. IMP, Ref 5) allowing realistic sensor coverage diagrams to be displayed. The user can also chose to see a report giving the basis of the selection (i.e. description of the "ideal" platform for each platform, ROM matrix, etc). The operator can then work interactively to modify the system's solution (for example by exchanging the platforms between two stations; moving a particular station) and the assessment of these changes can be supported by individual rule-bases. The operator can finally select either his modified solution or the system's original solution. Fig. 7 shows a typical displayed output. This diagram, based on a photograph from an actual working display, shows the operating sectors allocated to a screening force of 6 frigates, 1 destroyer and 3 helicopters, so as to optimally defend a main body of 3 high value units, namely an aircraft carrier, a force guide, and one of the defending ships which has been specifically placed with the main force by the operator. MLA is the Main Line of Approach of the force and "ZZ" is the centre of the position co-ordinates.

FLOATS is currently being further developed to provide the Flag Officer with a range of management aids, and it is intended to incorporate variants of the EW tasking and Flying Programme planning systems being developed at ARE.

Although EXPERT 4 software supports the mathematical representation of certainty, with -5 representing "definitely false", 0 "don't know", and +5 "definitely true", where possible FLOATS avoids the necessity for making judgements. Weightings are used in the calculations of the FOM matrix, where the balance in considering air, surface and sub-service threats is allowed for by giving the threats weightings which are used in the calculation of the FOM's.

Real Time Planning Aids

John McDermott of Carnegie - Mellon University is reported to have identified the following planning functions:

(1) Configuration:

To identify required components and integrate them into an acceptable system.

(2) Scheduling:

To determine and sequence actions that will achieve desired ends.

(3) Design:

To create an object whose components are arranged in a way that satisfies (the most important) placement constraints and whose components have been selected to facilitate placement.

Using these definitions the FLOAT force deployment programme is design. It does not schedule, nor does it say how the ships, etc., are to get from their present position to form the screen.

However, Naval forces often get involved in complex situations which involve scheduling, such as replenishment, amphibious operations and disaster relief. Such applications are subject to pre-planning. In the real-time world the operator managing the situation (e.g. our Task Force Flag Officer) needs the support of a planning aid, which can store the prepared plan and then support the development and modification of the plan in the light of an evolving real world situation and the decisions of the operating team.

It was suggested earlier that the E.W. tasking aid devised by ARE could be considered to be a plan, held in the knowledge base in the form of an inference network, which was modified by the operator to form an acceptable solution. We are investigating the possibility of representing the synthesis of plans in the form of a forward chaining Expert System, which would result in a collection of goal assertions of activities to be performed. These activities would not be temporarily ordered in sequence at this pre-planning stage, although dependencies and inter-relationships with other activities would form part of their description.

To form a current plan the Expert System would be evoked to prepare schedules showing which activities have to pre-empt other activities taking account of the conflicts arising from demands on various resources and the evolving real-time situation.

It is proposed that the role played by the system during the development of the plan will be that of an adviser. The plans generated will be capable of being influenced during the development process by the operator, extensive use being made of graphical representations of the decision processes active within the machine. This would appear to require a clearer representation to the operator of the dominant paths influencing the goal assertions within the inference network than is currently available with today's Expert System technology. The system must be able to assist in decisions where relevant information may be uncertain or unknown.

Usually Expert Systems are in the class where goal hypotheses are confined to a single instance in time. This however does not constitute a plan. A plan is a collection of assertions viewed as an abstract simulation of the future i.e. what is required is a means of representing states of a process that change with time. A software package for implementing Expert Systems caled ART (Automated Reasoning Tool) has recently been announced by Inference Corporation of Los Angeles. An important feature of ART is that it has a viewpoint mechanism, a facility for modelling hypothetical alternatives and situations that change with time. Among other facilities of ART is its support of Belief Rules, which are used to recognise desirable viewpoints, either to aid processing or to streamline it.

As a means of demonstrating these ideas we have been considering a conceptual command and control system called DRAMA (standing for Disaster Relief Automated Management Aid). This is an application which involves a number of phases of planning, an often difficult management environment, and the application of a number of types of expertise (i.e. a multi-agent environment). It could also potentially involve very large data knowledge bases. A proposed schematic of the system is shown in Fig. 8.

An outline of the proposed demonstration software is given in Fig. 9. It consists of a number of expert agents each responsible for generating plans appropriate to their own domains of expertise (e.g. the generation of evacuation plans, disaster assessment plans; medical support plans, economic assistance plans; etc) cooperating over a blackboard to produce a fully interleaved integrated plan which will allow as many actions to proceed in parallel as possible.

The user will be able to interact with the system during the planning process, watching it proceed automatically or taking control as desired. The planning system is capable of generating both the single agent plans and the final multi agent plan. Invalid user defined activity sequences (i.e. those that break precedence constraints) are brought to the users attention, as are incompatibilities and conflicts between or within single agent plans.

A system such as this will find application in military Command and Control applications, such as the control of amphibious landings.

5. CONCLUSIONS

From our experience so far we have concluded that the development of Expert Systems has reached the state where worthwhile operational systems can be provided. The successful applications to date usually apply to a single expert domain where there are either recognised experts or established rules and procedures, as illustrated by E.W. Tasking and Force Deployment.

In current systems, such as Close In Weapon Systems, where there are real-time constraints and simple rules, algorithmic processes are the viable mechanism. In complex situations, where the solutions cannot be predicted, some form of knowledge based solution is required. Mechanisms are required which can handle real world situations - uncertainty of information and the impossibility of full goal satisfaction during execution. We are entering a period where the tools available to realise Expert Systems will improve significantly, as witnessed by new software development packages and the increasing performance and lowering of cost of LISP processors and Data Base Management Systems. To an increasing extent the problem of new weapons systems will lie not in the weapons themselves, but in our ability to control them.

Such systems, as exemplified by President Reagan's "Star Wars" initiative, will be a driving force in the development of "Expert Systems" and other IKBS techniques. Although we anticipate major advances over the next decade, it is difficult to predict the ultimate effects on more conventional Command and Control Systems, such as Naval Command systems. We believe that, because of the constraints of ship construction and refit schedules, the automation of Naval Command and Control will be introduced by a process of evolution. It will be interesting to see how A.I. will impact on human decision making, for this is seen as being central to both the responsibilities of Naval staff and their careers.

ACKNOWLEDGEMENTS

The paper is largely based on work and ideas originating from the Admiralty Research Establishment (Portsdown) and the Bracknell Division of Ferranti Computer Systems Limited. The author would particularly like to acknowledge contributions from Dr. W. Johnston, Mr. R.S. Whitlock, Mr. W.A. Hoggarth, Mr. W.J. Bingham and Mr. J.D. Stacey.

REFERENCES

1. D.J. Shellard, "C3 Within a Naval Ship", Advances in Command, Control and Communication Systems, April 1985, IEE Conference Publication No. 247, pp 186-191.
2. Dr. W. Johnston, "Proposal to Use the Anti-Missile Ship-Defence Problem to Assess Command System Theory Methods", unpublished Ferranti internal memorandum, October 1983.
3. W.L. Lakin and J.A.H. Miles, "IKBS in Multi-Sensor Data Fusion", Advances in Command and Control and Communications Systems, April 1985, IEE Conference Publication No. 247, pp 234-240.
4. Mrs. A. Roberts, "A Knowledge-Based System for E.W. Tasking Plan Evaluation", Proc. of the Third Seminar on Application of Machine Intelligence to Defence Systems, June 1984, RSRE, Malvern, England.
5. R.E. Wright, P.J. Burton, C. Lambert, J.S. Barr, "The Tactical Applications of Microwave Propagation Prediction", Agard Conference Proceedings No. 331 (Supplement), Propagation Effects of ECM Resistant Systems in Communication and Navigation, pp 17-1 to 17-13.

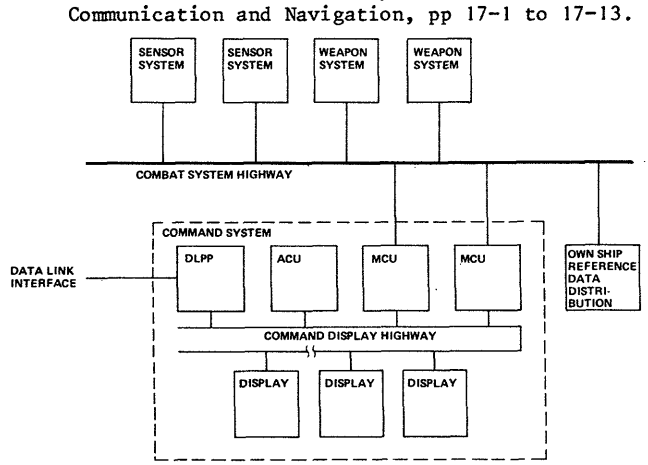


FIGURE 1 CACS–CONCEPTUAL COMBAT SYSTEM

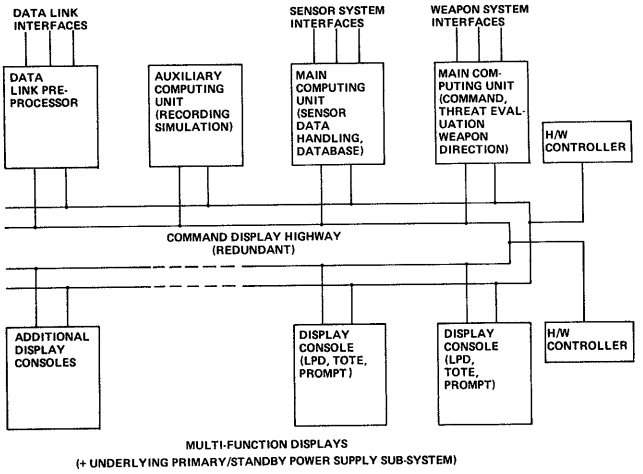


FIGURE 2 CACS–CONCEPTUAL COMMAND SYSTEM

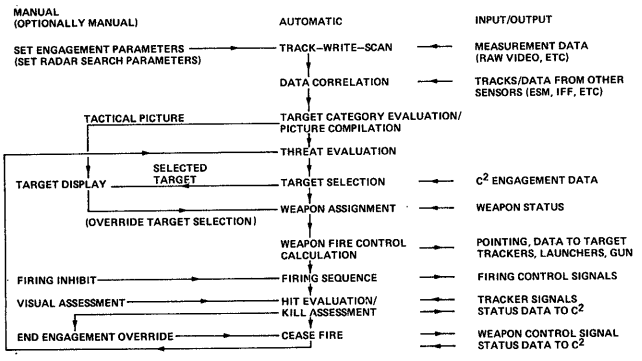


FIGURE 3 TYPICAL CIWS FLOW DIAGRAM

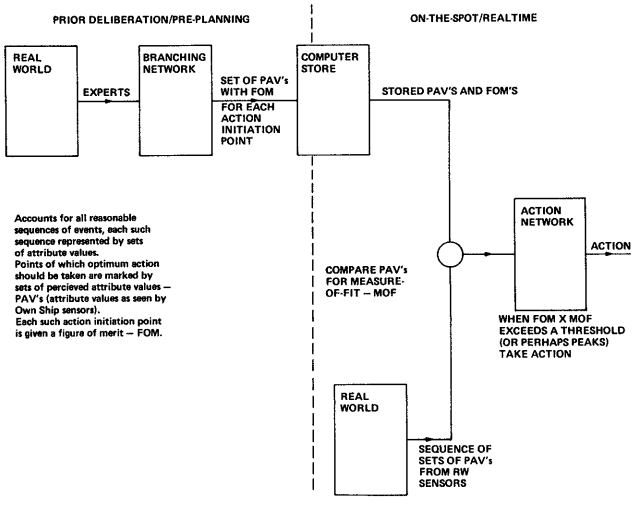


FIGURE 4 CONCEPTUAL DECISION SYSTEM

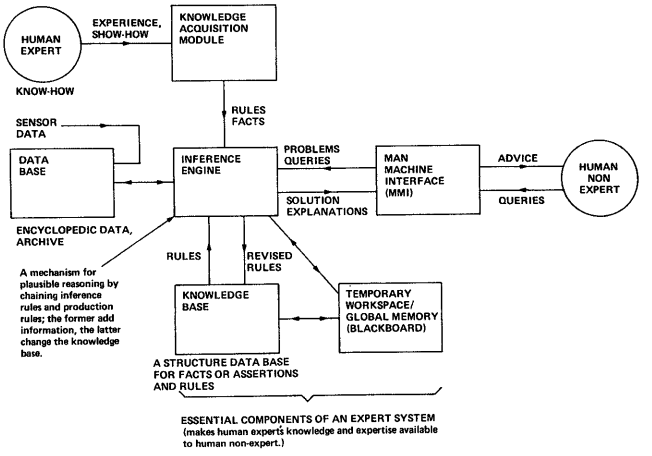


FIGURE 5 EXPERT SYSTEM STRUCTURE

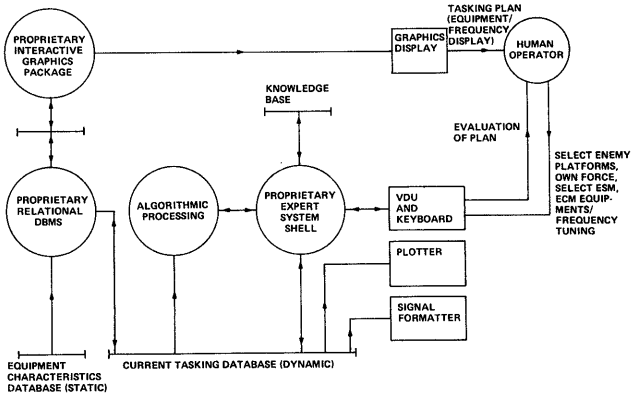


FIGURE 6 ARE EW TASKING ADVISER

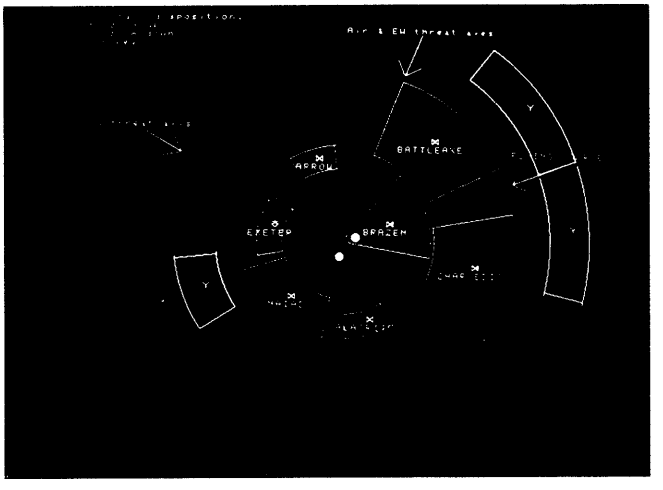


FIGURE 7 TYPICAL FLOATS PLAN POSITION DISPLAY

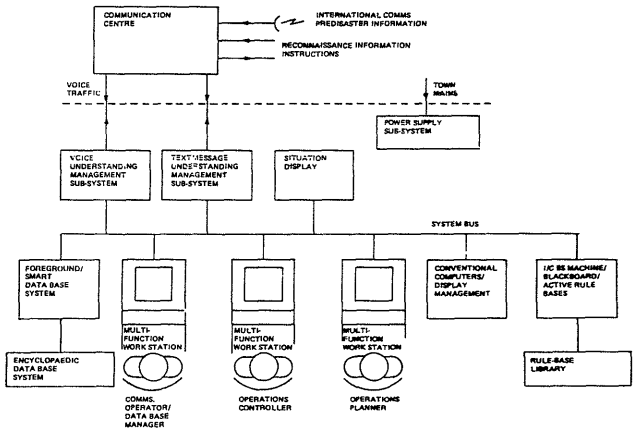


FIGURE 8 CONCEPTUAL DRAMA PRODUCTION SYSTEM CONFIGURATION

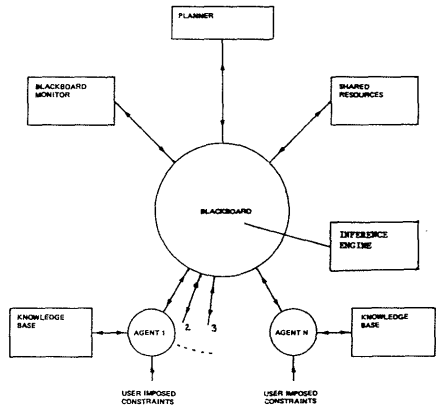


FIGURE 9 DRAMA DEMONSTRATION SYSTEM SOFTWARE

MARKOVIAN MODELING OF CANONICAL C^3 SYSTEMS COMPONENTS

Izhak Rubin* Israel Mayk#

*IRI Corporation, Tarzana, CA and UCLA, Los Angeles, CA
#USArmy CECOM, COMM/ADP Center, Fort Monmouth, NJ

Abstract

We have developed a family of Markovian stochastic process models for generic C^3 systems based upon the C^3 Canonical Reference Model [1]. The Markovian models are constructed to characterize the dynamic evolution of the underlying C^3 system. The models depend upon the existence of attrition and supply processes for both friendly (F) and adversary (G) forces. The attrition and supply intensities for these processes are determined from the capabilities of the C^3 systems of the F and G forces and thus completely determine the evolution of the force state. The functional description of these intensities are determined parametrically in terms of the characteristic capabilities of the various C^3 systems building blocks. The state space variables which are incorporated in this description involve the force states, identification and counter states, command and control states and communication network states. Performance equations are derived to compute the probability of win and the mean time to terminate the battle. Results are illustrated through the introduction of a specific C^3 pure-attrition model, the MPAC^3 I model, for which explicit performance behavior and sensitivity results are presented. We also show that the MPAC^3 I model provides an extension of the well-known Lanchester equations for aimed-fire and area-fire [2]. Extension to other combat theoretic equations commonly used in deterministic C^3 models may be made as well.

INTRODUCTION

The canonical reference model for C^3 systems [1] describes two major C^3 subsystems which combine to form generic C^3 systems. Even complex nested architectures of C^3 systems may be decomposed into the identification (ID) subsystem and the counter (CO) subsystem. The identification subsytem is responsible for surveillance maneuvering, making observations and comparing them to intelligence in an effort to assess the threat. Intelligence is simply defined as a set of observations which was previously processed and correlated for potential threat capability. The counter subsystem is responsible for carrying out appropriate counter actions such as weapon maneuver, engagement and disengagement. The aforementioned subsystems may be further decomposed into a set of interconnected irreducible primitive elements. The command element is responsible for allocating its plant(sensor or weapon) resources. The control element is responsible for maneuvering and activating the plants. The transmission and reception elements are responsible for timely communications between and among all of the elements which may be dispersed and/or distributed on the battlefield. Finally, the identification and counter plants provide for an effective means with which the C^3 system may interact with the environment including the threat. The identification plant is responsible for receiving signatures from the environment in the form suitable for signal processing and target parameter identification. The counter plant is capable of inflicting damage upon suspected or confirmed targets.

MISSIONS AND PLANS

To breathe life into the canonical reference model the C^3 system must have a well defined mission or a prioritized set of missions. The mission provides the goals and objectives to be achieved. A typical mission for a tactical C^3 system might be, for example, to seek out the enemy and win a likely confrontation in a reasonable amount of time and with minimum losses. The commander and his staff develop a plan to support the mission. The plan should address how resources are allocated to perceive and observe the environment and the status of his own C^3 system, i.e., the state-of-nature. Next, the plan must provide for understanding the state of nature and assessing consequences of possible courses of actions and strategies (a set of observation-action pairs). The consequences should be relevent to the mission. Finally, the plan should address how available resources are allocated to request or direct, execute and report actions consistent with the planned strategies. The plan should be reviewed for consistency with the mission and updated as often as possible. Once the plan is approved by the commander, it is delivered to the control element, typically of lower echelon or rank. The controller then implements the plan in a set of procedures, tasks and/or orders which control the sensor and/or weapon system actions.

MEASURES OF PERFORMANCE AND EFFECTIVENESS

Given the structure of a C^3 system one may readily define measures of performance (MOPs). For each C^3 process, one can measure the time it takes a given input to produce a desired output and call that a MOP. The structure of a C^3 system, however, is only a "means to an end", i.e., the means to accomplish the mission. A given MOP may or may not reflect upon the potential for the success of the mission. When a MOP is directly related or correlated with the success of the mission, it may contribute to or be associated with a measure of effectiveness (MOE). Typical MOEs measure the ability to make certain types of observations or to take certain types of actions with given error and false-alarm rates. In addition, losses, delays, throughput and timeliness are used to define both MOPs and MOEs.

To illustrate the distinction between MOPs and MOEs, consider a C^3 system with the mission to destroy hostile objects. Let the C^3 system include a range-independent missile-based CO subsystem and a radar-based ID subsystem designed to identify and track various objects. Assume that the radar can provide azimuth, height and range information. The missile on the other hand can only use azimuth and height information. Thus, the ID subsystem is chartacterized by MOPs relating identification capabilities to azimuth, height and range information whereas the CO subsystem is characterized by MOPs relating counter capabilities to azimuth and height information. Since in this case range is irrelevent to the CO subsystem, only azimuth- and height-related MOPs contribute to the C^3 system MOEs. The ID MOPs which are range-related do not contribute to the mission of the C^3 system and by definition cannot contribute to the C^3MOEs. It is a challenge to the C^3 scientist to identify and evaluate MOPs relevant to MOEs in the context of a given mission. The problem is especially complicated by the fact that, typically, MOEs are functions of more then one MOP. Moreover, since the same MOP may be found in more than one MOE, typical MOEs should follow naturally from the statement of the mission of the C^3 system and are often inter-dependent as we show below. Note also that it is also likely that MOEs may be function of lower level MOEs as is the case with MOPs. It is therefore important to define and obtain MOEs and MOPs most relevent to the mission statement of the problem i.e., the mission objective function.

THE MISSION OBJECTIVE FUNCTION

The C^3 system mission objective function U_f is a well defined mathematical statement which captures the essence of the mission of the C^3 system. Cast in terms of key C^3 MOPs or even MOEs it may be possible to optimize U_f, in structure and performance alike, to yield the greatest probabilities for the best possible mission results. Thus in general

$$U_f = U_f\{MOE_1(), MOE_2(), \ldots, MOE_i(), \ldots, MOE_n()\}$$

where

$$MOP_i() = MOP_i\{MOP_{i1}(), MOP_{i2}(), MOP_{ij}()\ldots, MOP_{im}()\}.$$

Consider, for example, the generic mission statement: "Seek out the enemy and deal a decisive blow in a minimum amount of time and with minimum losses". We use the assumptions and definitions inherent in the mission statement to formulate a quantitative objective function. We assume that only two forces are involved: a friendly force F and an adversary/enemy force G. Immediately prior to the battle/confrontation, the strength of force F and G are denoted by l_i^F and l_i^G, respectively. At the end of the battle, the desired win-strength-threshold for of the surviving force F is denoted by l_f^F and that of force G is denoted by l_f^G Hence, the initial and final force state boundary conditions are defined by $\underline{l}_i = (l_i^F, l_i^G)$ and $\underline{l}_f = (l_f^F, l_f^G)$, respectively. During the course of the battle the force state strengths are denoted by $\underline{N} = (N^F, N^G)$ where $l_i^F \geq N^F \geq l_f^F$ and $l_i^G \geq N^G \geq l_f^G$. Due to political, reinforcement and/or logistics reasons the mission must be accomplished within an average time $D_b < D_m$. Subjectively, for example, for all $N^G < 0.5 l_i^G$, force F may be said to have dealt a "decisive blow" to force G and that for all $N^F > 0.5 l_i^F$, force F incurred "tolerable losses". The statement of the mission, therefore, implies that the mission objective function should be a function of the initial and final force state strengths. The mission objective function, however, is a function of the MOEs and therefore the MOEs in turn must be functions of the initial and final force state strengths.

We identify at least three critical MOEs for this objective function: the probability of win, P_w, the expected friendly losses given a friendly win, L_w, and the expected duration of battle, D_b. Using these high level MOEs, one may try to optimize meaningful structures for $U_f = U_f(P_w, L_w, D_b)$ as, for example,

$$U_f = (P_w)^p \cdot (L_w)^{-q} \cdot (D_b)^{-r}$$

or as

$$U_f = (a_1 P_w)^p + (a_2 L_w)^{-q} + (a_3 D_b)^{-r}$$

where a_1, a_2, a_3, p, q and r are non-negative real numbers.

THE MARKOVIAN ATTRITION-REINFORCEMENT FORMALISM

The states of a C^3 system may be characterized by a vector stochastic process $\underline{Z}(t) = \{\underline{Z}_t, t \geq 0\}$ which describes the dynamic evolution of the C^3 system. The global random vector state variable $\underline{Z}_t$ represents the system state random variables at time t. $\underline{Z}(t)$ incorporates the continuous-time process definitions. Embedded in $\underline{Z}(t)$, we define discrete-time state/jump and point/arrival processes denoted by $\underline{Z}(n) = \{\underline{Z}_n, n=0, 1, 2, \ldots\}$.

We assume, in addition, that $\underline{Z}(t)$ or $\underline{Z}(n)$ may be characterized by a coherent and consistent set of stochastic process models such as Markov, semi-Markov, Markov-Renewal, Regular Point [3], Regular Jump [4] and m-memory [5] stochastic process models.

Let $\mathfrak{Z}$ represent the total C^3 state space available for $\underline{Z}(t)$ or $\underline{Z}(n)$. $\mathfrak{Z}$ consists of two major state subspaces, $\mathfrak{Z}_\alpha$ and $\mathfrak{Z}_\beta$ such that $\mathfrak{Z} = \{ \mathfrak{Z}_\alpha , \mathfrak{Z}_\beta \}$. The state space $\mathfrak{Z}_\alpha$ includes the functional (e.g. offensive /defensive, mission-oriented) observation/decision/action spaces which apply to identification and counter processes and their associated command, control, and communications processes. Thus $\mathfrak{Z}_\alpha$ is called the functional state space. The state space $\mathfrak{Z}_\beta$ is called the physical state space and includes observables applied to the spatial and temporal distributions of assets which may be associated with both forces F and G. As shown in Figure 1, $\mathfrak{Z}_\alpha$ consists of two major state subspaces, $\mathfrak{Z}_\alpha^F$ and $\mathfrak{Z}_\alpha^G$ and similarly, $\mathfrak{Z}_\beta$ consists of two major state subspaces, $\mathfrak{Z}_\beta^F$ and $\mathfrak{Z}_\beta^G$. Thus $\mathfrak{Z}_\beta^F$ represents the physical state space available for the force F vector process $\underline{X}^F(t)$ given by $\underline{X}^F(t) = \{\underline{X}_t^F, t \geq 0\}$ and $\mathfrak{Z}_\beta^G$ represents the physical state space available for the force G vector process $\underline{X}^G(t)$ given by $\underline{X}^G(t) = \{\underline{X}_t^G, t \geq 0\}$ where $\underline{X}_t^F$ and $\underline{X}_t^G$ represent the state random variables of friendly and adversary assets, respectively, at time t. In general, $\underline{X}_t^F$ and $\underline{X}_t^G$ include variable components for readiness and deployment status such as the number and type of units, location, movement and logistic levels.

The C^3 system model assumes that state space transitions within and between the functional state spaces cause transitions to occur in the physical state spaces. Similarly, state space transitions within the physical force space may cause and affect transitions within and between functional spaces. Since the perceived states are only stochastic estimates of the true states, stochastic behavior is exhibited for many transitions in both types of spaces even though many highly deterministic relationships may be defined.

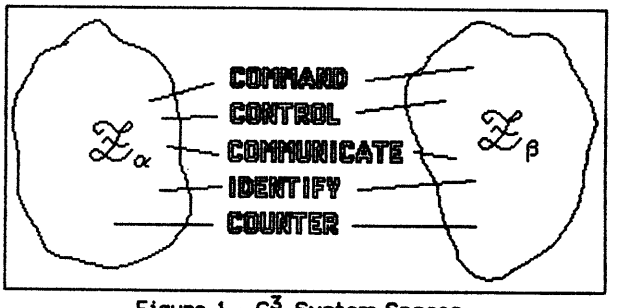


Figure 1 - C^3 System Spaces

Consider a sample of attrition and reinforcement transitions which occur in the physical state space $\mathfrak{Z}_\beta$ given by the multi-dimensional vector process $\underline{X}(t) = \{\underline{X}_t, t \leq 0\}$ where $\underline{X}_t = (X_t^F , X_t^G)$. One state space of $\underline{X}(t)$ represents the total number of C^3 units or assets in each of the forces. The corresponding two-dimensional stochastic vector process is given by $\underline{N}(t) = \{\underline{N}_t, t \leq 0\}$ where $\underline{N}_t = (N_t^F, N_t^G)$ represents the total number of effective friendly and adversary fighting units in the respective C^3 system, at time t. Thus, N_t^F and N_t^G are random variable components of $\underline{X}_t^F$ and $\underline{X}_t^G$, respectively, and therefore $\underline{N}(t)$ is a four-dimensional counting stochastic process which is decomposed in terms of a) the two-dimensional state sequence process given by $\underline{N}(n) = \{\underline{N}_n, n=0, 1, 2, \dots\}$ and b) the two-dimensional point process given by $\underline{A}(n) = \{\underline{A}_n, n=0, 1, 2, \dots\}$. The state sequence random variable vector is given by $\underline{N}_n = (N^F_n, N^G_n)$ where N^F_n and N^G_n are the random variables representing the total assets inherent in force F and G , respectively following the n^{th} attrition/reinforcement transition in assets. The point arrival random variable vector is given by $\underline{A}_n = (A^F_n , A^G_n)$ where A^F_n and A^G_n are the random variables representing the n^{th} attrition/reinforcement arrival times for the total assets inherent in force F and G, respectively. Note that $\underline{N}(t)$ is therefore a jump vector stochastic process whose state within each small time increment either stays the same , decreases due to an attrition or increases due to a reinforcement. $\underline{N}(t)$ is a continuous-time process for which simultaneous occurences of events occur with probability 0.

Presently, let the discrete-time C^3 process be given by

$$\underline{Z}(n) = \{\underline{N}_n , \underline{A}_n , n = 0, 1, 2, \dots\}.$$

$\underline{Z}(n)$ is completely characterized by the following set of conditional probabilities which we term as the C^3 process kernel ,

$$K(\underline{i}_{n+1}, t, n \mid \underline{i}_0, \dots, \underline{i}_n, t_0, \dots, t_n) =$$

$$P(\underline{N}_{n+1}=\underline{i}_{n+1}, A_{n+1}-A_n > t \mid \underline{N}_0=\underline{i}_0, \dots, \underline{N}_n=\underline{i}_n, A_0=t_0, \dots, A_n=t_n)$$

for any state levels $\underline{i}_0, \dots, \underline{i}_n$, and times $t_0, \dots, t_n$, $t \geq 0$, $n \geq 0$.

For a C^3 process which is characterized by a Markov-Renewal stochastic process the C^3 process kernel is given by

$$K(\underline{i}_{n+1}, t, n \mid \underline{i}_n, \dots, \underline{i}_n, t_0, \dots, t_n) =$$

$$P(\underline{N}_{n+1} = \underline{i}_{n+1}, A_{n+1}-A_n > t \mid \underline{N}_n = \underline{i}_n, A_n = t_n)$$

for any states $\underline{i}_0, \dots, \underline{i}_n$, and times $t_0, \dots, t_n$, $t \geq 0$, $n \geq 0$. Note that the associated continuous-time stochastic vector process $\underline{N}(t)$ is semi-Markov and the embedded state sequence process $\underline{N}(n)$ is a discrete-time Markov chain, governed by a transition probability function given by

$$R(\underline{i}_n, \underline{i}_{n+1}) = P\{\underline{N}_{n+1} = \underline{i}_{n+1} \mid \underline{N}_n = \underline{i}_n\}$$

where $\underline{i}_n = (i^F_n , i^G_n)$ represents the total number of C^3 units/assets counted at $t_n \le t < t_{n+1}$. We characterize the process $\underline{N}(n)$ as a time-homogeneous semi-Markov reinforcement-attrition C^3 process by noting that four types of transitions may occur between any two successive jumps n and n+1, as induced by the events depicted in Figure 2 and described below:

a) Attrition in the force F. The magnitude of the attrition is given by the random variable $u_-^F(\underline{i})$ C^3 units with probability $u_-^F(k) = P\{ u_-^F(\underline{i}) = k \}$. The time delay since the last event is given by the random variable $T_-^F(\underline{i})$ distributed according to $T_-^F(\underline{i},t) = P\{ T_-^F(\underline{i}) \le t \}$.

b) Attrition in the force G. The magnitude of the attrition is given by the random variable $u_-^G(\underline{i})$ C^3 units with probability $u_-^G(k) = P\{ u_-^G(\underline{i}) = k \}$. The time delay since the last event is given by the random variable $T_-^G(\underline{i})$ distributed according to $T_-^G(\underline{i},t) = P\{ T_-^G(\underline{i}) \le t \}$.

c) Reinforcement of force F. The magnitude of the reinforcement is given by the random variable $u_+^F(\underline{i})$ C^3 units with probability $u_+^F(k) = P\{ u_+^F(\underline{i}) = k \}$. The time delay since the last event is given by the random variable $T_+^F(\underline{i})$ distributed according to $T_+^F(\underline{i},t) = P\{ T_+^F(\underline{i}) \le t \}$.

d) Reinforcement of force G. The magnitude of the reinforcement is given by the random variable $u_+^G(\underline{i})$ C^3 units with probability $u_+^G(k) = P\{ u_+^G(\underline{i}) = k \}$. The time delay since the last event is given by the random variable $T_+^G(\underline{i})$ distributed according to $T_+^G(\underline{i},t) = P\{ T_+^G(\underline{i}) \le t \}$.

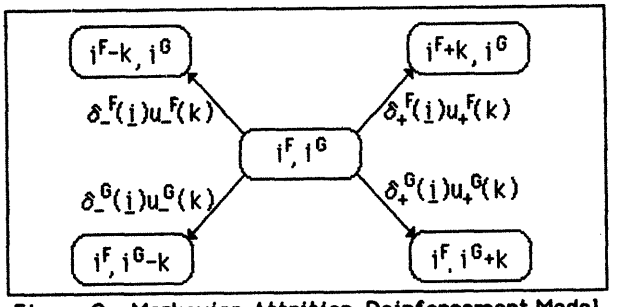


Figure 2 - Markovian Attrition-Reinforcement Model

It is assumed in this model that at each event, the computation of the distributions of the time to the next F or G attrition or reinforcement is reset and, therefore, the distributions of the time delays depend only on the C^3 process state $\underline{i}$. It is noted that $T_\pm^F$ and $T_\pm^G$ express the time durations that need to expire while $\underline{N}_n = \underline{i}$ before the corresponding next event occurs. However, we observe that only one of the four possible events will be realized. This event is the outcome with the shortest time interval.

Let $\delta_-^F(\underline{i})$ represent the probability that the next event is an attrition of force F and let $\delta_+^F(\underline{i})$ denote the probability that the next event is a reinforcement of force F. Similarly, for force G we define $\delta_-^G(\underline{i})$ and $\delta_+^G(\underline{i})$. Then, we have

$$R(\underline{i}_n , \underline{i}_{n+1})$$

$$= \delta_-^F(\underline{i}_n)u_-^F(k), \quad \text{for } i_{n+1}^F = i_n^F - k , \; i_{n+1}^G = i_n^G$$

$$= \delta_-^G(\underline{i}_n)u_-^G(k), \quad \text{for } i_{n+1}^F = i_n^F, \; i_{n+1}^G = i_n^G - k$$

$$= \delta_+^F(\underline{i}_n)u_+^F(k), \quad \text{for } i_{n+1}^F = i_n^F + k , \; i_{n+1}^G = i_n^G$$

$$= \delta_+^G(\underline{i}_n)u_+^G(k), \quad \text{for } i_{n+1}^F = i_n^F, \; i_{n+1}^G = i_n^G + k$$

corresponding to each type of event a)-d) above. Note that the probability that the next event is a force F event is given by

$$\delta^F(\underline{i}) = \delta_-^F(\underline{i}) + \delta_+^F(\underline{i}).$$

Similarly, the probability that the next event is a force G event is given by

$$\delta^G(\underline{i}) = \delta_-^G(\underline{i}) + \delta_+^G(\underline{i}).$$

Therefore,

$$\sum_{\underline{i}_{n+1}} R(\underline{i}_n , \underline{i}_{n+1}) = \delta^F(\underline{i}_n) + \delta^G(\underline{i}_n)) = 1.$$

Recall that the C^3 process starts with the state $\underline{i}_i = (i_i^F, i_i^G)$ and continues until the state

$$\underline{N}_n = (N_n^F > i_f^F, N_n^F = i_f^G) \quad \text{or} \quad \underline{N}_n = (N_n^F = i_f^F, N_n^G > i_f^G)$$

is reached. If i_f^G is reached then force F is said to have won the battle. If i_f^F is reached then force G is said to have won the battle. To obtain the probability of win for force F ,therefore, we need to prescribe $\underline{i}_i = (i_i^F, i_i^G)$ and $\underline{i}_f = (i_f^F, i_f^G)$. It is assumed that when the size of force F reaches its final value i_f^F, it cannot continue to operate effectively and it is said to have lost the confrontation. Let the first hitting times of any state i^F by force F and of any state i^G by force G be given, respectively, by

$$T_h^F(i^F) = \min\{ t : t \ge 0 , N_t^F = i^F \},$$

and

$$T_h^G(i^G) = \min\{ t : t \ge 0 , N_t^G = i^G \}.$$

The probability of win $P_w(\underline{i}_i)$ is defined by the probability that that the confrontation is won by force F, which will happen if force G reaches its final state i_f^G before force F reaches its final state i_f^F. Thus,

$$P_w(\underline{i}_i) = P\{ \text{Force F wins} \mid \underline{i}_i \} = P\{ T_h^G(i_f^G) < T_h^F(i_f^F) \mid \underline{i}_i \}.$$

The mean-time-to-termination of the battle D_b is defined as the average time it takes for force F or G to win. Thus,

$$D_b = E[\min[T_h^F(i_f^F), T_h^G(i_f^G)]].$$

Consider the much simplified but highly instructive case under which only attrition events are possible (no reinforcements), each attrition is of unit magnitude and the strength process is Markov, then the time delays for these events are exponentially distributed, according to

$$P[T\_^F(\underline{i}) > t] = \exp[-\mu^F(\underline{i})t], \; t \geq 0;$$

and

$$P[T\_^G(\underline{i}) > t] = \exp[-\mu^G(\underline{i})t], \; t \geq 0.$$

The attrition intensities are defined by

$$\mu^F(\underline{i}) = \lim_{\Delta t \to 0} (\Delta t)^{-1} P[N^F_{t+\Delta t} - N^F_t < 0 \mid \underline{N}_t = \underline{i}]$$

and

$$\mu^G(\underline{i}) = \lim_{\Delta t \to 0} (\Delta t)^{-1} P[N^G_{t+\Delta t} - N^G_t < 0 \mid \underline{N}_t = \underline{i}]$$

From these definitions we note that

$$P[N^F_{t+\Delta t} - N^F_t < 0 \mid \underline{N}_t = \underline{i}] = \mu^F(\underline{i})\Delta t + o(\Delta t)$$

$$P[N^G_{t+\Delta t} - N^G_t < 0 \mid \underline{N}_t = \underline{i}] = \mu^G(\underline{i})\Delta t + o(\Delta t)$$

where $\lim_{\Delta t \to 0} o(\Delta t)/\Delta t = 0$

Given that the C^3 process is in state $\underline{N}_t = \underline{i}$, therefore, the probability that in a small interval of duration Δt an attrition event for force F will occur is approximately given by $\mu^F(\underline{i})\Delta t$ and similarly for force G.
In observing the evolution of this process, we note that the process stays in state $\underline{i}_n$ for a period $T(\underline{i})$ which is exponentially distributed with probability

$$P[T(\underline{i}) > t] = \exp[-\mu(\underline{i})t], \; t \geq 0,$$

where

$$\mu(\underline{i}) = \mu^F(\underline{i}) + \mu^G(\underline{i}).$$

The process then experiences an attrition in the force F with probability $\delta\_^F(\underline{i}) = \mu^F(\underline{i})/\mu(\underline{i})$, and an attrition in force G with probability $\delta\_^G(\underline{i}) = \mu^G(\underline{i})/\mu(\underline{i})$.

The probability of win $P_w(\underline{i})$ may therefore be computed by solving the following set of recurrence relations:

$$P_w(\underline{i}) = \delta\_^F(\underline{i})\, P_w(i^F-1, i^G) + \delta\_^G(\underline{i})\, P_w(i^F, i^G-1)$$

for $i^F > i_f^F$ and $i^G > i_f^G$ under the final boundary conditions

$$P_w(i_f^F, i^G) = 0 \text{ for } i^G \geq i_f^G;$$

and

$$P_w(i^F, i_f^G) = 1 \text{ for } i^F > i_f^F.$$

The mean-time-to-termination of the battle is computed by solving the following set of recurrence relations:

$$D_b(\underline{i}) = [\mu(\underline{i})]^{-1} + \delta\_^F(\underline{i})\, D_b(i^F-1, i^G) + \delta\_^G(\underline{i})\, D_b(i^F, i^G-1)$$

for $i^F > i_f^F$ and $i^G > i_f^G$ under the final boundary conditions

$$D_b(i_f^F, i^G)] = 0 \text{ for } i^G \geq i_f^G;$$

and

$$D_b(i^F, i_f^G)] = 0 \text{ for } i^F > i_f^F.$$

Note that the MOEs P_w and D_b are totaly determined by $\mu^F(\underline{i})$ and $\mu^G(\underline{i})$. Hence, $\mu^F(\underline{i})$ and $\mu^G(\underline{i})$ are also MOEs and any MOP which may be relevant to P_w and D_b must also be imbedded in $\mu^F(\underline{i})$ and $\mu^G(\underline{i})$. This point is illustrated using a basic C^3 system model refered to herein as the Markov Pure Attrition C^3 (MPAC^3) Model I.

THE MPAC^3 MODEL I

Consider the capability of the ID subsystem of force F to detect force G of magnitude $N_t^G = i^G$, at a magnitude error $E_I^F = k$ for any given detection event D^F. For the purpose of illustration let the error E_I^F in estimating N_t^G within a characteristic sensing time interval be Poisson distributed. Therefore,

$$P[D^F, E_I^F = k \mid N_t^G = i^G] = P_D^F(i^G)\, P[E_I^F = k \mid N_t^G = i^G, D^F]$$

where

$$P[E_I^F = k \mid N_t^G = i^G, D^F] = \lambda_I^F(i^G)^k (k!)^{-1} \exp[-\lambda_I^F(i^G)], \; k = 0,1,2,\ldots$$

Thus the average size of the error *i.e.* the deviation in estimating the size of force G, given that there are i^G units and that detection of force G occured is given by

$$E[E_I^F \mid N_t^G, D^F] = \lambda_I^F(i^G).$$

Analogously, consider the capability of the ID subsystem of force G to detect force F of magnitude $N_t^F = i^F$ at a magnitude error $E_I^G = k$, for any given detection event D^G. For the purpose of illustration let the error E_I^G in estimating N_t^F within its characteristic sensing time interval be also Poisson distributed. Therefore,

$$P[D^G, E_I^G = k \mid N_t^F = i^F] = P_D^G(i^F)\, P[E_I^G = k \mid N_t^F = i^F, D^G]$$

where

$$P[E_I^G = k \mid N_t^F = i^F, D^G] = \lambda_I^G(i^F)^k (k!)^{-1} \exp[-\lambda_I^G(i^F)], \; k = 0,1,2,\ldots$$

Thus the average size of the error in estimating the size of force F, given that there are i^F units and that detection

of force F occured is given by

$$E[\, E_I^G \mid \mathbf{N}_t^F, D^G \,] = \lambda_I^G(i^F).$$

To simplify the illustration we also assume that

$$P_D^F(i^G) = P_D^G(i^F) = 1.$$

Without loss in generality let

$$\lambda_I^F(i^G) = \lambda_I^F f_I^F(i^G) \quad \text{and} \quad \lambda_I^G(i^F) = \lambda_I^G f_I^G(i^F).$$

where

$$\lambda_I^F,\ \lambda_I^G \geq 0 \quad \text{and} \quad f_I^F(i^G),\ f_I^F(i^G) \geq 0.$$

As a special case, it is reasonable to asssume that

$$f_I^F(i^G) = i^G \quad \text{and} \quad f_I^G(i^F) = i^F, \qquad i^G, i^G \geq 0.$$

For illustrational purposes, therefore, in this model, the error in size is directly proportional to the size being estimated.

Given a detection D^F and an error $E_I^F = k$ when $\mathbf{N}_t = \underline{i}$, the probability of a hit H^F which causes a unit magnitude attrition in force G within Δt units of time is given by

$$P\{\, H^F, \Delta t \,|\, D^F, \mathbf{N}_t = \underline{i}, E_I^F = k \,\} = \mu^G(D^F, \underline{i}, k)\, \Delta t + o(\Delta t).$$

Thus, the mean-time to the next unit attrition of force G by force F is given by

$$E\{\, T_h^F \mid D^F, \mathbf{N}_t = \underline{i}, E_I^F = k \,\} = 1/\mu^G(D^F, \underline{i}, k)$$

and

$$\mu^G(D^F, \underline{i}) = \sum_{k=0}^{k=\infty} \mu^G(D^F, \underline{i}, k)\, P\{\, E_I^F = k \,|\, D^F, \mathbf{N}_t = \underline{i} \,\}$$

Since we assumed that $P_D^F(i^G) = 1$, we must have

$$\mu^G(\underline{i}) = \mu^G(D^F, \underline{i}).$$

Analogously, given a detection D^G and an error $E_I^G = k$ when $\mathbf{N}_t = \underline{i}$, the probability of a hit H^G which causes a unit magnitude attrition in force F within Δt units of time is given by

$$P\{\, H^G, \Delta t \,|\, D^G, \mathbf{N}_t = \underline{i}, E_I^G = k \,\} = \mu^F(D^G, \underline{i}, k)\, \Delta t + o(\Delta t),$$

and the mean-time to the next attrition of force F by force G is given by

$$E\{\, T_h^G \mid D^G, \mathbf{N}_t = \underline{i}, E_I^G = k \,\} = 1/\mu^F(D^G, \underline{i}, k).$$

Averaging over all possible ID errors

$$\mu^F(D^G, \underline{i}) = \sum_{k=0}^{k=\infty} \mu^F(D^G, \underline{i}, k)\, P\{\, E_I^G = k \,|\, D^G, \mathbf{N}_t = \underline{i} \,\},$$

and since we also assumed that $P_D^G(i^F) = 1$, we must have

$$\mu^F(\underline{i}) = \mu^F(D^G, \underline{i}).$$

As a special case, we postulate that the attrition intensities may be approximated as

$$\mu^F(D^G, \underline{i}, k) = a_C^F\, i^F\, f_C^F(i^G, k),$$

and

$$\mu^G(D^F, \underline{i}, k) = a_C^G\, i^G\, f_C^G(i^F, k),$$

for each i^F, i^G, k . The factors $a_C^F\, i^F$ and $a_C^G\, i^G$ exhibits the linear law for aimed-fire as postulated by Lanchester. The form factors $f_C^F(i^G, k)$ and $f_C^G(i^F, k)$ may account for any degradation in aimed-fire resulting from errors such as estimating the size of the adversary force strength. To illustrate the impact of the ID errors on the CO hit rate, we assume that for a given estimation error $E_I = k$, the degradation to aimed-fire attrition decreases as the CO subsystem becomes less sensitive to the ID subsystem errors. For illustration ,let

$$f_C^F(i^G, k) = f_C^F(k) = \exp[-a_{CI}^F\, k] \ , k \geq 0 ,$$

and

$$f_C^G(i^F, k) = f_C^G(k) = \exp[-a_{CI}^G\, k] \ , k \geq 0 .$$

Note, that as desired, we have a) $f_C^F(i^G, 0) = 1$, when no identification errors occur, i.e., $E_I = 0$ and b) $f_C^F(i^G, k)$ decreases as k increases and thereby resulting in more area-fire and less aimed-fire. The parameters a_{CI}^F, a_{CI}^G are the ID-CO sensitivity factors which couple the MOPs of the two subsystems of force F and G, respectively. Averaging over k, we obtain

$$\mu^F(\underline{i}) = a_C^G\, i^G \exp[-\lambda_I^G\, i^F (1 - \exp[-a_{CI}^G])$$

and

$$\mu^G(\underline{i}) = a_C^F\, i^F \exp[-\lambda_I^F i^G (1 - \exp[-a_{CI}^F]).$$

Note that if $a_{CI}^F \ll 1$ and $a_{CI}^F \ll 1$

$$\mu^F(\underline{i}_n) = a_C^G\, i^G \exp[-\lambda_I^G\, i^F a_{CI}^G]$$

and

$$\mu^G(\underline{i}_n) = a_C^F\, i^F \exp[-\lambda_I^F i^G a_{CI}^F].$$

and, furthermore, if $\lambda_I^G\, i^F a_{CI}^G \ll 1$ and $\lambda_I^F i^G a_{CI}^F \ll 1$

$$\mu^F(\underline{i}_n) = a_C^G\, i^G (1 - \lambda_I^G\, i^F a_{CI}^G)$$

and

$$\mu^G(\underline{i}_n) = a_C^F\, i^F (1 - \lambda_I^F i^G a_{CI}^F).$$

Thus we see how the contribution of aimed-fire is mitigated by poor identification capability resulting in mixed aimed-area fires.

PERFORMANCE RESULTS FOR THE MPAC^3 I MODEL

To illustrate the types of performance results which may be obtained for the C^3 process using the first-order Markov single-unit attrition model described above, refer to Figures 3-10.

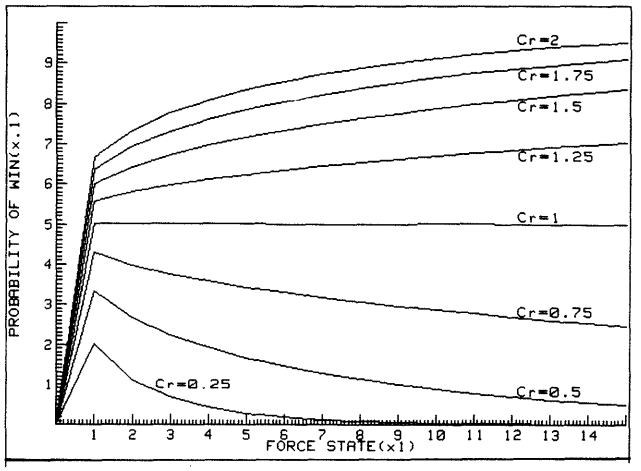


Figure 3 - P_w versus $I_i^F = I_i^G$ and C_r.

Figure 3 depicts results for P_w versus the initial forces which characterize the size of the battle given by $I_i^F = I_i^G$ and $I_f^F = I_f^G = 0$. For any initial forces, P_w is greatly impacted by the C^3 single-unit force counter capability ratio $C_r = a_C^F/a_C^G$ assuming a perfect identification function ($\lambda_I = 0$). We assumed that Force G is characterized by $a_C^G = 1$ successful counter action per unit C^3 system per unit time. Note that as expected when $a_C^F = a_C^G$ $P_w = 0.5$, when $a_C^F > a_C^G$, $P_w > 0.5$ and increases with diminishing returns as the size of the battle is increased, and finally when $a_C^F < a_C^G$, $P_w < 0.5$ and decreases with diminishing returns as the size of the battle is increased.

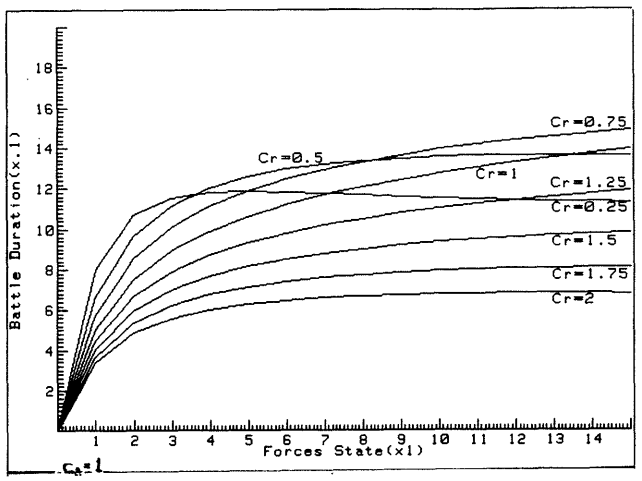


Figure 4 - D_b versus $I_i^F = I_i^G$ and C_r.

We therefore make the important observation that when $C_r < 1$, it is better to fight small battles and vice versa when $C_r > 1$, it is better to fight larger battles.

Figure 4 depicts results for D_b versus the same independent parameters described for P_w in Figure 3 above. We observe that for any given initial force $I_i^F = I_i^G$, D_b increases as C_r decreases to 0.75. This is due to the decrease in the counter capability of force F. Note, however, the crossover phenomenon which occurs for $C_r = 0.5$ and $C_r = 0.75$ when $I_i^F = I_i^G = 8$. As shown in Figure 3, as C_r decreases below $C_r = 1$, Force F has better chances of winning when the battles are smaller. This, however, results in larger D_b for small battles. As the size of the battle increases and C_r decreases the battle is ended in less time on the average since force G now is more and more likely to win.

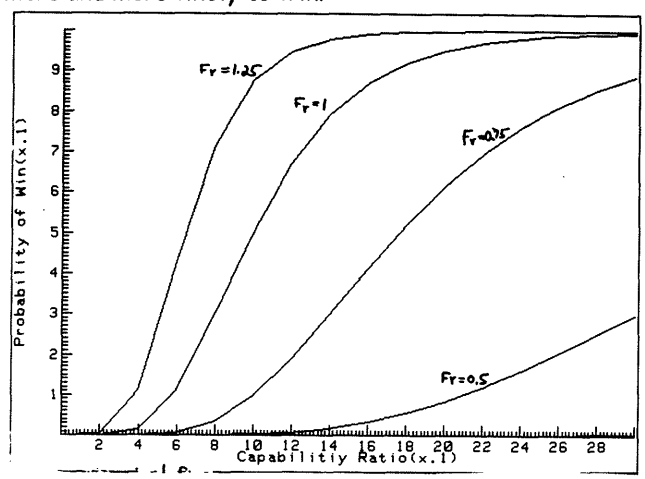


Figure 5 - P_w versus C_r and F_r.

Figure 5 depicts results for P_w vs. the capability ratio obtained as before by letting $a_C^G = 1$ and varying a_C^F from 0.1 to 3.0. For a given $C_r = a_C^F/a_C^G$, P_w is greatly impacted by the initial force ratio $F_r = I_i^F/I_i^G$. This is illustrated by letting $I_i^G = 16$. As expected, P_w increases as both F_r and C_r are increased thus demonsrating the well known force multiplier concept. Note, however, that as C_r increases above $C_r = 2$ and F_r increases above $F_r = 1$, P_w increases with diminishing returns. Moreover, P_w is much more sensitive to fighting outnumbered than to fighting with comparably inferior forces.

Figure 6 depicts results for D_b versus the same independent parameters described for P_w in Figure 5 above. For each F_r , note the global maximum for $D_b(C_r)$. This maximum occurs at higher C_r values as F_r decreases. This is due to the greater sensitivity of this MOE to F_r than to C_r for $C_r < 10$. Note, moreover, that as F_r increases, D_b is increasingly sensitive to C_r.

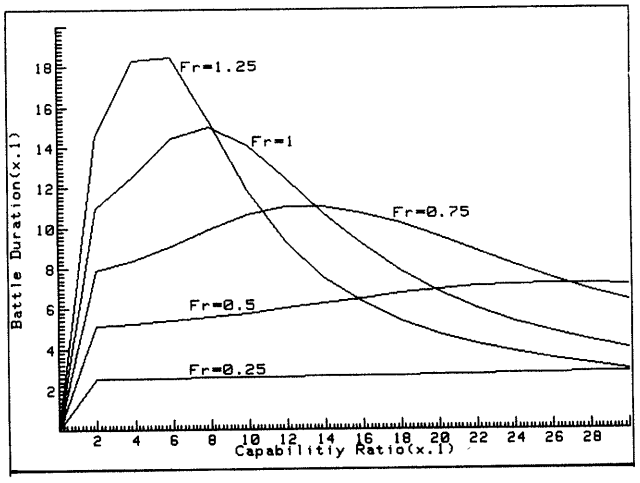


Figure 6 - D_b versus C_r and F_r.

In Figure 7 we see how P_w varies with the battle size for various identification capability ratios $I_r = \lambda_I^G / \lambda_I^F$. Recall that λ_I is the identification subsystem parameter which provides the average error in estimating the opponent force state per unit opponent. Thus a high I_r is desirable for force F. Indeed P_w increases as I_r increases for any battle size, as expected. Note that force F has the fire power advantage given by $C_r = 2$. When $I_r > 1$, P_w is not as critically affected as when $I_r < 1$. When $I_r < 1$ smaller battles are more desirable for force F.

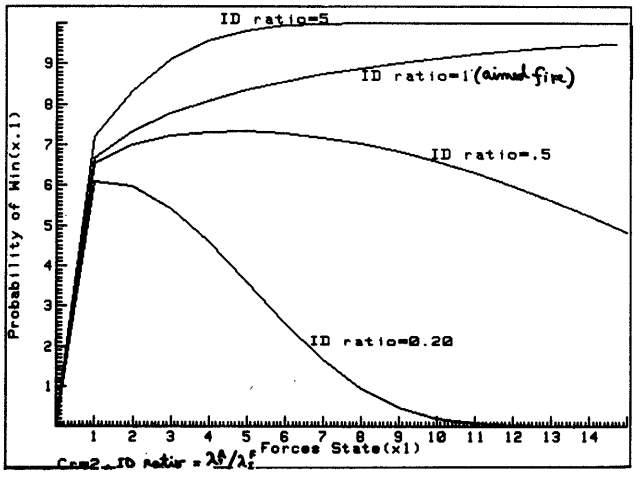


Figure 7 - P_w versus $I_i^F = I_i^G$ and I_r.

In Figure 8 we see how D_b varies with the battle size for various identification capability ratios. Note that when there are identification errors on both sides regardless of who has the advantage, battles last longer than when identification errors are zero (as is the case for perfect identification). Thus D_b is lower-bounded by the aimed-fire case. Significant departures from the minimum D_b occur when $I_r < 1$ with small scale battles ending quicker than large scale battles.

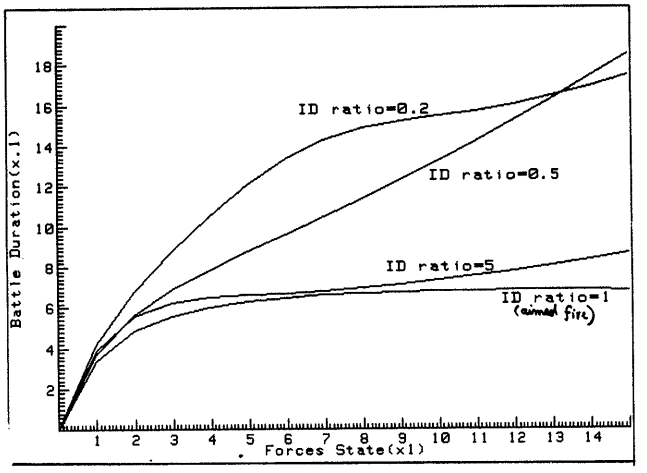


Figure 8 - D_b versus $I_i^F = I_i^G$ and I_r.

In Figures 9 and 10 we obtained P_w and D_b as functions of the initial force state while we varied the ratio of the ID-CO sensitivity factors of forces F and G given by $C_s = a^G_{CI} / a^F_{CI}$. Recall that as the sensitivity factor a_{CI} increases the counter becomes more dependent upon the identification subsystem, i.e. , $C_s > 1 (C_s < 1)$ implies that force G is more (less) sensitive to identification errors than force F. Therefore, as shown in Figure 9, P_w is higher for force F when $C_s > 1$ than when $C_s < 1$. Note, however, that when $C_s > 1$ only marginal improvements are obtained for force F. In contrast, when $C_s < 1$ force F loses the counter power advantage ($C_r = 2$) significantly, as the size of the battle increases.

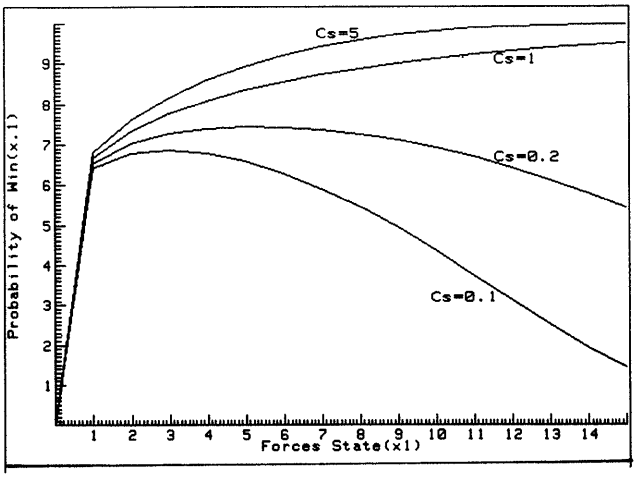


Figure 9 - P_w versus $I_i^F = I_i^G$ and C_s.

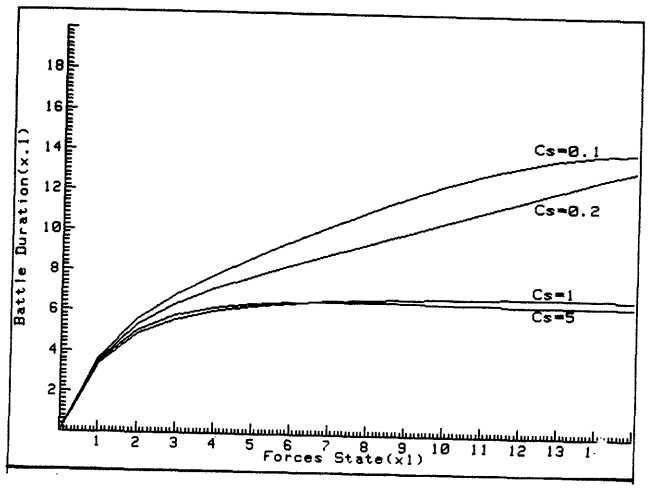


Figure 10 - D_b versus $I_i^F = I_i^G$ and C_s.

CONCLUSIONS

The Markovian modeling of C^3 processes seem to have the potential for providing a rich and comprehensive mathematical structure necessary to formulate, understand and explore the impact of various MOP's and MOE's relevant to various missions of C^3 systems. This is illustrated by simplistic first-order Markov single-unit attrition processes whose intensity functions are characterized by the counter power, identification power and their mutual counter-identification sensitivity factor for each force F and G. This model also provides a natural extension to the well-known Lanchester equations for aimed-fire, area-fire and mixed-fire. In this model, degradations similar to those occuring for area-fire result from poor identification capability and high sensitivity of the counter function to the identification function. Thus area-fire is interpreted as a correction effect mitigating aimed-fire. Our current on-going research investigations, modeling and evaluating the performance of C^3 systems in more detail, have led to significant extensions of the models described in this paper.

ACKNOWLEDGEMENT

The authors wish to acknowledge the encouragement to persue this area of research given to them by Messrs. L.D. Diedrichsen, and J.M. Dressner and by the Joint Directors of Laboratories C^3 Research and Technology Program. The authors also wish to acknowledge CPT. L. Poli, for assisting in the programming necessary to obtain the numerical results. This research was supported under U.S. Army CECOM Contract Number DAAB07-84-C-K577.

REFERENCES

1. I. Mayk, S. Rosenstark, J. Frank, "Analysis of C^3 Systems based on a Proposed Canonical Reference Model," Proceedings of the 7th MIT/ONR Workshop on C^3 Systems, pp. 45-54, June 1984.

2. J. G. Taylor, Lanchester Models of Warfare, Vols. I and II, Ketron, Inc., Military Applications Section, Operations Research Society of America, Arlington, Va. March 1983.

3. I. Rubin, " Regular Point Processes and Their Detection," IEEE Transactions on Information Theory, Vol. 18, pp. 547-557, September 1972.

4. I. Rubin, "Regular Jump Processes and Their Information Processing," IEEE Transactions on Information Theory, Vol. 20, pp. 617-624, September 1974.

5. I. Rubin, "Reduced Memory Likelihood Processing of Point Processes," IEEE Transactions on Information Theory, Vol. 20, pp. 729-738, November 1974.

ARTILLERY CONTROL ENVIRONMENT

Samuel C. Chamberlain & Virginia A. Kaste

US Army Ballistic Research Laboratory
Aberdeen Proving Grounds, MD 21005-5066

Fire support control (FSC) is a key field artillery problem area. While HELBAT 8 (Human Engineering Laboratory Battalion Artillery Test 8), a Nov '81 field exercise, was being planned to study FSC, it was recognized that field exercises and computer models alone are inadequate to study this complex problem area. Therefore, Ballistic Research Laboratory (BRL) members of the HELBAT Working Group initiated a major work effort to develop a hybrid simulator that would permit live, real-time interplay of manned FSC devices with interactive computer models in a computer-controlled, laboratory environment. A generalized simulator technology, called the Artillery Control Environment (ACE), was developed and is now being transferred to other Army agencies. ACE software has been incorporated into a joint HEL/BRL Command Post Exercise Research Facility (CPXRF) and the first FSC experiment was completed in July 83. In April 84, the facility was used to automatically collect and reduce large volumes of digital data from a major U.S. Army field exercise (the Fire Support Team Force Development Testing and Experimentation #2, FIST FDT&E II). Activities are also underway to assist other Army communities in determining how to use exported ACE technology for more cost-effective developmental testing and soldier training. It is hoped that ACE techniques and concepts will be utilized as of a new way of doing RDT&E business in the computer age - using evermore powerful commercial computers to automate the RDT&E of tactical command and control computer systems.

THE CONCEPT

Through the exploitation of newly developed interactive, operating systems (software), a real-time, multiplayer simulator technology, called ACE for Artillery Control Environment, was conceived and is now evolving. With the ACE concept, components of the fire support control ADP system can be played a number of ways: (1) devices can be emulated on low-cost, commercial video computer terminals; (2) devices or functions can be simulated in interactive computer programs; or (3) actual tactical equipment, fielded or experimental, can be accommodated through the use of the ACE Bit Box, a device that interfaces any equipment employing the TACFIRE message protocol and format to the commercial computers on which ACE can run. Conceptually, a particular ACE setup can be configured with any combination or number of these components as is needed for the desired application or the organization and operation to be played. Fire support control components that are not actively played or inputs that are external to the organization structure being studied can be represented by scenario-based, time-ordered TACFIRE messages read into ACE from predefined computer files or by a tactical equipment operator with cue cards. ACE components are interconnected by a program named "Ether", which simulates radio nets, and characterizes communications from perfect to a selected, degraded probability level of successful data communications for each net. A Master Control and Display Management Program provides for computer control of a particular experiment and permits experimenters to monitor real-time message flow on a large-screen TV or other suitable monitor or printer to instantly extract data such as processing time for a particular node (see Figure 1).

In March 1982, it was agreed that the ACE and HELBAT activities should be joined to create a research or test-bed facility with which a combination of laboratory and field exercises could be conducted. The facility, located in the newly built HEL building, uses ACE software provided by BRL and computer hardware and mock-up artillery facilities provided by HEL. Through radio links, laboratory-based exercises can include field elements such as mobile command post vehicle, howitzer, and ammunition handling test-beds (see Figure 2). This facility will not eliminate the need for live field exercises, but it can be used to perform, for example, time and motion studies of the total artillery fire support system, and alternatively, to evaluate selected individual components thereof in a total operations context. Flexibility is provided by the ability to mix simulated and real (live) players, which in the future may even include remote players in another part of the country interconnected to the facility via commercial telephone lines or the Defense Data Network. The data resulting from controlled experiments can be used to derive inputs and parameters concerning human and system performance for command and control models. The extreme versatility of this type of evolving test-bed facility is obvious, with a broad range of potential applications.

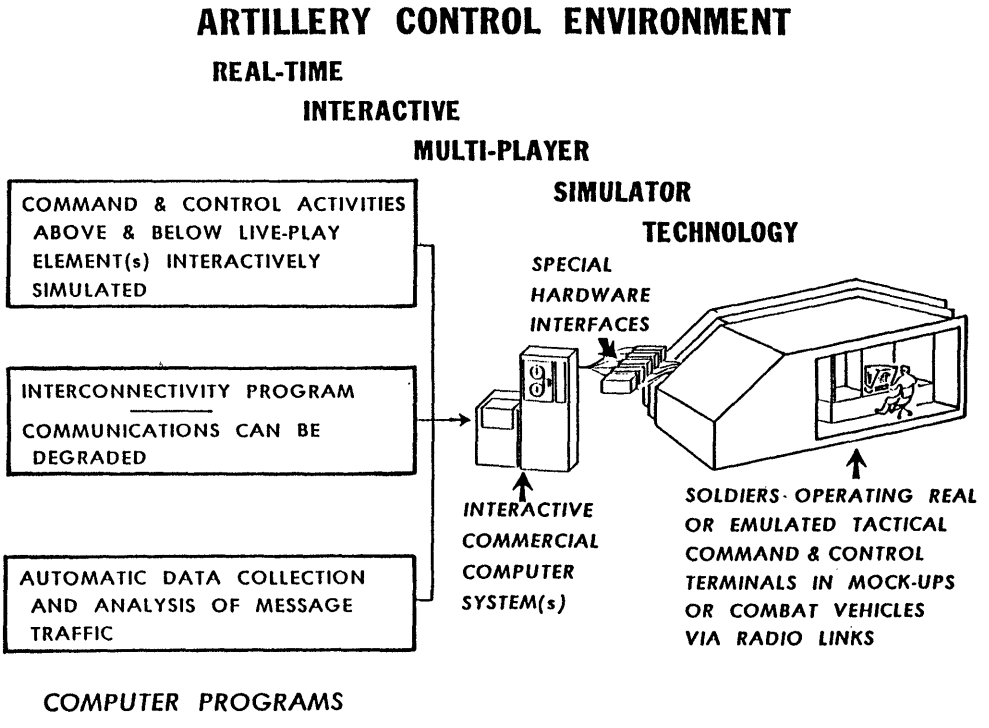


Figure 1

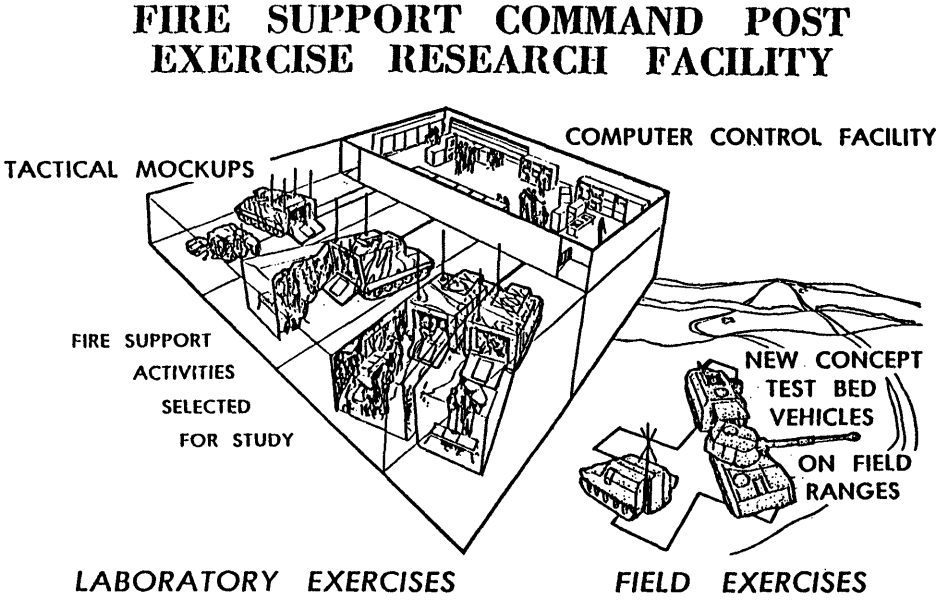


Figure 2

The experience gained from the first experiment led to the application of ACE software and hardware in the role of tactical ADP testing. At the request of the US Army Field Artillery Board, the CPXRF was used to support the FIST FDT&E II, conducted at Ft. Riley, Kansas, to evaluate issues concerning the FIST concept. Although it was originally hoped that a real-time facility could be moved to the test site, time and money constraints forced a less risky approach to be used. Data was collected at the field site by recording the analog (Frequency Shift Keyed, FSK) messages directly from radios that monitored the six digital FM radio nets. These tapes were shipped every 24 hours to the CPXRF (located at Aberdeen Proving Grounds, Maryland) where the data was digitized, time-tagged, and stored in a computer. The digitized messages were then sorted into predesignated categories: bonafide Fire Missions, Miscellaneous (not used in fire missions), and Unknown messages. An usually successful attempt was made to identify the messages in the unknown category and place them into one of the other categories. The categorized messages were finally put on standard computer tape and sent back to the Field Artillery Board (located at Ft. Sill, Oklahoma) to be used in their data reduction process. Some data reduction work was also conducted at BRL to produce "baseline" information about current Army field artillery systems for use by modelers and to serve as a measuring stick for future exercises (see Figures 4 & 5). In all, over 60 mega-bytes of data was collected at the FIST FDT&E. There were approximately 171,000 messages transmitted over six radio nets during the 320 hours of testing. To our knowledge, this is the first time such a database has been collected. Detailed information concerning this project is available in the BRL reports: *Field Artillery Digital Message Collection and Reduction Software*; BRL-IMR-822; Hartwig, Kaste, Brodeen, Hansen, Walter, Chamberlain; June 1984; and *Description of the Digital Data Collected from the FIST FDT&E II*; BRL-IMR-840; Kaste, Brodeen, Winner; February 1985.

Another CPXRF experiment is in the planning and development stage; this experiment will have three objectives designed to gained a better understanding of artillery fire control. Objective #1 will be to investigate the fire control interactions between the battery fire direction center (FDC) and its howitzers in order to gain insight into the problems associated with semi-autonomous howitzer operations. Objectives #2 and #3 focus on the responsibility of the battalion fire direction officer (FDO) to provide tactical fire control decisions, in this case, the selection of the amount and type of ammunition to expend on a target. Objective #2 is to establish production rules that can be used in a prototype expert system. Objective #3 is to investigate automatic procedures to evaluate nodal performance based upon the *quality* of decisions rather than simply service times.

The ACE and CPXRF technology is also being imported by the Field Artillery School for training applications. Plans are being finalized to build a flexible and integrated Training Development Facility at Ft. Sill that can be used to support research in training developments and operational concepts, as well as to train and evaluate soldiers on scarce, automated weapon control systems. The capability provided by the Bit Boxes, that have been produced commercially in a jointly funded buy with the National Training Center, will allow integrated field and command post exercises to be conducted over standard FM radio links. This will allow real observer teams and howitzer crews to fire live missions while the tactical command and control aspects of the fire mission are handled in a command post exercise environment within the facility. Real-world problems, that might not surface in a totally simulated environment, can be identified while new, innovative command and control approaches are wrung-out within the facility. The expansion of the Defense Data Network will eventually connect the Ft. Sill facility and the CPXRF to provide a significant communications link between the user and developer communities.

In the out-years, ACE software and the CPXRF will support a variety of research projects dealing with the applications of artificial intelligence techniques, man-machine interface studies, and systems integration concepts. It is hoped that ACE will be one of the forerunners of a new way of doing business in the computer age, using evermore powerful commercial computers to automate the RDT&E of tactical command and control computer systems.

APPLICATIONS

A major milestone in the evolution of the HEL/BRL CPX Research Facility (CPXRF) was the completion of the first true experiment, which successfully demonstrated the feasibility of using ACE technology to conduct sound statistical and automated fire support control experiments utilizing actual tactical ADP gear and their human operations. In this first experiment, a single fire support control element, the Fire Support Team (FIST) HQ with a prototype FIST Digital Message Device (DMD), was isolated and tactically loaded by interactive, semi-intelligent software that simulated the actions and reactions of subordinate and higher-echelon players and their tactical ADP gear (see Figure 3). This provided for tight, statistical control of the experiment and eliminated the need to use large numbers of personnel serving as controllers. The experiment consisted of running cells with varying degrees of fire mission request rates and communication degradation. Each message received at or sent from the FIST DMD was time-tagged, stored in a file for later analysis, and displayed in real-time on a computer terminal in the control room. A total of 72 two-hour cells were run in the experiment to collect statistical data on the effects of message intensity and communications degradation on the fire support coordination performance of a FIST HQ. Over 45,000 messages were collected during this experiment. The data reduction effort has produced information on the response time of the FIST HQ, the effect of the reception of redundant (duplicate) messages and the retransmission of failed messages, and a variety of other pertinent issues. Detailed information is available from the Defense Information Center in the BRL reports: *Fire Support Team Experiment*; BRL-MR-3422; Smith, et al; December 1984; and *Approach to Modelling the Distribution of the Message Service Times for the Fire Support Team Experiment*; BRL-MR-3461; McKaig; September 1985.

FIRST PRODUCT: A STUDY OF THE EFFECTS OF DATA COMMUNICATIONS INTENSITY AND COMMUNICATIONS DEGRADATION ON FIST HQ'S ABILITY TO PERFORM FIRE SUPPORT COORDINATION.

FIRST

CPX

RESEARCH

FACILITY

EXPERIMENT

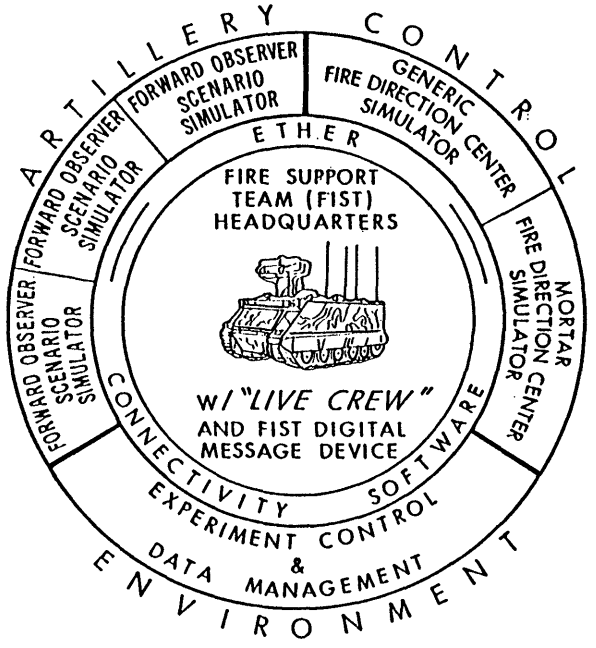


Figure 3

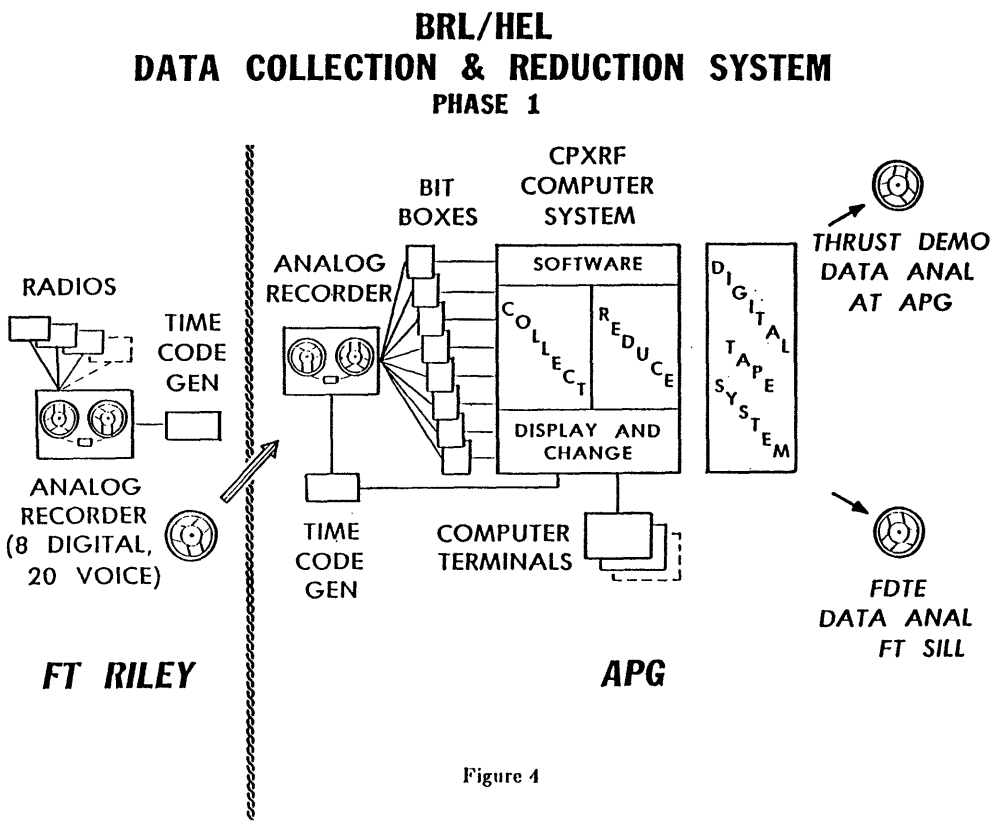


Figure 4

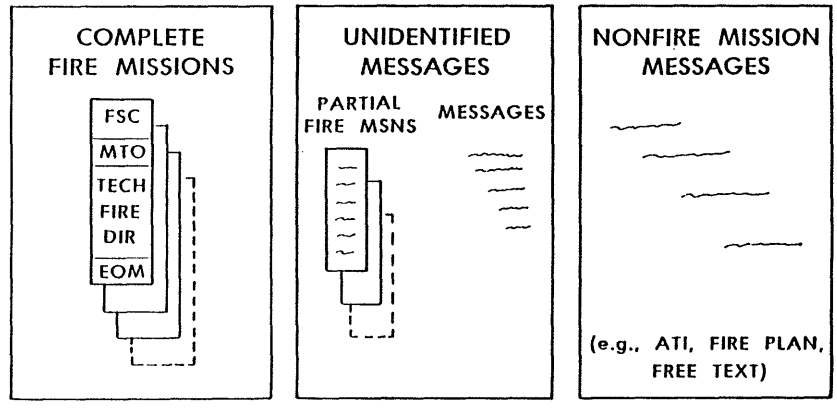


Figure 5

Analysis of The Fire Support Team Force Development Testing and Experimentation II

Virginia A. Kaste & Samuel C. Chamberlain

U.S. Army Ballistic Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

The U.S. Army Ballistic Research Laboratory (BRL) has developed a real-time interactive multi-player simulator technology known as the Artillery Control Environment (ACE). ACE evolved from the Human Engineering Laboratory Battalion Artillery Test (HELBAT) series of field exercises. By combining computer programs and live players, ACE permits realistic simulation of fire support control for any low-echelon battlefield slice.

ACE software was incorporated into a joint BRL/HEL Command Post Exercise Research Facility (CPXRF) in a successful demonstration of the feasibility of using automated techniques for fire support control experiments. Experience gained from this experiment led to applications of ACE software and hardware in tactical ADP testing.

The CPXRF was recently used to support the U.S. Army Field Artillery Board in its Force Development Testing and Experimentation (FDT&E) of the Fire Support Team (FIST) concept. The test was conducted at Fort Riley, Kansas. A new automatic data collection and reduction technique based on ACE was used. Each day, frequency shift keyed messages were tape-recorded from each of six digital FM radio nets; IRIG-B time code (received via satellite) was also recorded on the tape. The analog tapes were shipped to Aberdeen Proving Ground, where they were digitized and sorted into fire mission categories. A display showing the message traffic as it was digitized from the tapes provided Field Artillery communications experts the capability to detect communications problems, which were then corrected via telephone conversations with the testers in the field. The sorted lists of messages resulting from the data reduction at Aberdeen were written onto digital tapes and sent to the Field Artillery Board for their analyses.

This accomplishment of collection and reduction of digital field test data represents the beginning of a new approach to development, testing, and analysis of tactical ADP systems. Analysis conducted at BRL produced information about current Army field artillery systems for use by modelers and to serve as a baseline for future exercises. This paper discusses some of the analytical results.

TEST CONCEPT

The purpose of the FIST FDT&E II was to examine the 24-hour operational effectiveness of the FIST Headquarters (HQ) equipped with FIST vehicles and digital communications equipment. The test comprised two iterations of Scenario Oriented Recurring Evaluation System (Europe V) based field exercises (FEX) which involved mechanized infantry and armor elements in a series of offensive and defensive maneuvers. The first FEX lasted 120 hours and the second, 102 hours. The third FEX was a free-play force-on-force exercise.

Test Configuration

Figure 1 is a diagram of the direct support (DS) field artillery assets of a typical Division 86 Heavy Brigade. There are three digital fire direction nets that link the battalion fire direction center (bn FDC), a battery FDC, a bn fire support element (FSE) and three FIST HQs. Company fire control nets link the FIST HQs to their three forward observers (FO). The combat observation lasing team (COLT) is attached to one of the FIST HQs on a fire direction net. The symbols enclosed in dashed lines in **Figure 2** depict the elements that were played during the FIST FDT&E II. The maneuver force was a mechanized infantry battalion task force composed of two mechanized infantry companies and one armor company. The three maneuver companies were task organized into three company-teams, each with two mechanized infantry platoons and one armor platoon. In reality, armor FISTs do not have FO parties; however, for the purpose of this test each FIST HQ controlled two FO parties (located in the mechanized infantry platoon HQ) and also received voice calls for fire from the armor platoon leader. The task force fire support assets included a Division 86 heavy mortar platoon (4.2"), two DS artillery platoons (155mm, M109A2), two general support (GS) artillery platoons (8 inch, M110A1), and on occasion, attack helicopter (AH-1S Cobras with HELLFIRE missiles) and Close Air (A-10's with Pave Penny) support. A DS artillery battalion from the 1st Infantry Division Artillery also provided a battalion FDC and a maneuver battalion fire support element (FSE).

Communication Nets

CFC Nets 1, 2 and 3 (see **Figure 2**) were the digital communication links between each of the three FIST HQ using FIST Digital Message Devices (DMD) and their two forward observers using FO DMDs. Using a FO DMD, the Mortar FDC communicated with the FIST HQ via the maneuver battalion Mortar Fire Direction (MFD) Net. At the same time, the FIST HQ communicated with the FSE via voice. Fire Direction (FD) Net 2 linked the bn FDC (using TACFIRE) to the two GS artillery platoon FDCs (using BCS) and FD Net 1 linked the three FIST HQ, FSE, bn FDC, and the two DS artillery platoon FDCs (using BCS). The FD Net 1 was fully loaded in this test; however, since only one-third of the Field Artillery (FA) assets of the maneuver brigade were played, there was only

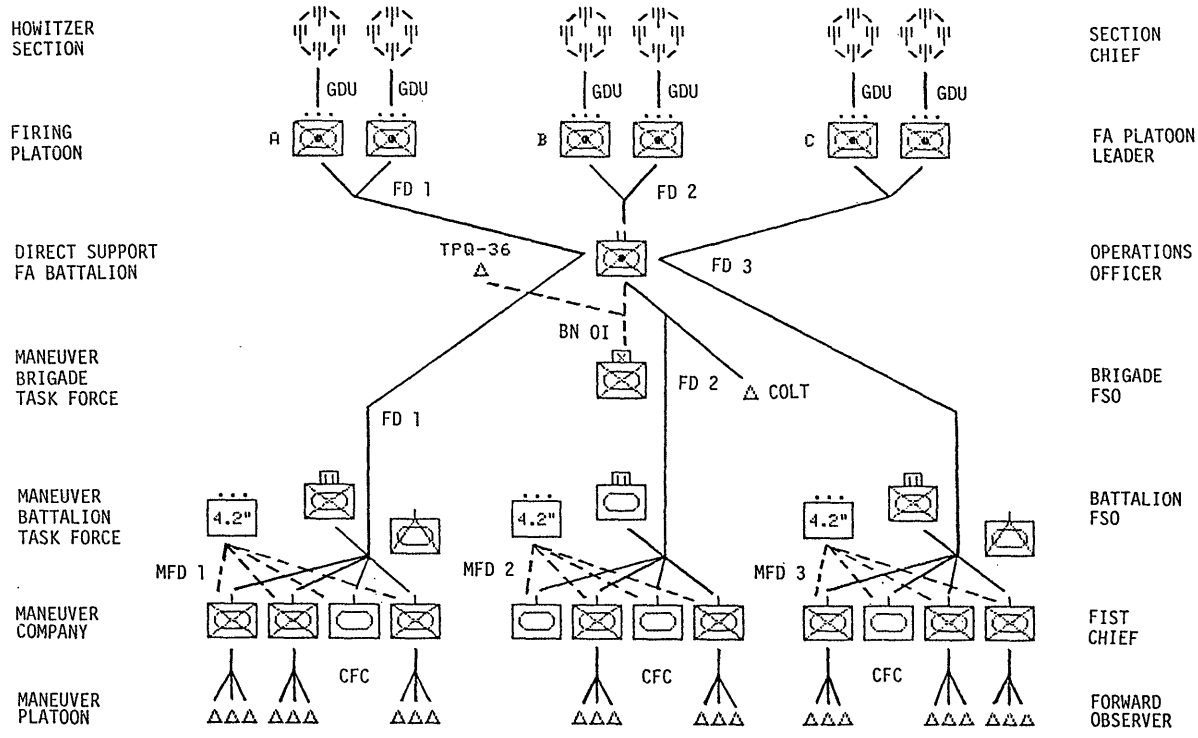


Figure 1. Division '86 Heavy Brigade & DS Field Artillery Assets

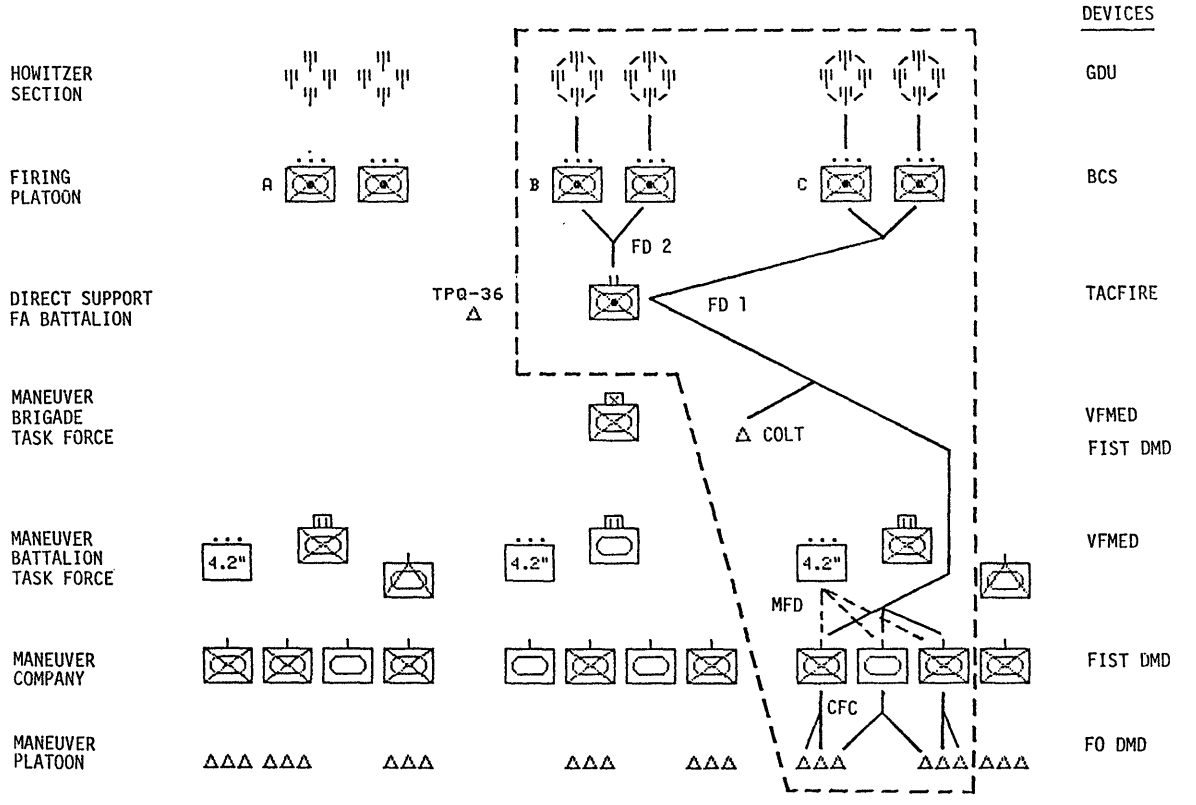


Figure 2. Division '86 Brigade Elements Played at the FIST FDT&E II

one-third the realistic loading on TACFIRE. Discussions of the results of net utilization will be based primarily on FD Net 1.

DATA COLLECTED

Three hundred forty-four hours of tape-recorded data were received from Fort Riley. **Table 1** shows the number of digitized messages resulting from 322 hours of tapes played through the data collection and reduction programs. The remaining 22 hours of tapes contained communications collected from non-controlled parts of the test. Of these, four hours of communications were from FEX I and eighteen from FEX III.

Unintelligible messages were messages that did not conform to the basic TACFIRE message format as determined by the program COLLECT. Such messages fell into three categories: 1) not entirely received by the recording site, 2) garbled in transmission, 3) artifact of the Bit Boxes receiving noise. All transmitters on the CFC nets are less powerful than those on the FD nets; hence, the percent of unintelligible messages is larger for the CFC nets than for the FD nets. The mortar FDC was given an FO DMD (with AN/PRC-77 1.5 - 2 watt transmitter) and was located a long distance from the data collection antenna, hence the large percentage of unintelligible messages on the MFD net.

NET UTILIZATION

Platoon forward observers communicate to their FIST HQ via a CFC net. In turn, each FIST HQ communicates with the artillery bn FDC (where TACFIRE is located) on the FD Net 1. The battery FDC also communicates via FD Net 1, but there is only one-way communication between it and the FIST HQ. In short, FD Net 1 was fully loaded with the appropriate number of players. The discussion in this section deals with net utilization for FD Net 1.

"Utilization" of a radio network is a function of the number of messages passing through, the lengths of the messages and the radio warm-up time (also known as preamble time). Analysis of net utilization revealed that the FD net was occasionally almost 50% loaded (see **Figure 3**). This is

TABLE 1. DIGITAL MESSAGE TRAFFIC COLLECTED

NETS	GOOD MESSAGES	UNINTELLIGIBLE MESSAGES	PERCENT UNINTELLIGIBLE	TOTAL MESSAGES
CFC 1	10,103	746	7	10,849
CFC 2	9,268	1,341	13	10,609
CFC 3	11,839	1,398	11	13,237
FD 1	97,880	2,631	3	100,511
FD 2	24,552	1,430	6	25,982
MFD	5,780	3,651	39	9,431
TOTAL	159,422	11,197	7	170,619

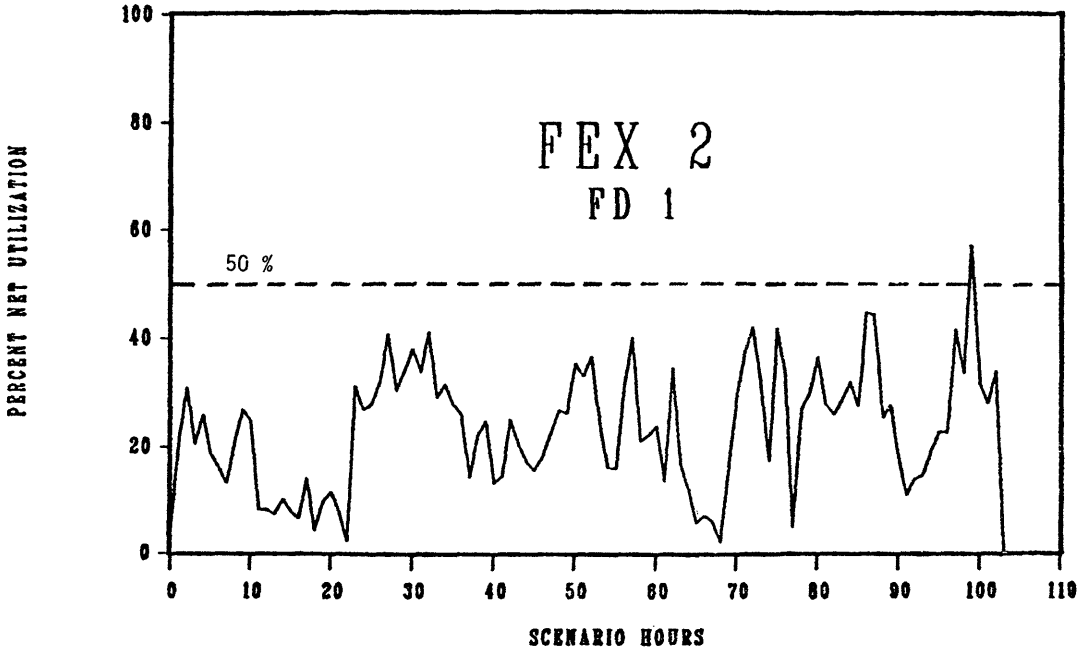


Figure 3

a serious problem because as net utilization increases, the chances of net contention, detection, interception, and destruction also increase. Therefore, net usage should be kept to a minimum. The two major contributors to the high net utilization were the lengths of variable format messages and preamble times. The Tactical Fire Direction System (TACFIRE) generates variable format (VF) messages and much shorter fixed format (FF) messages. If VF messages could be changed to encoded messages of shorter length, then net usage would decrease. This, however, is not an immediate solution due to the time and cost of modifying software for tactical equipment.

A message transmission always includes a preamble, synch characters and the message itself. The preamble time used during the FIST FDT&E II was 2.1 seconds, the default warm-up time for TACFIRE radio operations. The time necessary to transmit a FF message without the preamble is 0.51 seconds at 1200 bits per second, and the time necessary to transmit a VF message of average length without the preamble is approximately 2.2 seconds at 1200 bits per second. In general, the time required to transmit the preamble is longer than the time necessary to transmit the information of a message.

Figure 4 and **5** compare net utilization with and without the preamble, the average number of messages per hour on the FD net was used. When a 2.1 second preamble was used, the average net usage was 23.3%; however, with no preamble the average net usage dropped to 5.9%. Preamble time is the major contributor to high net usage; reducing it is an important goal.

TIME REQUIRED TO SERVICE A MESSAGE

The following section presents service time information for the FIST HQ equipped with a FIST DMD, the artillery bn FDC with TACFIRE and the battery FDC with a BCS. Service time for each of these players is defined to be the time from the confirmed receipt of a message (i.e., the automatic transmission of an acknowledgement message) until a response to that message is first transmitted. Thus, service time includes not only the time it takes a player to process a given message, but also the time the message is queued in the digital device. If a message is deleted from the message queue by the DMD operator, then service time for that message does not exist (in terms of digital computation). In order to interpret nodal service times, one must first understand the message traffic flow involved in initiating a fire mission.

Figure 6 outlines the message traffic flow for a FO initiated fire mission. Numbers 1-6 enclosed in parentheses on the figure indicate the sequence. A fire mission begins when an FO sends a fire request (FR) to the FIST HQ for review. The FIST HQ's responsibilities include checking the FO's target coordinates for range and possible previous mission assignment, and determining the need for a possible readjustment of the FO's assigned priority for the target. Upon completion of this review, the FIST HQ chooses one of following three options: (1) deny the fire request, (2) authorize mortar fire on the target, or (3) forward the fire request to the bn FDC. If a fire request is forwarded to the bn FDC then TACFIRE processes the fire request by transmitting a "Fire Mission;Fire Command" (FM;FC) message to the battery FDC. After sending the FM;FC, TACFIRE dispatches a "Message to Observer" (MTO) message to the FIST HQ. This MTO, supplying the FIST DMD with the essential information it needs for the ensuing events, is an essential message in fire mission processing. The FIST DMD, in turn, is responsible for passing TACFIRE's MTO along to the FO. After receiving TACFIRE's FM;FC, the battery FDC determines the additional ballistic information required for each gun, and transmits that information via wire to the howitzer sections' gun display units (GDUs). Thus, battery service time includes not only BCS service time, but also the howitzer section's processing time for loading and laying the guns. The battery FDC should notify FIST HQ with a SHOT message when the guns fire on the target, and this message is forwarded to the FOs by FIST HQ. However, in some instances during the FDT&E, the battery FDC incurred significant delays in sending the SHOT message, and this affected the service time.

Service times for the FIST HQ, battalion FDC and the battery will be presented in histograms, and for each histogram the mode and median will be given for each entire sample.

FIST HQ Service Time

The FIST HQ to processes FO initiated fire requests with the FIST DMD in two modes of operation, Auto/Auto and Review/Auto (further reference to Auto/Auto is Auto and to Review/Auto is Review). When the FIST DMD is in Auto mode, all messages are transmitted directly through the FIST DMD. However, when the FIST DMD is in Review mode, all messages except subsequent mission messages (such as SUBQ ADJ, EOM&SURV) are first inspected before they are transmitted to their final destination.

Figures 7 and **8** present FIST HQ service times for fire requests with the FIST DMD in the Auto mode and Review mode, respectively. A few observations for each FIST DMD mode were greater than one hundred and fifty seconds and are not shown in these figures. A detailed investigation of the fire requests in Review mode has not yet revealed the cause of the apparent bimodal distribution of service times.

With the FIST DMD in Auto mode, it was expected that FIST HQ service times would range from zero to two seconds. The time code generator had a resolution to the nearest second. Thus, a zero second service time really indicates a service time of less than one second. Service times in the range of three to six seconds indicate that the FIST DMD was checking to see if the net was busy with either voice or digital traffic. If the net was busy, message transmission was delayed until the net monitor indicated that the net was no longer busy.

At first glance, the FIST HQ service times for fire requests with the FIST DMD in Auto mode (see **Figure 7**) seem to suggest these expected time ranges (0-2 secs and 3-6 secs) might be conservative. However, an examination of data collected manually at the FIST HQ during the FEXs has revealed that several DMDs were actually in a mode other than the mode reported for a particular control cell of the FDT&E.

Figure 8 indicate that there were FR messages processed by the FIST HQ in two seconds or less when the FIST DMD was in Review mode. This processing time is rather small considering the FIST DMD operator has to call up the message from the message queue, read it, process it and then transmit it. Again, further investigations revealed that not

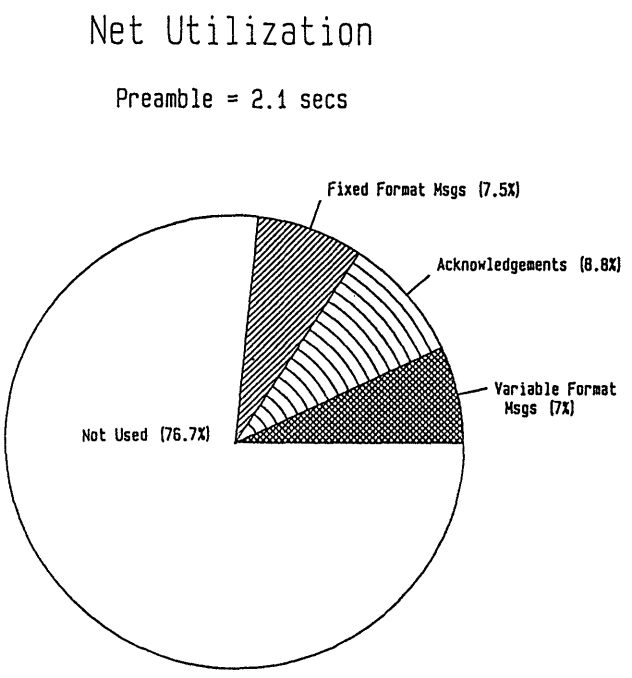


Figure 4

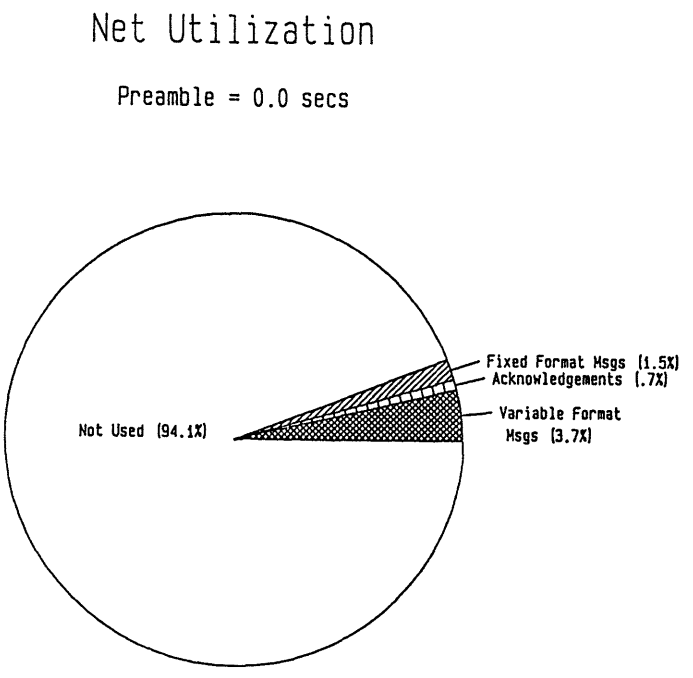


Figure 5

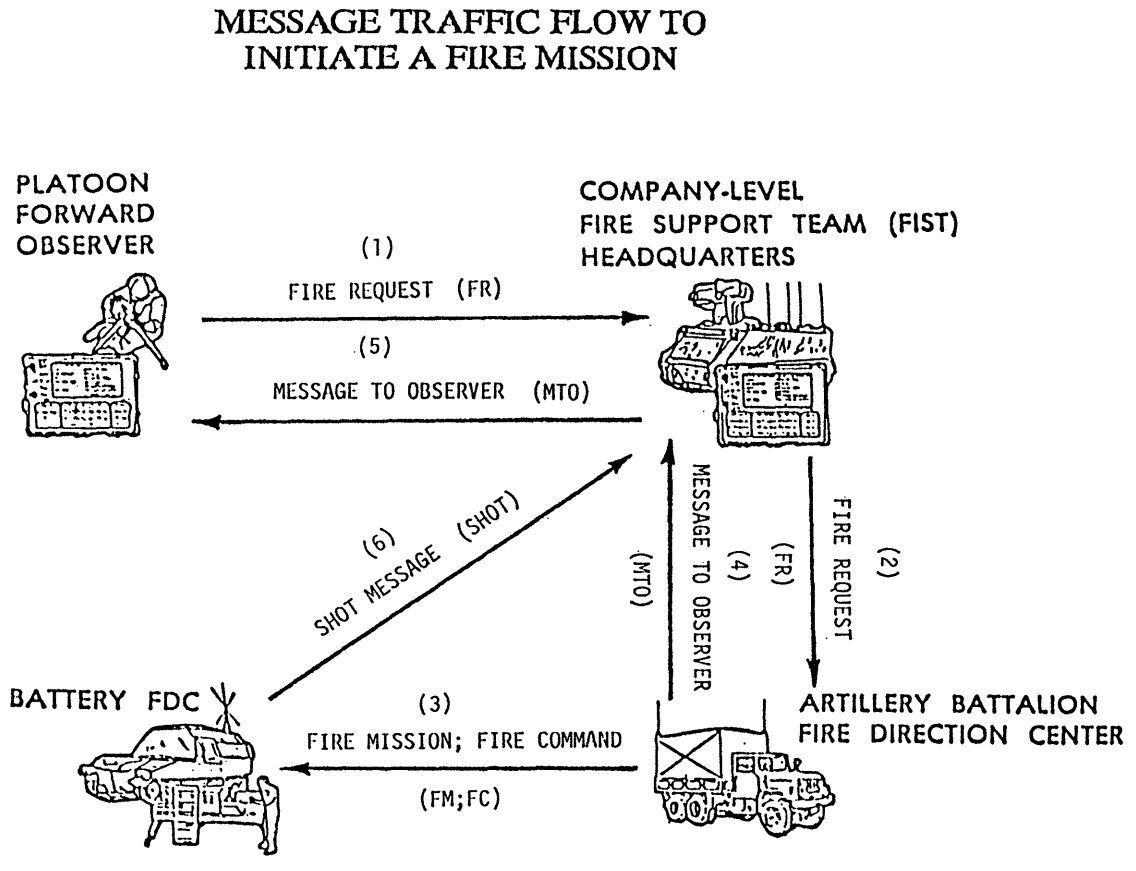


Figure 6

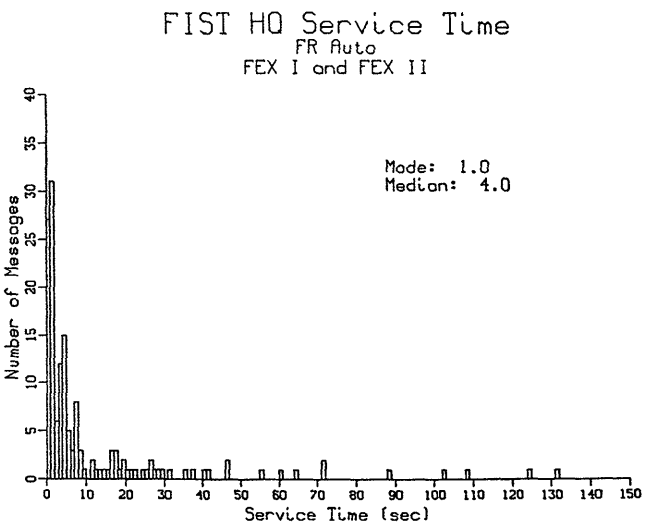


Figure 7

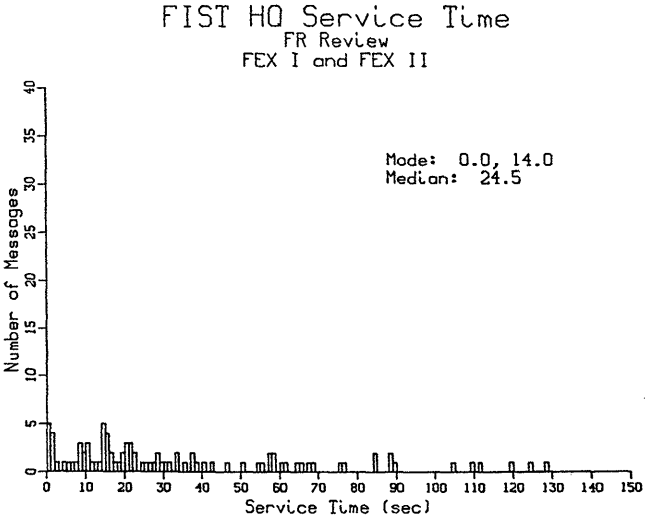


Figure 8

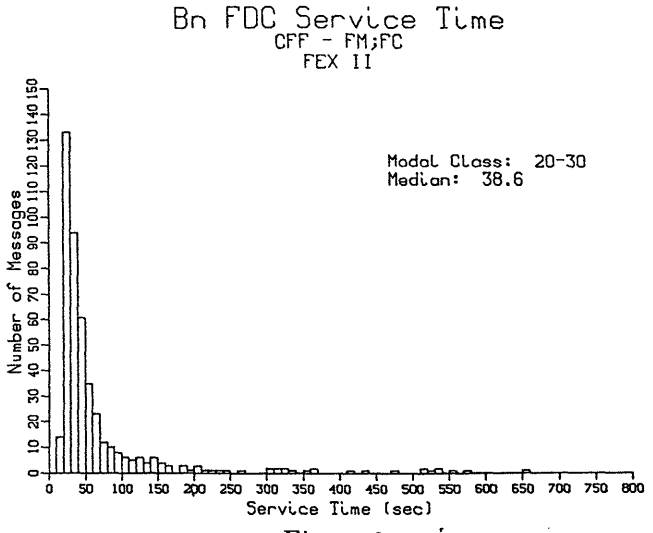


Figure 9

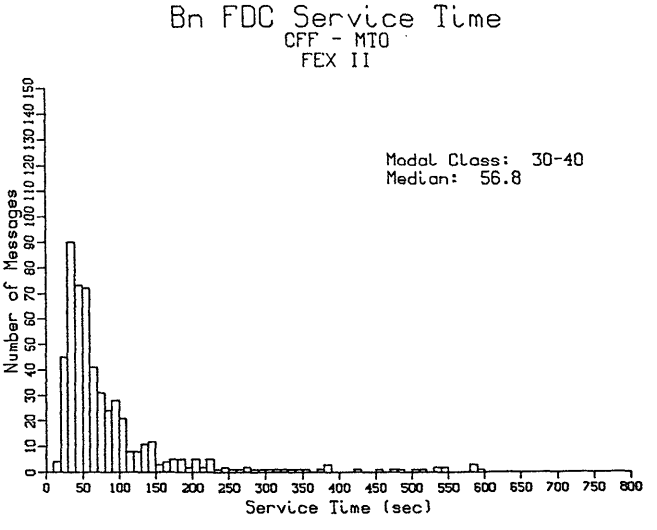


Figure 10

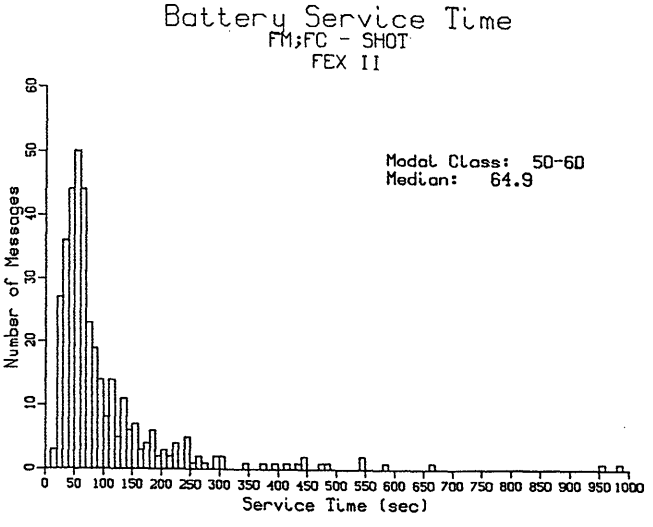


Figure 11

all FIST DMDs were in the properly assigned control cell mode.

As expected, the mode of the FIST DMD was a significant source of variability. Messages required a longer time to process in Review mode (24.5 secs) than in Auto mode (4.0 secs). This, however, does not indicate that Auto mode is better than Review mode; Review service time reflects FIST HQ fire control intervention, absent from Auto processing.

Bn FDC Service Time

During the FDT&E, FO initiated fire missions were not as prevalent as FIST initiated fire missions. Since the bn FDC processes all fire requests approved or initiated by the FIST HQ, the sample sizes for the FIST HQs' service times for fire requests are small in comparison to the sample sizes for the bn FDC service times.

Figures 9 and **10** present bn FDC service times computed for FEX II. Bn FDC service times are presented for both the call for fire (CFF, e.g., fire request subsequent adjust) to the FM;FC, and for the CFF to the MTO.

As anticipated, the median for the CFF to the MTO service times (58.5 secs) is larger than the median for the CFF to FM;FC service times (39.5 secs). Recalling that TACFIRE dispatches the FM;FC before it dispatches the MTO, the aforementioned differences are a direct consequence of the manner in which TACFIRE processes a fire request.

Battery Service Time

Figure 11 presents battery service times. Battery service time is calculated as the time difference from the battery FDC's receipt of the FM;FC until the battery FDC's first transmission of a SHOT message to the FIST HQ. Thus, battery service time includes the time required by the howitzer section to lay and load the guns. The median battery service time for FEX II is 64.9 secs.

CONCLUSIONS

The collection of digital data provided a means to compute the amount of time a communication net was used. It was shown that the fully loaded FD Net 1 was used on the average 20 minutes of each hour. The consequence of such high net usage could be quite serious; net usage should be kept to a minimum.

FIST HQ service times, as expected, indicate a significant time difference between FIST DMD Auto and Review modes, since messages processed in Auto mode are not examined for content by the FIST HQ. Doctrine writers and commanders will have to make decisions concerning which messages require this tactical supervision, and equipment should be sufficiently flexible to accommodate changing requirements.

During the FIST FDT&E II, FIST DMD settings were occasionally inadequately controlled. Information concerning FIST HQ service times in this report is subject to improvement based on tedious compilation of both manually collected field data and digital data.

At only one-third loading, the bn FDC required 40 - 45 seconds (median values) to process a CFF and produce a FM;FC to send to the BCS. Slightly over a minute was needed for the bn FDC to send an MTO to the the FIST HQ after a CFF was received. However, if TACFIRE was fully loaded, the time to service a CFF would be at least at large as the findings in this report. An experiment could be conducted to reveal how well TACFIRE functions under full communications loading operations.

It has been demonstrated that the ACE technology can be used to collect and reduce digital data from a field exercise and can offer field testers timely feedback, data in a uniform format and economical data reduction. This effort to collect and reduce field test data has also led to a start of a unique data base of field test data for building and evaluating fire support control computer models.

ASSESSMENT OF TIMELINESS IN COMMAND AND CONTROL*

Philippe H. Cothier
Service Technique Des Telecommunications et Equipements Aeronautics, Paris, FRANCE

Alexander H. Levis
Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139

ABSTRACT

A methodology for assessing the timeliness of Command, Control and Communication (C^3) systems is developed. The notion of the window of opportunity is shown to be fundamental in the analysis; system effectiveness is evaluated by considering the management of time available within the window. The methodology is based on comparing the properties of the system and the mission requirements, expressed as loci in a commensurate attribute space. Trade-offs between uncertainty reduction and response speed, or response speed and response quality can be investigated quantitatively. Comparison of different military doctrines, as well as different options for a decisionmaker confronted with time constraints is also made possible. The assessment of timeliness of an existing C^3 fire support system is presented to illustrate the methodology. Two basic fire doctrines are assessed and compared within this context.

1. INTRODUCTION: THE CONCEPT OF TIMELINESS

Time plays a fundamental role in most Command, Control and Communication (C^3) systems. Improvements in weapon system technology, higher capacity and speed in the transmission of data, combined with an increasing complexity of the battlefield, impose severe time constraints on both the hardware and the human decisionmakers. It is necessary then to develop methodologies for assessing C^3 systems that take into account time. Time has always been of crucial importance in combat; furthermore, it differs from any other attribute of a C^3 system. This uniqueness, combined with the growing concern of system designers, has motivated the study of time in C^3 systems explicitly.

As Lawson (1981) relates, "in a typical discussion of Command and Control, it is taken as axiomatic that the information presented to the commander must be 'timely' as well as accurate, complete, etc.,... Little or nothing is said about how timely is timely enough; nor is any yardstick given by which to measure 'timeliness'. Rather, the clear implication is that all would be well if only communications and computers were 'faster'. In addition, this attention to rates (e.g. information processing rates, rate of fire, etc.,...) in which time only appears in the denominator, has led to a preoccupation with the performance characteristics of the component parts of a C^3 system. It does not provide any means of comparing the effect of an increase in one 'rate' with that of an increase in some other rate".

*This work was carried out at the MIT Laboratory for Information and Decision Systems with partial support by U. S. Army Research Institute under Contract No. MDA903-83-C-0196 and the Naval Electronic Systems Command under Contract No. N00039-83-C-0466.

In this paper, a methodology for assessing the effectiveness of C^3 systems by directly taking into account the issue of timeliness is proposed. The methodological framework is the one first proposed by Dersin and Levis (1981, 1982) and then applied to C^3 systems by Bouthonnier and Levis (1984). The key idea is to relate the performance of a system to the mission it has to fulfill. One of the main advantages of this methodology in the case of the assessment of timeliness, is that it allows comparison of the effectiveness of different doctrines used with the same system. From the insights that the analysis yields, conclusions can be drawn not only for the design of C^3 systems, but also for their integration in the military doctrine.

The aspects that time can take in a warfare environment are numerous. The most important ones, whose subtleties the assessment methodology should be able to embed and to exhibit, follow.

System response time. It characterizes the time delay between the moment when the C^3 system receives a stimulus and the moment it can deliver a response. It is the sum of all the time delays at every level of the process.

Tempo of operations. In most military situations, rates are used to express the important quantities, e.g., rounds per minute, miles per hour. The term in common usage for the operating rate of a C^3 system is its "tempo". Lawson (1981) defines it as the number of actions per unit of time which the system is executing and states, further, that "the tempo tells us how complex an environment the system can handle (i.e., its bandwidth) while the response time tells us when it responds in time (i.e., the phase delay in the system)".

When a C^3 system initially receives a stimulus (e.g., a blip on an air defense radar), there is a great deal of uncertainty. The decisionmaker cannot take any action until this uncertainty is reduced below an acceptable threshold. Such a reduction takes time and effort. This presents the first trade-off: the more time is spent to reduce the uncertainty, the longer the response time, but the more adequate the response.

Two types of uncertainty can be distinguished: The first one, which can be called interscenario refers to what the commander is confronted with when he tries to identify what scenario is actually taking place (e.g., an enemy attack as opposed to a mere reconnaissance mission). The second one, which can be called intrascenario refers to the uncertainty within the scenario itself. The issue is to estimate the parameters of this scenario, such as the number of the enemy forces, their velocities or, the intensity of the attack.

For each type of stimulus, the decisionmaker has

to choose among a set of options which one to implement as a response. Not to do anything (underreaction) is also an option. These options can be ranked according to two criteria: their desirability and the time required for implementation. A given option may take a longer time to be implemented but with a more desirable outcome. The decisionmaker must take into account these aspects, and an enhanced methodology for assessing timeliness should be able to express the notion of quality of option.

These notions depend on what is actually taking place, i.e., the scenario. The event that stimulates the C^3 system is only the partial perception by the system of a global scenario. Different scenarios can be perceived through identical events and the system is confronted with uncertainty. Once the scenario is identified with enough certainty, then an option must be selected. Some options are quite appropriate for certain scenarios while some others are completely irrelevant. It appears that any assessment of a C^3 system must consider the crucial role of the scenario: to each scenario corresponds an evaluation of the effectiveness of the system. Finally, these different measures can be merged into an overall measure of effectiveness for a given range of possible scenarios.

Timeliness is a concept that embeds all the above notions. Timeliness appears to be closely related to the notion of time interval, the so-called window of opportunity. There are basically two types of windows: one characterizes the system response capabilities, while the other expresses the requirements of the mission the system is expected to fulfill. Once the system has received a stimulus, no response can be delivered before some amount of time has elapsed. The lower bound is defined by the shortest response time possible. On the other side, there may be a latest response time after which no response can be implemented. The time interval between these two boundaries constitutes the window of opportunity for the system capabilities: (t^*, t^{**}).

Any response to the stimulus must come in time in order to be effective. There comes a moment when any response is preempted: this defines the upper boundary for the response time. A lower boundary can also be defined: for example a carrier may have to wait until a submarine enters the territorial seas before taking any course of action. The time interval between these two boundaries constitutes the window of opportunity for the mission requirements: (θ^*, θ^{**}).

When the two windows are superimposed, different configurations can be sketched for the residual window of opportunity (Figure 1).

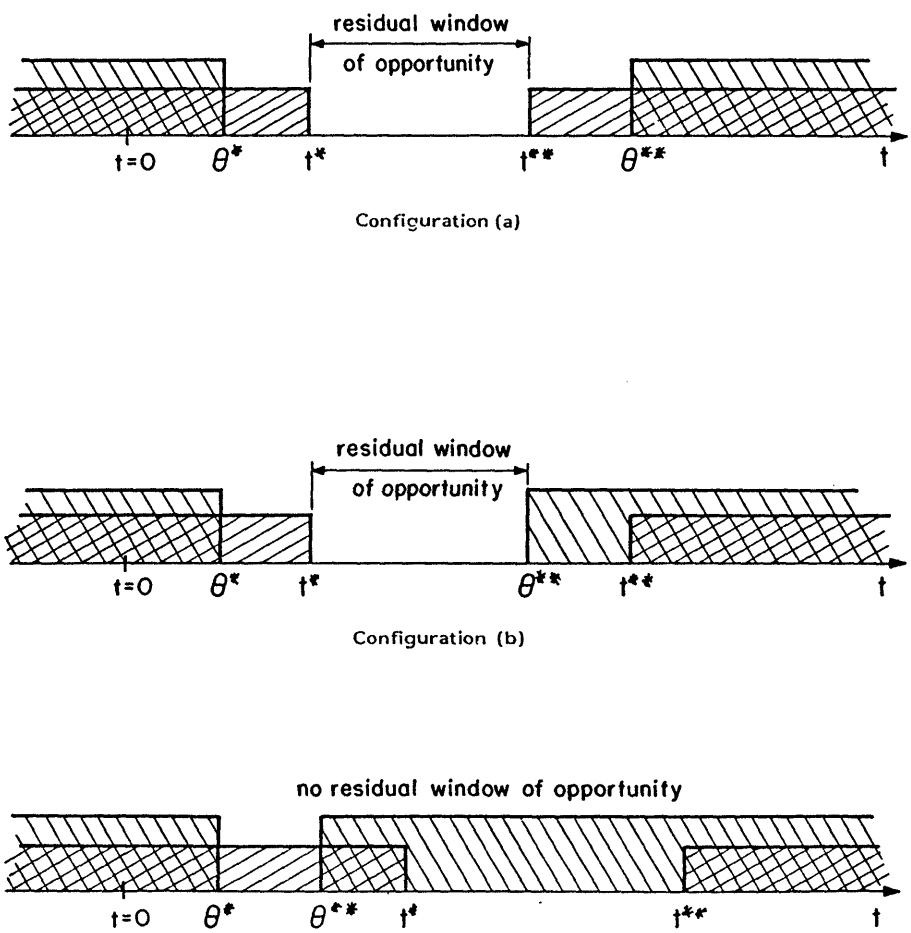


Figure 1. Different Configurations for the Windows of Opportunity

However, a measure based only on the window of opportunity is not satisfactory. While, it appears that timeliness is intrinsically related to the notion of a time interval, a time interval is not sufficient to convey the concept of timeliness; one must also consider the way this time is employed, which depends on the actual time rather than the time interval. Thus timeliness refers to the quality of time management within a given window of opportunity. In that sense, it appears that a measure of effectiveness based upon this time management can be an effective measure of timeliness. Therefore, in assessing the timeliness of a C^3 system, one should consider not only the C^3 system, but also the doctrine that is used, as well as the options from which the decisionmaker can choose; the time available and its management depend on the consideration of systems, doctrine and options. The better the effectiveness of the combination, the more timely the C^3 system.

2. SYSTEM EFFECTIVENESS ANALYSIS METHODOLOGY

The analytical aspects of the methodology described in this chapter address mainly the relationships between hardware characteristics, system structure, and standard operating procedures (SOP) to system performance (Dersin and Levis, 1981, 1982). A detailed description of the methodology has been presented in Bouthonnier and Levis (1984).

The methodology is based on six concepts: system, commander's needs for the mission, context, primitives, attributes, and measures of effectiveness. The first three describe the problem, while the last three define the key quantities in the analytical formulation of the problem.

The system consists of components, their interconnection and a set of standard operating procedures. A communication network or a fire support system are typical systems.

The commander's needs are derived from a set of mission objectives and tasks that the commander would like to accomplish. Their description must be as explicit and specific as possible so that they can be modeled analytically. For example, a requirement such as "to protect headquarters" is too broad, while a more useful specification would be "to prevent the shelling of battalion headquarters with a satisfactory level of confidence".

The context denotes the set of conditions and assumptions, i.e., the environment, within which the system operates. A fire support system operating in an urban area or in the mountains or in the desert define typical C^3 system environments.

Primitives are the variables and parameters that describe the system and the commander's requirements. For example, in the case of a fire support system, system primitives may include parameters describing the detection equipment, computational time delays, inaccuracy in a cannon battery, kill radius of the munition, and failure probabilities associated with the components, to name but a few. Primitives of the tasks may be the military pressure (see Lawson, 1980), the tempo of operations and the size of the engagement. Let the system primitives be denoted by the set $\{x_i\}$ and the mission primitives by the set $\{y_j\}$.

Attributes are quantities that describe system properties or commander's requirements. At times, they are refered to as Measures of Performance (MOPs). System attributes for a command and control system may include reliability, survivability, cost, size of the window of opportunity, and kill probability. The commander's requirements may be expressed by the same quantities as the system attributes, e.g., minimum reliability or survivability, maximum cost, or minimum kill probability. The system attributes are denoted by the $\{A_s\}$ and the requirements by $\{A_r\}$.

Measures of Effectiveness (MOEs) are quantities that result from the comparison of the system attributes and commander's requirements. They reflect the extent to which the system meets the requirements.

The seven steps of the methodology and their interrelationships are shown schematically in Figure 2. The diagram emphasizes that the system and the commander's requirements must be modeled and analyzed independently, but in a common context. The system capabilities should be determined independently of the commander's requirements and the commander's requirements should be derived without considering the system to be assessed. Otherwise, the assessment is biased. The steps of the methodology, as it applies to the assessment of timeliness, will be described through application to a tactical fire direction system.

3. ASSESSMENT OF TIMELINESS

The issues discussed in the introduction will be illustrated by applying the methodology to a hypothetical, but realistic, Army fire support system (Cothier, 1984).

One can isolate three main elements in the fire support system at the battalion level: the forward observer, the battalion fire direction center and the field artillery cannon battery. The system can include several forward observers and several batteries connected to the same central battalion computer.

The Forward Observer (FO) is the part of the system that receives the initial stimulus by detecting an enemy threat. The FO is equipped with vehicle position determining equipment and a laser rangefinder. The FO is also equipped with the Digital Message Device (DMD). The FO uses the DMD to communicate estimates of the position and velocity of the target and requests for fire to the battalion computer.

The Battalion Fire Direction Center (BN FDC) is provided with a central computer. Digital communication over any standard Army communication means (radio or wire) provides for input of data into the computer center and for the return of the results. Forward observers and firing batteries are provided with remote terminal equipment to obtain data from the central computer.

The Battery Display Unit (BDU) is the cannon battery's link with the C^3 system. The BDU assists execution of fire plans by receiving and printing firing data for each target that the battery will fire.

While this is the basic configuration, additional equipment is maintained in parallel to augment the basic system.

- Voice communication links can be added in parallel with the digital links, for instance between the battalion fire direction center and the cannon battery. Voice communication is slower, more vulnerable, but still very useful, if the digital link fails.

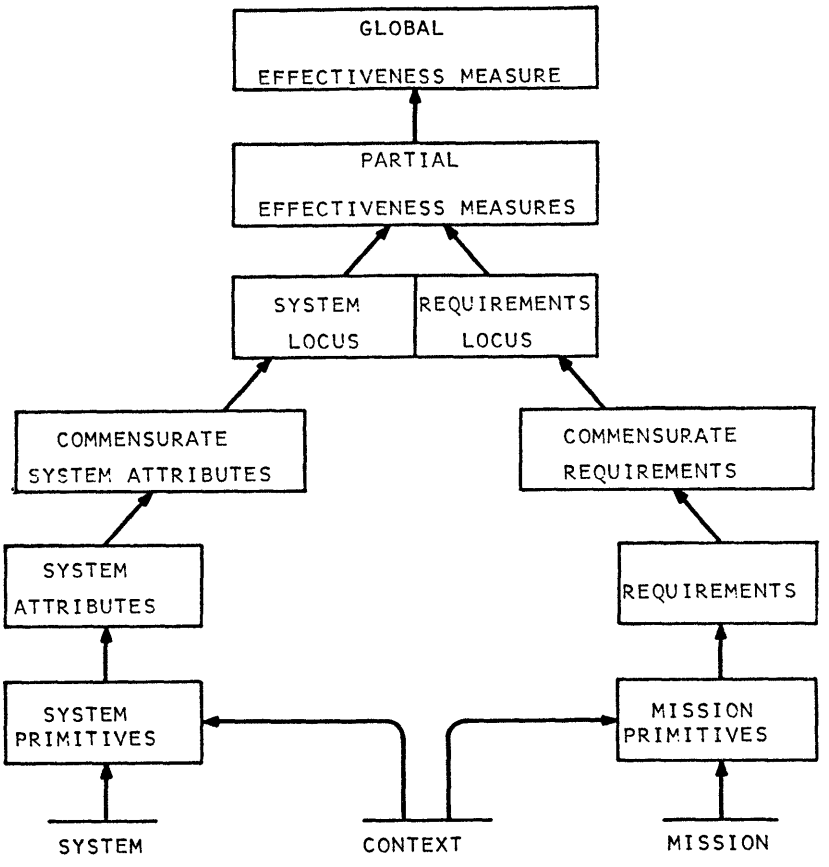


Figure 2. The Methodology for System Effectiveness Analysis

- If the fire support system computer become fails at the battalion level, the battery has the capability to do the firing computations locally. This alternative is slower, though.

A representation of the system that will be analyzed is shown in Figure 3. Seven links are shown. Nodes are not subject to failure; only links are. A voice link is in parallel with the digital link between the battalion fire direction center and battery B.

If the BN FDC computer does not work, the target estimates from the FO can be sent to battery B through voice communication (the BN FDC acts as a simple relay). The battery crew can then compute the firing data manually.

- In the case where the firing data are computed at the BN FDC level and transmitted by voice communication to battery B, neither the BDU nor the manual technique have to be used. The voice communication of the firing data reaches directly the firing platform of the battery.

In order to assess properly the effectiveness of this system it is necessary to specify the context in which it operates as well as the scenario.

The context and scenario that will be considered are shown in Figure 4. Some vital point of the blue forces (i.e., headquarters) is situated at the end of a valley. A road along this valley leads to these headquarters. The topography of the area is perfectly known by the blue forces, and the road is the only access to the blue camp. A fire support battalion including one forward observer FO, one battalion fire direction center BN FDC and two batteries B_1 and B_2, have been positioned to protect this access. This battalion is equipped with the fire support system (Fig. 3). The batteries cannot see the road; they shoot according to the firing directions that are computed on the basis of the observer's estimates.

An enemy tank (threat) appears in the area of detection of the forward observer. It is moving on the road towards the blue forces with hostile intentions. The mission of the fire support battalion is to prevent the attack on the blue headquarters by destroying or incapacitating the threat.

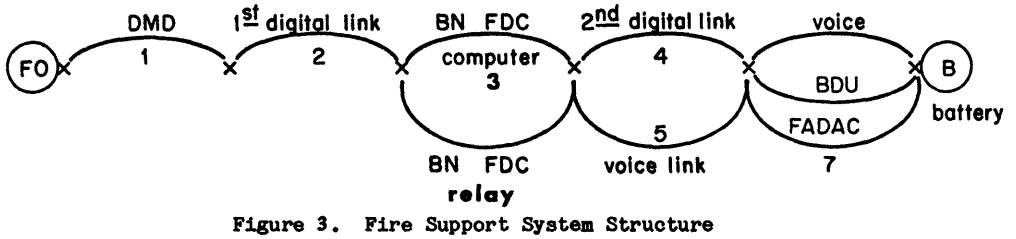


Figure 3. Fire Support System Structure

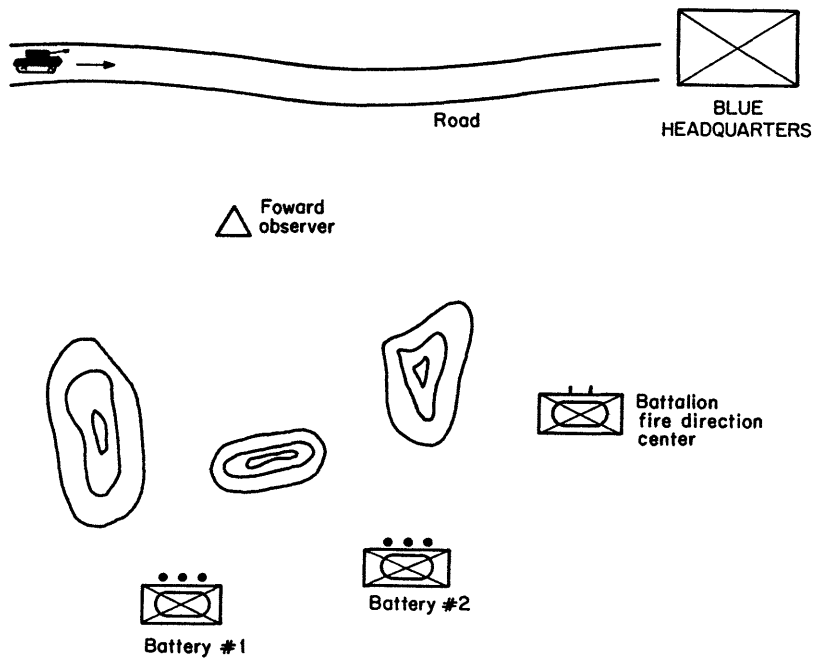


Figure 4. Context for Operations

It is assumed that the threat cannot attack the fire support battalion directly; the only countermeasure that will be considered is the jamming of the communications by the enemy. It is also assumed that the threat will pursue its attack, even after it is fired upon. It will try to carry out its own offensive mission, as if it encountered no reaction from the blue forces.

Definition of Attributes

The window of opportunity for the system response capabilities is defined by the ordered pair of attributes $(t^{**}, \Delta t)$, where t^{**} is the latest time at which the target can be destroyed, and Δt is the width of the window.

In order to characterize the ability of the system to destroy or incapacitate the target, the third attribute is the overall kill probability (OKP). Choosing such a quantity as an attribute raises a very interesting point in the system effectiveness analysis methodology. Indeed, the OKP can be considered as an attribute (an MOP) since it is a function of the system characteristics (hardware and procedure), but it is also a measure (MOE) in itself since it evaluates the destructive capabilities of the system. Such a duality can be used advantageously, because the mission requirements can be expressed fairly simply in terms of such a measure/attribute.

On the system side, the third attribute OKP is computed on the basis of the system primitives and the first two attributes t^{**} and Δt. On the mission side, the fire support battalion is required to prevent the attack on the headquarters with a desired level of confidence. Since the commander is only concerned with the outcome of the fire support, the mission requirements are simply expressed as conditions on the third attribute OKP. The first two attributes which describe the window of opportunity are not taken into account at that level.

Definition of Primitives

Each node and link of the system is assumed to have a probability of failure, independently of the countermeasures of the enemy. Only the technical characteristics of the system are considered. This refers to the concept of reliability. The system is operating in a hostile environment. The communication links are subject to jamming from the enemy. Therefore, each node and link has a probability of failure due to enemy countermeasures. This refers to the concept of survivability. Although the two concepts of reliability and survivability are distinct (the two sets of probabilities of failure can be considered as independent), they are merged in the present analysis to reduce the dimensionality of the problem. A single probability vector $\underline{p}$ is considered for the set of nodes and links of the system: it embeds considerations both of reliability and survivability.

One of the simplest way to illustrate the influence of the event that is actually taking place is, for instance, to choose the speed w of the threat as a system primitive. This way a whole range of slightly different versions of the same scenario can be investigated by varying w.

It is assumed that the only uncertainty comes from the target estimates by the forward observer (intrascenario uncertainty). An appropriate system primitive can be, for example, the angle β that separates the two sightings (distance measurements) of the observer. Intuitively, the larger the angle β the more accurate the speed estimate but the longer the response time.

Perturbing the system primitives $\underline{p}$, β and w defines the system locus in the attribute space $(t^{**}, \Delta t, OKP)$.

The issue of the quality of option can be

addressed by considering two batteries instead of a single one. Then coordinated fire as opposed to uncoordinated can be studied.

Geometric Analysis

The geometric relations for this scenario are shown in Figure 5.

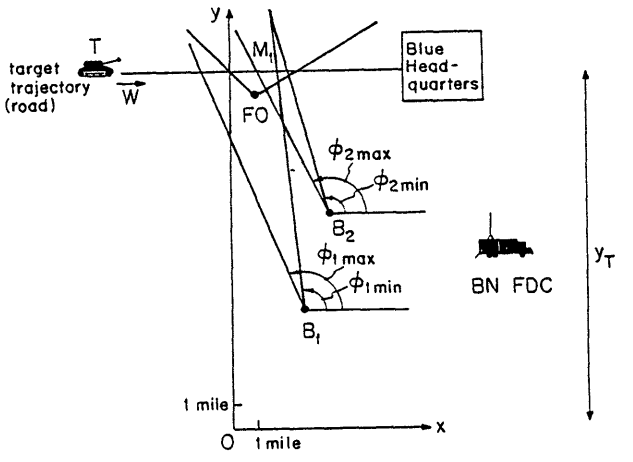


Figure 5. Geometric Relations of the Situation

Figure 6 shows the chronological sequence of the response process. The impact time, t_{impact} is given by:

$$t_{impact} = t_{obs} + \sum_{i=1}^{3} \Delta\tau_i \qquad (1)$$

$\Delta\tau_1$ is computed from the geometric properties of Figure 5. It is a function of the speed w, the angle β and the observation time (Cothier, 1984):

$$\Delta\tau_1 = \Delta\tau_1(w,\beta,t_{obs}) \qquad (2)$$

A sensitivity analysis shows that it is legitimate to consider $\Delta\tau_3$ as a constant for this topography and characteristics of the weapon system. In the present analysis, this constant is:

$$\Delta\tau_3 = 36 \text{ seconds} \qquad (3)$$

Let $t^* = \min\{t_{impact}\}$. For a given angle β and a given target velocity w, the earliest impact time corresponds to the earliest possible observation time, i.e., $t_{obs} = 0$ (detection time), and to the minimal time delay $\Delta\tau_{2_{min}}$ between the end of the estimation and the actual firing of the battery. Thus:

$$t^* = \Delta\tau_1(w,\beta,0) + \Delta\tau_{2_{min}} + \Delta\tau_3 \qquad (4)$$

Let M_1 be the point on the trajectory where the threat leaves the area covered by battery B_1 (see Fig. 5). For battery B_1 to be able to destroy the threat, the impact time must not occur after the threat has passed M_1, that is after time t^{**}. This creates an

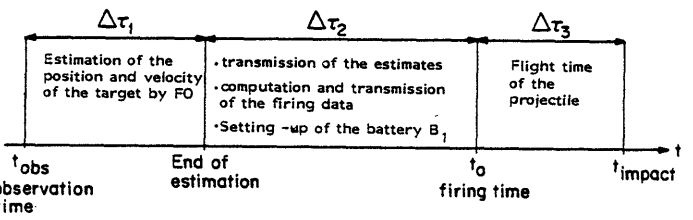


Figure 6. Time Profile of the System Response

upper constraint on the system capabilities:

$$t^{**} = \max\{t_{impact}\} \qquad (5)$$

Again, from geometric considerations,

$$t^{**} = \frac{K}{w} \qquad (6)$$

where K is a constant depending on the geometry of the situation. The quantity t^{**} characterizes the limit of the system capabilities when considering the latest response time possible to the initial stimulus.

Therefore, there are both a lower and an upper limit on the system capabilities as far as its response time to the stimulus is concerned. This time interval is the system window of opportunity: the system can deliver a response to the stimulus at any time t_{impact} lying between t^* and t^{**} (for $t^* < t^{**}$). The window of opportunity is completely characterized by the ordered pair $(t^{**}, \Delta t)$, where $\Delta t = t^{**} - t^*$.

The single shot kill probability $SSKP(t_{impact})$ associated with the impact time is easily computed by taking into account the uncertainty in the speed estimate, and the kill radius of the munition. For fixed values of w and t_{obs} the shape of the variations of SSPK with t is given in Fig. 7; the latter also shows an important trade-off. As β increases, the width of the window of opportunity decreases because it takes a longer time for the FO to make his estimation. But at the same time, a large β yields a more accurate estimate of the speed of the target. Therefore the kill probability is increased. The upper limit t^{**} is unaffected by changes in β.

Also, as time goes by, the uncertainty on the exact position x_T of the threat increases and therefore SSKP decreases with time.

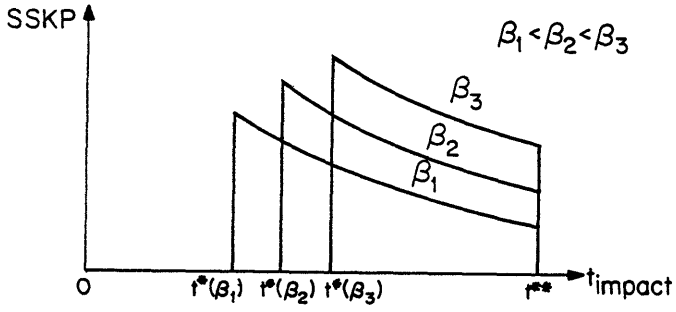


Figure 7. Single Shot Kill Probability as a Function of Impact Time

The seven element structure of the C^3 system has been presented in Fig. 3. The analysis reveals that out of the tem possible paths, six paths do not lead to the transmission of the information from FO to B_1.

Four paths lead to a successful communication between FO and B_1. For each path i (i=1,...4), the following quantities are defined:

q(i): probability that the path #i is operational

$u(i) = \Delta\tau_{2_{min}}(i) + \Delta\tau_3$, i.e., u(i) is the minimum time delay between the estimates by the FO and the impact time.

v(i): minimum time delay necessary to recompute new firing data based on the initial estimates, to transmit them and to set up the battery accordingly. If the system recomputes the firing data immediately after each shot and fires in sequence, then v(i) represents the minimum time delay between two shots ("minimum reshooting time").

Doctrine

The management of the time available for the system response has been shown to be a key point in the assessment of timeliness. The notion needs now to be applied to the example.

The earliest response time to the stimulus is t^*. The system can use the remaining time within the window of opportunity to deliver other responses, e.g., to fire again, therefore increasing the overall kill probability. This can be done in many different ways. This analysis focuses on two of them, which are classical military doctrines, known as "LOOK-SHOOT-SHOOT-SHOOT..." and "LOOK-SHOOT-LOOK-SHOOT...".

The "LOOK-SHOOT-SHOOT-SHOOT..." Doctrine: The observer initially makes estimates of the speed and position of the threat, and then the battery keeps on shooting at the target, recomputing each new firing data on the basis of these initial estimates.

The observation time is $t_{obs} = 0$ for each shot since there is no updating of the estimates. The time delay between two shots in thus the reshooting time v. The battery fires as many shots as possible within the window of opportunity since there is no feedback from the observer.

The "LOOK-SHOOT-LOOK-SHOOT..." Doctrine: After each shot, if the threat is neither destroyed nor incapacitated, the observer makes new estimates of its speed and position, new firing data are computed on the basis of these updated estimates, the battery shoots according to these new firing data, and so on until the upper limit of the window of opportunity is reached.

Derivation of the System Attributes

The three system attributes (t_i^{**}, $\Delta\tau_i$, OKP_i) are derived first for each path i of the 10 possible paths. In a second step, an overall probabilistic description of these attributes is given. For any of the 6 paths that fail to transmit the information from the FO to B_1,

$$\Delta t_i = 0 \qquad \text{for} \quad i = 5,\ldots,10.$$

The Overall Kill Probability is equal to zero for any of the paths #5 to 10, but for paths #1 to 4, it varies according to what doctrine is chosen.

The first attribute, the upper bound t^{**}, is assumed to be non-probabilistic. There are several different paths with associated probabilities. The width of the window of opportunity and the overall kill probability depend on what path is used. The relevant attributes to consider are thus the expected values of these quantities.

$$E(\Delta t) = \sum_{i=1}^{4} q(i) \cdot \Delta t_i$$

$$E(OKP) = \sum_{i=1}^{4} q(i) \cdot OKP_i \qquad (8)$$

From now on, only the expected values E(Δt) and E(OKP) will be considered. To simplify the notation however, they will be denoted by Δt and OKP, despite their probabilistic nature.

System Locus

The dependence of the system attributes on the system primitives is shown in Figure 8. It is

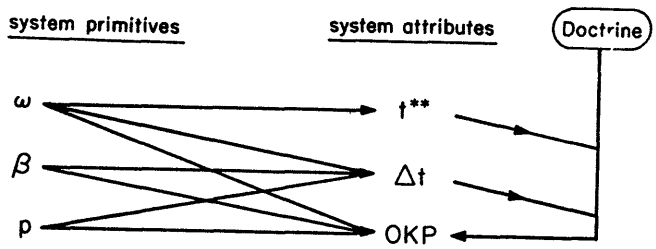


Figure 8. Mapping of the System Primitives into the System Attributes

interesting to note that OKP does not only depend on the primitives w, β and p, but is also computed from the two other attributes t^{**} and Δt (i.e., the window of opportunity) on the basis of the doctrine used. In other words, the primitives are mapped twice in the third attribute OKP, at two different levels.

At each value of the primitive set (w,β,p) corresponds a point in the attribute space (t^{**},Δt,OKP). Now consider all the allowable values that the primitives may take:

$$w_{min} < w < w_{max}$$

$$\beta_{min} < \beta < \beta_{max} \qquad (9)$$

$$p_{min} < p < p_{max}$$

If the primitives are allowed to vary over their admissible ranges, then the variations define a locus in the attribute space. This is the system locus L_s.

Mission Locus and Measure of Effectiveness

The analysis of the mission is much simpler since the mission requirements can be expressed directly at the attribute level, although it would be

preferable to find the mission locus in the attribute space by perturbing the mission primitives. More precisely, the mission requirements reduce to a single condition on the third attribute OKP that translates the level of confidence that is desired by the commander for achieving the fire support mission objectives. If λ is the level of confidence, where $0 < \lambda < 1$, then the mission locus is the region in the attribute space (t^{**}, Δt, OKP) that verifies the inequality:

$$1 \geq OKP \geq \lambda$$

For the present analysis, a simple measure of effectiveness (MOE) has been chosen. Let (L_s) be the volume of the system locus. Let $(L_s \quad L_r)$ be the volume of the intersection of the system and mission loci. Then the measure of effectiveness E is given by the ratio of these two volumes (Figure 9):

$$E = \frac{\mathcal{V}(L_s \quad L_r)}{\mathcal{V}(L_s)} \qquad (10)$$

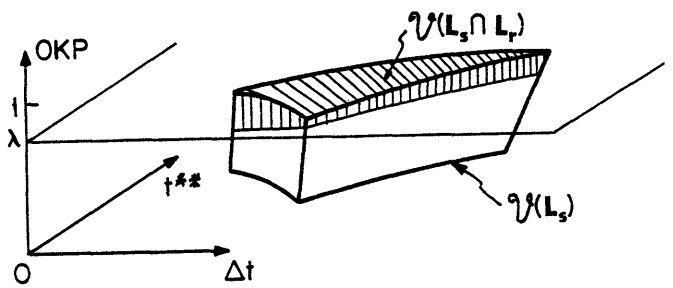


Figure 9. Measure of Effectivenes

4. COMPARISON OF DOCTRINES AND OPTIONS BASED UPON THE ASSESSMENT OF THEIR TIMELINESS

The One-Battery Case: Comparison of Two Doctrines

Figures 10 and 11 show the system locus and its intersection (shaded region) with the mission locus for both doctrines. The ratio of the shaded volume over the total volume of the system locus is larger for doctrine 1 than for doctrine 2:

E_1 = E(1 battery, doctrine 1) .55

E_2 = E(1 battery, doctrine 2) .50

When the threat moves rapidly, the window of opportunity is small: it is better to make a good measurement of its speed once and then fire in sequence without taking time to make new estimates, rather than to make an estimate, shoot, make a new measurement, and so on. Therefore, the "LOOK-SHOOT-SHOOT" doctrine has an overall effectiveness which is larger than that of the "LOOK-SHOOT-LOOK" doctrine. Its timeliness is thus better.

The Two Battery Case

When the two batteries B_1 and B_2 are considered, it appears from Fig. 5 that their areas of coverage overlap. Therefore the threat moves first on a part of the road that is covered by one battery (B_1), then on a part that is covered by two batteries (B_1 + B_2), then again on a part that is covered by only one battery (B_2). Intuitively, the probability of kill varies with time, suddenly increasing then decreasing. Assuming a "LOOK-SHOOT-SHOOT-SHOOT..." doctrine, two different options for the fire support commander will be considered:

Option 1: the two batteries shhot at the thret independently, each one using the maximum of its own window of opportunity. There is no coordination between the two batteries.

Option 2: Battery B_1 starts firing only when the threat enters the area covered by both batteries. In other words the commander decides not to fire immediately with B_1, but to wait until coordinated fire can be achieved, i.e., both batteries B_1 and B_2 shooting so that their projectiles hit the target trajectory at the same impact time. The global window of opportunity of the system is thus reduced to that of battery B_2. The time interval during which B_1 holds its fire can be used to keep the observer's estimate updated.

Figures 12 and 13 show the system locus and its intersection (shaded region) with the mission locus for both Options 1 and 2. The evaluation of the effectiveness of the system for both options, measured by the ratio of the shaded volume over the total volume of the system locus yields, the following results. Let E_3 be the MOE when Option 1 is used and E_4 when Option 2 is used.

Then

$$E_3 \simeq E_4 \simeq .6$$

Therefore, both options result in approximately the same value for the effectiveness of the system. The notion of the quality of option is appropriate here. In Option 2 fewer shots are fired than in Option 1. Therefore, coordination reduces costs for the same kill probability. Besides, in Option 2, while the battery B_1 is waiting, the threat does not know it is tracked and will not request any increase in the countermeasures (e.g., enemy jamming), nor start shooting at the blue force positions. In Option 1, this may happen as soon as B_1 starts firing, before B_2 has the opportunity to shoot. The survivability of the overall system is thus higher in Option 2 than in Option 1. Considering the closeness in the value of the effectiveness measure, one can thus conclude that Option 2 (wait and coordinate) is of better quality than Option 1 (immediate uncoordinated fire).

It is important to note that the quality of Option 2, coordinated fire, is better than that of Option 1, although its window of opportunity is much narrower. In fact, the time available is better managed: it is more effective to wait in order to implement a better option. This example shows that the quality of an option and the size of the window of opportunity are two independent characteristics.

5. CONCLUSION

This paper addresses the need for a measure of timeliness as described by Lawson (1981). Without such a measure, any assessment of a command and control system is incomplete because the information is axiomatically assumed to be "timely", i.e., the

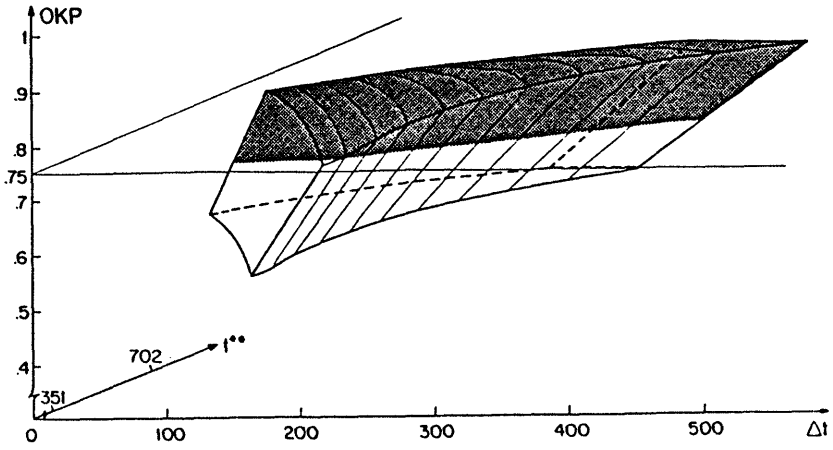


Figure 10. Doctrine 1 (LOOK-SHOOT-SHOOT) System and Mission Loci

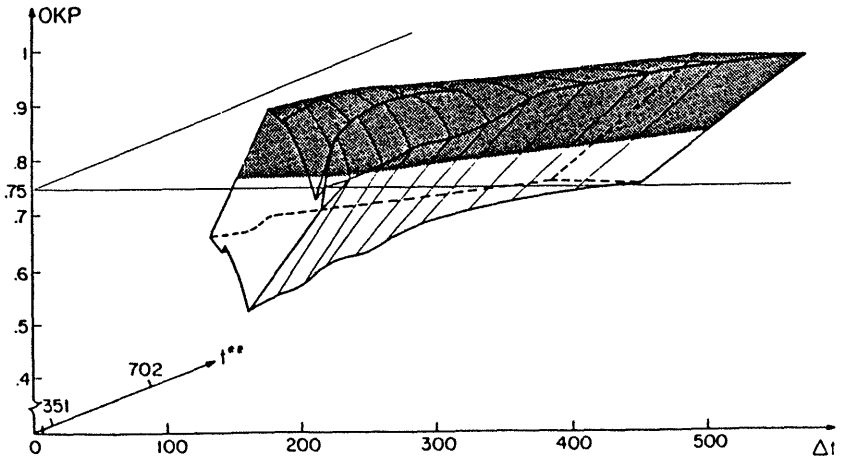


Figure 11. Doctrine 2 (LOOK-SHOOT-LOOK) System and Mission Loci

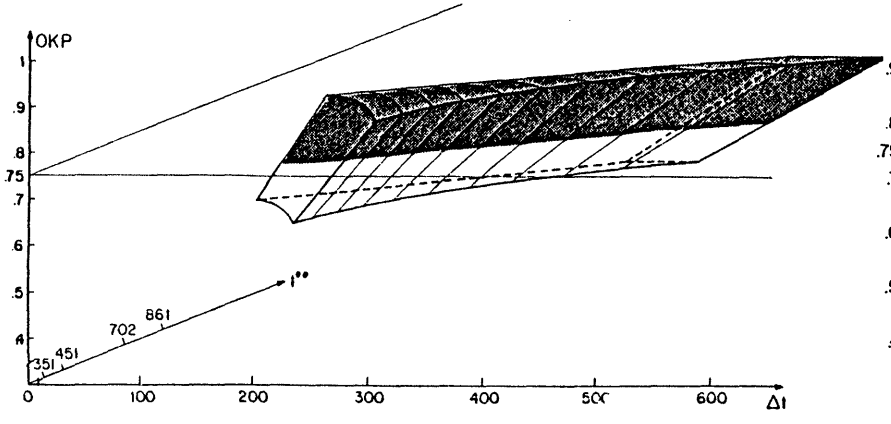


Figure 12. Option 1 (immediate fire without coordination) System and Mission Loci

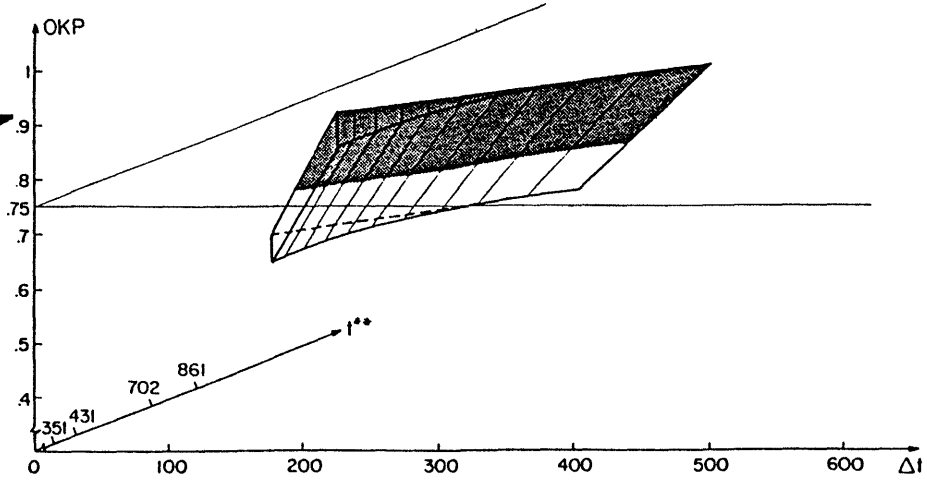


Figure 13. Option 2 (wait and coordinate) System and Mission Loci

issue of timeliness is not addressed. In this paper, the temporal characteristics of the system are treated on the same level as the other performance characteristics. More precisely, time is not taken into account only as a denominator in the definition of rates, but as a fundamental factor with its own special characteristics. The proposed methodology allows the evaluation of a measure of effectiveness embedding all the time-related notions: response time, tempo of operations, uncertainty, quality of options, scenario, and window of opportunity. The elusive concept of timeliness that rests upon these notions can thus be captured and modeled quantitatively.

In developing the methodology, approcahes to important issues on the influence of time in command and control have been introduced. First of all, partial measures of effectiveness allow the quantitative comparison of different doctrines. Some doctrines are shown to make better use of the available time than others and effectiveness analysis can aid in the selection of doctrines appropriate to a given situation. Without such a tool, the comparison can only be carried out through simulations or tests; these are, however, much more expensive assessment methods (Zraket, 1980).

A second point has been illustrated by considering the relationship between different aspects of the system components. While the speed of processing and transmission of data can be improved, the effectiveness of the system may not change if, for instance, the reliability and survivability of the system's components are not also improved. Faster does not necessrily mean better; it can even mean worse, if the increase in speed is gained at the expense of the system's survivability. The proposed methodology allows a decisionmaker to relate a change in one part of the system to a change in another part. The strength of system effectiveness analysis is the ability to carry the assessment on an overall basis: the variations in the features of a given system are not considered separately, but jointly. This yields useful perspectives for future design of C^3 systems. The influence of any modification either in the components, or the organization, or the doctrine, can be evaluated using the proposed measure of effectiveness of the system. Lawson (1981) pointed out the consequences of insufficient attention to timeliness: the system designers' effort is focused primarily on the performance characteristics of the system components (e.g., bit rate or capacity). This methodology shows promise as a tool in the computer-aided design of such systems.

A third point refers to the window of opportunity. A wider window does not mean a more timely system. Timeliness is a more subtle concept and this is the reason why a measure based on the relative window widths is not meaningful. The quality of the management of the time available, i.e., the window of opportunity, is at least as important as the width of the window. Therefore, the size of the window of opportunity is not a sufficient determinant of a system's timeliness. The set of possible options, and their respective quality as responses to the initial stimulus, must also be considered. Desirable responses may be implemented within a narrow window, whereas a wider window may allow undesirable ones to be considered. The feature of the methodology presented in this paper is that it stresses the importance of the quality of options and embeds this, as well as the window widths, in the measure of effectiveness.

Extensive use of such a methodology could lead to a system design philosophy that integrates timeliness into system design. Moreover, the methodology is flexible enough to be adapted to many kinds of C^3 systems, (Bouthonnier and Levis, (1984); Karam and Levis (1984)) as well to other systems, e.g., automotive (Levis et al, 1984) manufacturing ones (Washington, 1985) or guidance and control (Ponty, 1984).

6. REFERENCES

Bouthonnier, V. and A. H. Levis, "Effectiveness Analysis of C^3 Systems," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-14, No. 1, January/February 1984.

Cothier, P.H., "Assessment of Timeliness in Command and Control," M.S. Thesis, LIDS-TH-1391, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1984.

Dersin, P. and A. H. Levis, Large Scale Systems Effectiveness Analysis. Report LIDS-FR-1072, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, February 1981.

Dersin, P. and A. H. Levis, "Characterization of Electricity Demand in a Power System for Service Sufficiency Evaluation," European Journal of Operations Research, Vol. 11, No. 3, November 1982.

Karam, J. G. and A. H. Levis, "Effectiveness Assessment of the METANET Demonstration," Proc. 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1984.

Lawson, J. S., Jr., "The State Variables of a Command and Control System," Proceedings for Quantitative Assessment of Utility of Command and Control Systems, National Defense University, Washington, D.C., January 1980.

Lawson, J. S., Jr., "The Role of Time in a Command Control System," Proceedings of the Fourth MIT/ONR Workshop on Distributed Information and Decision Systems Motivated by Command-Control-Communication (C^3) Problems, Report LIDS-R-1159, Vol. 4, C^3 Theory, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, October 1981.

Levis, A. H., P. K. Houpt and S. K. Andreadakis, Effectiveness Analysis of Automotive Systems, LIDS-P-1383, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, June 1984.

Ponty, P., Calibration et evaluation en dynamique de centrales inertielles liees. Etude theorique d'evaluation - Rapport de synthese, Association pour le Development de l'Enseignement et de la Recherche en Systematique Appliquee (ADERSA), Commande SFIM, No. 317-966. Marche DRET No. 82/188, Lot No. 3, Palaiseau, FRANCE, February 1984.

Washington, L. A., "Effectiveness Analysis of Flexible Manufacturing Systems," M.S. Thesis, LIDS-TH-1430, Laboratory for Information and Decision Systems, MIT, Cambridge, January 1985.

Zraket, C. A., "Issues in Command and Control Evaluation", Proceedings for Quantitative Assessment of Utility of Command and Control Systems, National Defense University, Washington, D.C., January 1980.

Assessing the Organizational Responsibility of Headquarters Under Differing Level of Stress

Prof. Michael G. Sovereign
CDR Joseph S. Stewart

Naval Postgraduate School

Introduction

This paper describes the second in a series of full scale computer aided wargames which have applied a new approach in quantitative measurement of command and control. This new approach incorporates the use of the Headquarters Effectiveness Assessment Tool, which was developed by Defense Systems Incorporated (DSI) of McLean, Virginia, in measuring the responses of headquarters during full-scale exercises and subsequent simulations here at the Naval Postgraduate School (NPS).

The purpose of the Headquarters Effectiveness Assessment Tool (HEAT) developed under contract to Defense Communications Agency (DCA) is to enable a team of internal or external observers to objectively assess and quantify headquarters performance and effectiveness. HEAT combines elements from several different approaches to measuring effectiveness, particularly:

* Headquarters or adaptive control systems
* Effectiveness at military mission accomplishment, and
* Command and control processes as information management systems

The underlying conceptual model of the headquarters process is shown in Figure 1. HEAT provides a means of assigning measures of value to the ongoing processes and subsequently using the aggregates to assess performance. The use of HEAT requires the formation of a plan of action. The HEAT measures are used, thereafter, to quantify adherence to the plan or the adequacy of revisions to the plan.

FIGURE 1

The original applications of the Headquarters Effectiveness Assessment [Ref. 1] Tool (HEAT) were described by one of these authors in a paper given at the San Diego ONR-MIT meeting last year which described evaluation of Exercise Bold Eagle 84 and an NPS laboratory experiment in the fall of '83. Exercise Bold Eagle 84 was a full-scale exercise held at Eglin Air Force Base in October 1984. Prior to the exercise an evaluation team helped the JOINT TASK FORCE 7 staff develop command standards for performance in certain areas of headquarters performance. Of the 8 stated goals of the exercise, three of which were concerned with essential elements of information were examined by the HEAT observer team. They are shown in Figure 2.

FIGURE 2

Our 20 observers collected several thousand data sheets. The subsequent analysis supported observations seen during the exercise regarding staff performance. In summary:

* The staff failed to identify major incongruities between their plan and the events actually occurring.
* Major missions were not accomplished in the planned time frame,
* In general the staff did not provide timely or accurate information on enemy OR own force units according to their own standards.
* The planning and operations failed to establish authoritative and physical connectivity among the participants due in part to lack of modern communications and display equipment.

C2 Laboratory Experiment

HEAT principles were also used in and tested in a month-long command and control experiment in the Naval Postgraduate School (NPS) Wargaming Analyses and Research Laboratory (WAR LAB) during the month of November 1983. The design, conduct and analysis of the experiment was a joint Naval Postgraduate School, Defense Communications Agency and Defense Systems, Inc. effort.

The purpose for conducting the experiment was to attempt to corroborate findings, primarily by the Soviets [Ref. 2], which indicate the command structure supporting a battlefield headquarters influences that headquarters' effectiveness and thus impacts on the speed and correctness of decisions.

The experiment conducted in the War Lab used the Naval Warfare Interactive Simulation System (NWISS) hosted on a VAX 11/780 mini computer. NWISS is a large scale (250,000 lines of code) highly interactive, naval wargame with color graphics. A standard set of military problems were posed to military officer students who performed in distributed roles using several headquarters command structures. The data collection plan permitted use of HEAT Measures in an attempt to corroborate the Soviet findings. The physical design of the experiment closely resembles that reported on later in this paper. Four headquarters were established which simulated responsible headquarters under the Navy's Composite Warfare Commander (CWC) concept at coordination of the Antisubmarine Warfare (ASW), Anti Air Warfare (AAW) and Anti Surface Warfare (ASUW) commander by the Officer in Tactical Command (OTC) who is usually an admiral.

The command modes were restricted to communications via three different command structures, as shown in Figure 2, depending on the individual scenario being presented.

The results of the experiment are shown in Figure 3.

* Star structures are slightly faster than fully-connected structures but not to a statistically significant level.

* (Did not contradict Soviet findings.)

* The fully connected structure was able to reach a decision more often than the star structure but the decision error rate was about the same.

* (Did not contradict Soviets findings.)

* Fully connected structures were always slower to initiate hostilities mistakenly than were other structures.

* (An independent finding.)

Figure 3

In a continuation of sponsorship by the DCA, a second set of experiments was conducted by DSI and NPS. Lessons learned in the previous experiments allowed refinement of experimental design and the application of heat measures.

The war game which is the subject of this paper was designed by the DSI and NPS staffs and conducted in the Wargaming Analysis and Research Center by the faculty and officer-students and staff of the Naval Postgraduate School. While the earlier experiments attempted to study the effects of connectivity on the performance of the Navy's CWC concept, within a battle group, the current studies examined the performance of multiple battle groups as role specialization was varied from functional to geographic. The quantitative assessment of the performance of these headquarters was accomplished using HEAT measures and statistical analyses of times associated with message traffic between headquarters as well as overall exchange ratios.

In earlier work the effectiveness of distinct command structures when faced with equal threats was studied. In these latest experiments the question of effectiveness of a headquarters was studied as both the organizational lines of responsibility and the level of threat varied. The basic organization the major headquarters was as shown in Figure 1 and the physical setup for the experiment in Figure 4. The organization, resembling a basic star, prevented fully connected communications with higher levels of command staff headquarters but supported fully-connected communications between (2) "operational headquarters" (Figure 5). Within the fully-connected portion of the structure two conditions could exist; (1) Geographic organizational lines of responsibility or (2) Functional organizational lines. On the first -each operational headquarters controlled all friendly forces in all major warfare areas in his geographic sector (Figures 6, 7). In the second, shown in Figures 8A, 8B, 9, 10), the separate operational headquarters controlled all assets subordinated to their warfare area regardless of the physical location of the asset in the area of hostilities. Under each scenario, reactions of these headquarters were recorded for later analysis using HEAT measures.

Figure 4

Figure 5

Figure 6

Figure 7

Figure 8A

Figure 8B

Figure 9

Figure 10

The purpose of the experiments was threefold as shown in Figure 11.

Figure 11

Design of the Laboratory Experiment

In current planning the concept of fixed headquarters sites violates the assumption that mobility of the headquarters enhances survivability. The desire for mobility implies that a reduction in the size of the staff is warranted if it is to be mobile. A question then arises which may best be stated; "how small is too small".

To avoid continuous variability in one experimental parameter, however, it is necessary to hold the size of the staff constant at a relatively small number and to increase activity until the staff becomes ineffective. When the point of loss of effectiveness is determined the complimentary questions "How much activity could the small staff handle relative to a large staff" and "Is there a significant difference between the two levels?", can be answered to gain insight in answering the original query. A headquarters size of five persons was determined to be reasonable but small and the central assumption for the experiment was stated as a hypothesis.

Ho: Under increasing levels of activity (stress) a small headquarters organized geographically will show reduced effectiveness.

A second hypothesis.

Hl: Under increasing levels of activity (stress) a small headquarters organized functionally will show less effectiveness at all levels than the geographic case, due to the burden of coordination, unless or until the level of activity surpasses the ability of a small staff regardless of organization.

The alternate hypothesis.

Ho: There is no appreciable difference in the level of effectiveness of the small staff due to organization.

Figure 12 is a graphic depiction of these hypotheses.

Figure 12

The experiment was again designed and conducted using the Naval Warfare Interactive Simulation System (NWISS). Military officers who were previously trained in the system comprised the teams. A naval scenario was constructed which involved three carrier battle groups (CVB6) on station simultaneously in the Arabian Sea. A fourth formation, which acted as the center of attention, was a convoy of petroleum tankers in the confined waters of the Strait of Homuz being escorted by two surface units and a submarine detached from the supporting

CVBG's. Each CVBG consisted of a carrier with embarked air wing, five surface combatants, and a support ship. Additional supporting vehicles not dedicated to a specific CVBG were two direct support fast-attack nuclear submarines and land-based maritime air support.

Arrayed against this task force were air and sea forces of the USSR, IRAQ and IRAN. The levels of stimulation ranged from intimidation to provocation to attack. The headquarters was challenged to decide who the enemy was and then to make the appropriate response. To accomplish the tasks each station in the operating units was equipped with four devices; a player terminal, a status board terminal, a graphics display terminal and a communication terminal. Figure 13 shows a representative arrangement. The Commander 7th Fleet (C7F) position played by subjects and the Commander Pacific Fleet (CPF) position, played by the umpire team, consisted of a single communications terminal located in a space where charts were available. The subjects could communicate with the computer and the control team via the player (or order entry) terminal and with other headquarters solely through the communications terminal. Again the CVBG's had the potential for fully-connected comms whereas all traffic to CPR and C7F had to be passed through the OTC.

Figure 13

Forty-five subjects were randomly arranged into three groups and headquarters team membership within each section was also randomly determined. The groups participated in a practice session and six sessions for record as shown in the experimental design shown as Figure 14.

Figure 14

During each three-hour session the players divided responsibility between operating the terminals and being the battle group commander or an observer. Subjects were rotated between sessions to further insure randomness in player's skill and experience. At the beginning of each run the players were presented with the politico-military situation and reminded of established rules of engagement and the functional or geographic organization for that session. Thereafter the wargame was a free-play exercise.

The combination of opposing forces was predetermined by DSI. The combination of these forces and their level of activity was randomly combined with the organization being studied to avoid bias in the data and the opposing forces were prescripted. After game start the actions of the opposing forces were automatically accomplished by the computer according to the script file although control could override the prescript to adjust for actions taken by the CVBG commanders. The prescript helped to guarantee that all teams were exposed to identical threat scenarios for the first 90 minutes of a 2.5 hour session.

The Generation of Data

The NWISS software provides the capacity to collect and file every player position order and the computers response from each headquarters for the entire duration of each session. This capability was utilized to create archival files which were stored for immediate post-game analysis and transferred to tape for later reference. Each session provide four game files (CVG's A, B, C and Control). The control software of the host computer provided the capability to search these files for individual occurrences of interest or combinations of occurrences. In the experimental design phase, once the desired measures of effectiveness were determined, search programs were set up as necessary. Immediate postgame analysis consisted of operating on the four game files with the search files to generate hard-copy results from which HEAT scores could be generated. These were provided to DCA for further analyses by DSI.

In addition, during the work up phase a communication program, COMNET, was written to control the transfer of messages between the communications terminals. In addition to routing the traffic the same piece of software;

1) assigned key times to each message,
2) provided the capability to "jam" a particular terminal,
3) provided the ability to delay transmission
4) allowed the garbling of a percentage of the letters in each message from 10 to 90 percent, and
5) provided a hard copy of each message to each headquarters.

Control could further induce stress in selected sessions through intermittent jamming of one or more stations during the last hour of the session. The garble and delay were not used in this experiment.

All messages were sent to a central file which could be post processed. After being sorted into 1 of 9 categories each message was examined by the analyses section of the software. The differences between the four assigned times provided three statistics for each message: Throughput time, Delay time and Preparation time. By accruing the individual statistics over all messages for each run, session average times and standard deviations were provided. Considering that the average number of messages for a three-hour session was more than 350, the use of mean times for subsequent analyses of the eighteen sessions was justified. The results of the analysis of message times were provided by a student team as a class project and they will be utilized in a forthcoming thesis. [Ref. 3]

In summary, the data package for each session consisted of a selection of measures of effectiveness sorts of the orders issued by the battle group commanders, and a packet of descriptive statistics of times associated with the corresponding message traffic.

Preliminary Analysis of Data

The experiment was completed on 30 October with data reduction continuing til mid-November 1984.

The cursory analyses of message times shows results which support further analysis. In the analysis Throughput Time = Preparation Time + Delay Time. By examining the variability of the average Preparation time statistics and the average Delay time statistics, it was found that the variability in delay times accounted for the majority of the variability required to read and analyze messages waiting in the queue. Delay time would seem to be consistent with decreasing efficiency of the staff. In the worst case long delay times would indicate the failure to respond to orders and queries from outside the command due to collapse of the decision-making function in the headquarters.

For each of the mean statistics it is of interest to determine whether the organizational lines of responsibility effected performance in message handling. The mean times were segregated by geographic or functional organization and the hypothesis selected.

Ho: That the grand mean of the one distribution is not statistically different from the other.

$$\mu_o \quad \text{i.e.} \quad \mu g = \mu f$$

with

$$H1: \quad \mu g = \mu f$$

Given that all data points were themselves mean the resultant reduction is dispersion of the data points caused applicable tests to support Ho. In the process of forming the distributions of means, however, a marked kurtosis or peakedness was discovered for each subset and it could be shown that the mean and standard deviation for the functional organization trials were lower than the geographic trials. The suggested implication is that the functional organization allows the various warfare commanders to concentrate on a single type of prosecution and that the increase in effectiveness overshadows the added burden of coordinating with other battle groups to accomplish the warfare area mission over a wider portion of the globe. Further analysis using more powerful tests on these data is being considered.

A separate analysis of game files from all headquarters was conducted by DSI. Heat measures were applied where applicable and other measures of effectiveness were assesed, i.e. exchange ratios. The anticipated findings are shown in Figure 15.

Figure 15

The results were collected and can be seen by comparison in Figures 16 through 18. An additional assessment of causal linkages between the headquarters is still ongoing as of April 1985 with a final report to be presented to the Defense Communications Agency.

In summary, the most recent effort carried out in the WAR lab produced the results shown in Figures 19 and 20.

The efforts which have been waged currently in the analyses of headquarters effectiveness have developed a new methodology for C2 investigations. That methodology consists of analyses of full scale exercises, wherein many of the key variables are not controlled by the experimenter, coupled with subsequent laboratory experiments wherein the variables are more closely controlled. When the resultant data are analyzed using the same tool, understanding of those results should be enhanced. Movement from the macro to the micro examination of concepts may be a first step in providing confirmation of the existence of and identification of illusive principles of command and control.

Figure 16

Figure 17

Figure 18

Figure 19

Figure 20

REFERENCES

1. Experiments in C^2 Utilizing the Headquarters Effectiveness Assessment Tool (HEAT), Porter, G. R., Sovereign, M. G., Naval Postgraduate School, July 1984.

2. Durzhonen, V.V., Concept, Algorithm, Decision (a Soviet view), Moscow, 1972. Translated and published under the auspices of the United States Air Force.

3. Hardee, N.E., An Assessment of the Ability of the Headquarters Effectiveness Assessment Tool (HEAT) to Quantify the C2 System Effectiveness of a Simulated U.S. Navy Tactical-Level Headquarters Under Periods of Communications Stress, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1985.

EFFECTIVENESS ANALYSIS OF EVOLVING SYSTEMS*

Joseph G. Karam
ICF, Inc. 1850 K Street, Suite 950, Washington, D.C. 20006

Alexander H. Levis
Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139

ABSTRACT

A quantitative methodology for analyzing the effectiveness of evolving systems that will undergo a series of demonstrations is presented. Emphasis is placed on the design of the demonstration by assessing the effectiveness of alternative system configurations. The system's performance and the mission requirements are described in terms of a finite number of attributes, using a probability distribution and a utility function, respectively. The approach is illustrated through an example based on the development of a network of networks.

1. INTROUDCTION

Consider an organization that is developing a large-scale system such as a large communication network. The completion of this system will take a number of years and require sustained funding. The latter, however, is contingent on (a) the progress made in developing the system, and (b) the prospects it has for meeting the needs for which it is being designed. One way of checking whether these conditions are met is to set up a timetable in which several demonstrations are scheduled. The focus of these demonstrations will be to show that real progress has been made in developing the system, and that the latter will be capable of performning the tasks for which it was designed.

The methodology developed in this paper will help the organization understand better the basic trade-offs and design, with greater awareness of the consequences, demonstrations of the system. More generally, it aims at analyzing the effectiveness of evolving systems, that is, systems that are constantly upgraded as new technologies are made available and as the needs or interests of the various participant groups are redefined.

2. PROBLEM FORMULATION

The specific features of evolving systems affect all aspects of the System Effectiveness Analysis methodology (Bouthonnier and Levis, 1982). Indeed, they appear on the system side, the mission side, and the context, and contribute to the definition of the relevant attributes.

Figure 1 suggests the intimate interaction between the basic aspects of the methodology. It shows the system-context and mission-context interactions. Also, it sketches the joint contribution of the system, mission, and context, to the definition of the relevant attributes.

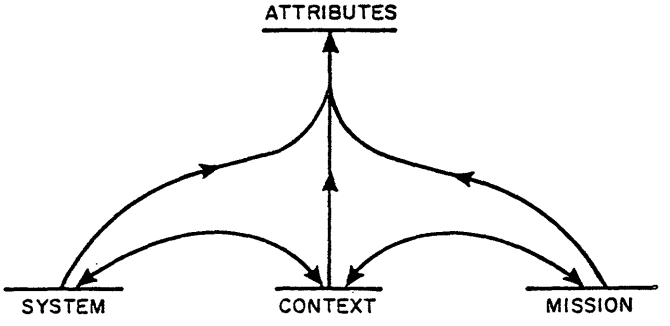


Figure 1. Evolving Systems: The Overall Picture

2.1 The Context

An evolving system typically undergoes a series of demonstrations. Such demonstrations consist, in general, of a succession of stages or events. A stage can be aimed at demonstrating a specific technology, carrying out a given function, or both. The sequence of events and their contents correspond to a scenario. Depending on the scenario adopted, the demonstration will be shaped differently. Hence, the choice of a scenario is a decision variable; the objective is to optimize the effectiveness of the demonstration.

2.2 The System

Let T_j denote the j-th component/technology of the system that is being developed:

$$S_\infty = \{T_1, T_2, \ldots, T_j, \ldots, T_J\} \quad (1)$$

The components T_j can be physical components, i.e., nodes of the network or gates between nets, or even switches, or they can be software implemented on specific hardware.

Since this is an evolving system, at any time t, a component T_j may not be fully operational. If $\lambda_j(t)$ denotes the degree to which T_j is functional, i.e.,

$$0 \leq \lambda_j(t) \leq 1 \quad (2)$$

and if $\underline{\lambda}_j$ denotes a threshold of operability for component j, then S(t) is the subset of S that is operational at time t:

*This research was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the Space and Naval Warfare Systems Command under Contract No. N00039-83-C-0466.

$$S(t) = \{T_j(t) \; ; \; \lambda_j(t) \geq \underline{\lambda}_j\} \qquad (3)$$

As time increases, the subset S(t) should expand until, at the end of the project period, it is equal to S_∞ (all component parts are completed). Out of the set S(t), system architectures can be configured that are suitable for demonstration. Not all configurations include all the operational components, and not all configurations are equally effective for the demonstration. These concepts can be stated formally as follows:

Let P(t) be the set of all subsets P of S(t),

$$P(t) = \{P, \; P \subset S(t)\} \qquad (4)$$

If S(t) contains $\#_T$ elements, then the number of subsets in P(t) is $2^{\#_T}$. However, not all of them lead to useful configurations. Let $\tilde{P}(t)$ be the subset of P(t) that merits consideration. It is expected that few non-trivial configurations would be possible at any time. The procedure for determining the set $\tilde{P}(t)$ of useful configurations is sketched out in Figure 2.

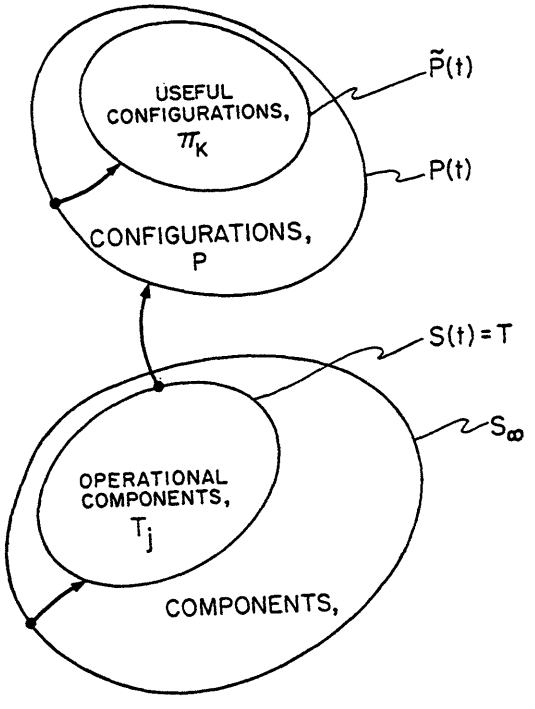


Figure 2. From the Ultimate System S_∞ to the Set $\tilde{P}(t)$ of Useful Configurations

This conceptual framework is applied now to the determination of $\tilde{P}(t)$. Let t_1 be the time at which the design of selected components is fixed so that prototype operational versions can be developed and let t_2 be the time of the proposed demonstration. Then the procedure can be described as follows:

(a) Consult with contractors to determine the components T_j that can be considered operational at time t_1 in the future.

(b) Consult with users to determine existing components and subsystems that could be made available for the demonstration at time t_2.

(c) Combine the results of (a) and (b) to determine set $S(t_2)$.

(d) Out of the elements in $S(t_2)$, design alternative system configurations, i.e., construct $\tilde{P}(t_2)$.

(e) Elements of $S(t_2)$ that have not been used in any of the candidate configurations in $\tilde{P}(t_2)$ should be dropped from further consideration for the demonstration at t_2.

The above procedure establishes the alternative system configurations for the demonstration. But to select the most effective one, the goals of the demonstration must be established.

2.3 The Mission

The demonstration of an evolving system has a dual goal. First, it should show the capabilities of the system that is being developed. It should also demonstrate progress and accomplishments in developing the system. This goal may be only partially shared by the various participants in the demonstrations. The first of the four major sets of participants consists of the contractors, the engineers and scientists who are developing the components, or are concerned with or system integration. The second participant is the agency that is the program sponsor and manager. The system contractors, I, and the agency, A_g, can be taken together to constitute a combined group, the developers (A). The third set of participants (B) consists of the system's users, the persons who are going to use it in carrying out their duties (ultimately as well as during the demonstration). Finally, there is the group of decisionmakers (C), who will observe the demonstration, and can make decisions about the program's continuation and eventual implementation.

All of them would like the demonstration to succeed. In addition to this common concern, group A would like to see more components demonstrated. Typically, each developer in group A would focus on "his" technologies and see to it that they are included in the demonstration. Conversely, group C would like to see more functions carried out during the demonstration. Typically, each decisionmaker in group C has a set of functions which he believes the demonstration should execute. The concept of function is used in contrast to that of end-product embodied by the components or technologies. In command and control, a function would be, for example, the interaction between commanders, or between a commander and a unit or organization. Let T and F denote the set of technologies and functions, respectively. Note that T is nothing but the set S(t) in developing the system model.

After having specified the context and developed the system and mission models, the attributes can now be introduced.

2.4 System Attributes

System attributes depend on variables (the system primitives) which describe the system's characteristics and on the context. In a given context, a system is not expected to realize a specific combination of values of its attributes $x_1,\ldots,x_n$ with probability one. Instead, a set of realizable combinations exists, each corresponding to a set of values taken by the system attributes. This set, L_S, is the locus of the system attributes. Any point $\underline{x}$ that belongs to L_S has a non-zero probability

of being actually achieved by the system. To model this concept, a probability distribution f is introduced which is a complete description of the system's performance in the specified context. Therefore, for each useful configuration π, $\pi \in \tilde{P}(t)$, let f_π be the probability distribution of the system attributes $\underline{x}$.

In accordance with the dual role of the demonstration, the attributes that are relevant to assessing the effectiveness of evolving systems belong to one of two classes: Type 1 and Type 2.

Type 1 attributes are those with which the System Effectiveness Analysis would be concerned, if it were applied to a non-evolving or fully developed system. These attributes characterize the effectiveness of the ultimate system; they form a vector $\underline{y} = (y_1,\ldots,y_n)$. In the case of communication networks, reliability, input flow, and time delay are examples of Type 1 attributes.

In general, Type 1 system attributes are continuous random variables. Let L'_S denote the system locus in the Type 1 attribute space, i.e.,

$$L'_S(\pi) = \{\underline{y} \quad ; \quad g_\pi(\underline{y}) > 0\} \tag{5}$$

The second stated goal of the demonstration is to show progress and accomplishments in developing the system. The achievement of this goal is expressed in terms of Type 2 attributes, denoted by the vector $\underline{z}$. In this case, the attributes are two: z_A and z_C. Attribute z_A is a weighted fraction of the technologies used in the demonstration, while attribute z_C is a weighted fraction of the functions carried out:

$$z_A = \sum_{i=1}^{\#T} \omega_A(T_i)\, \tau(T_i) \;/\; \sum_{i=1}^{\#T} \omega_A(T_i) \tag{6}$$

and

$$z_C = \sum_{j=1}^{\#F} \omega_C(F_j)\, \phi(F_j) \;/\; \sum_{j=1}^{\#F} \omega_C(F_j) \tag{7}$$

where

T_i denotes technology i , $i=1,\ldots,\#T$

F_j denotes function j , $j=1,\ldots,\#F$

$$\tau(T_i) = \begin{cases} 1 & \text{if technology } i \text{ is included in the demonstration} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(F_j) = \begin{cases} 1 & \text{if function } j \text{ is carried out in the demonstration} \\ 0 & \text{otherwise} \end{cases}$$

$\omega_A(T_i)$ weighting of technology i by the developers (group A)

$\omega_C(F_j)$ weighting of function j by the decisionmakers (group C)

These two attributes z_A and z_C defined by Eqs. (6) and (7) take discrete values between zero and one. For each system configuration π, a specific subset of the technologies T is used and a specific subset of the functions F carried out. The values taken by z_A and z_C are hence known with certainty:

$$z_A = z_A(\pi) \quad ; \quad z_C = z_C(\pi) \tag{8}$$

The Type 1 and 2 attributes form a vector, $\underline{x} = (\underline{y},\underline{z})$, which takes values in a subset of the (n+2) dimensional attribute space.

The distribution f_π is a Dirac function δ in the plane (z_A,z_C) at the point $(z_A(\pi),z_C(\pi))$. Distribution f_π can thus be written as follows:

$$f_\pi(\underline{x}) = g_\pi(\underline{y})\; h_\pi(\underline{z}) \tag{9}$$

where

$$h_\pi(\underline{z}) = \delta(\underline{z} - (z_A(\pi),z_C(\pi))) \tag{10}$$

The function $g_\pi(\underline{y})$, the component of $f_\pi(\underline{x})$ in the Type 1 attribute space, remains to be defined.

2.5 Mission Attributes

Mission attributes are used to describe the mission requirements in a specific context. Hence, they depend on variables which describe the mission characteristics (the mission primitives) and on the context. The set of combinations of attribute values that satisfy the requirements of the mission generates the locus of the mission L_M. Any point $\underline{x}$ that belongs to the mission locus satisfies, to some extent, the mission. However, all such points are not, in general, equally satisfactory. To model this concept, a utility function u is introduced that translates into a real number (between zero and one) the desirability, from the point of view of the mission, of each combination of attribute values. Since utility functions should be monotonically non-decreasing with respect to each of their arguments, the attributes should be defined in a way such that a higher value of any one attribute leads an to equal or higher utility, other things being equal.

Each group expresses its satisfaction -- or dissatisfaction — with the demonstration through some of the attributes. While all three groups are concerned about the values taken by the attributes $\underline{y}$, group A is, in addition, interested in the attribute z_A, and group C in the attribute z_C (see Figure 3). The partial utilities u_A, u_B, and u_C of groups A, B, and C respectively, can be written as:

$$u_A(\underline{x}) = v_A(\underline{y})\; w_A(z_A) \tag{11}$$

$$u_B(\underline{x}) = v_B(\underline{y}) \tag{12}$$

$$u_C(\underline{x}) = v_C(\underline{y})\; w_C(z_C) \tag{13}$$

The global utility is a function of the partial utilities introduced previously. For example,

$$u = a\, u_A + b\, u_B + c\, u_C \qquad \text{(additive)} \tag{14}$$

or

$$u = u_A^{\,a}\; u_B^{\,b}\; u_C^{\,c} \qquad \text{(multiplicative)} \tag{15}$$

where $a + b + c = 1$.

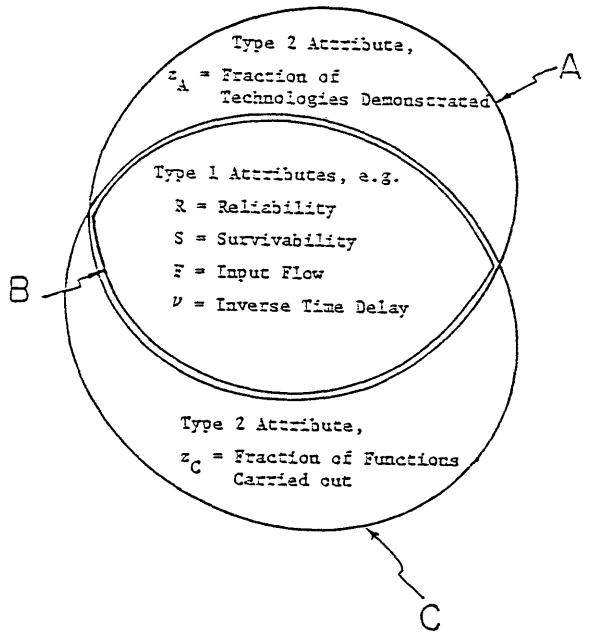


Figure 3. Partition of Attributes in Utilities of Participant Groups

Weights a, b, and c reflect the participants influence on decisions, regardless of their interaction. In reality, the three groups of participants in a demonstration are not independent. They interact before, during, and after the demonstration. Thus, it is important to sketch a model of the organizational interactions. One such model is shown in Figure 4.

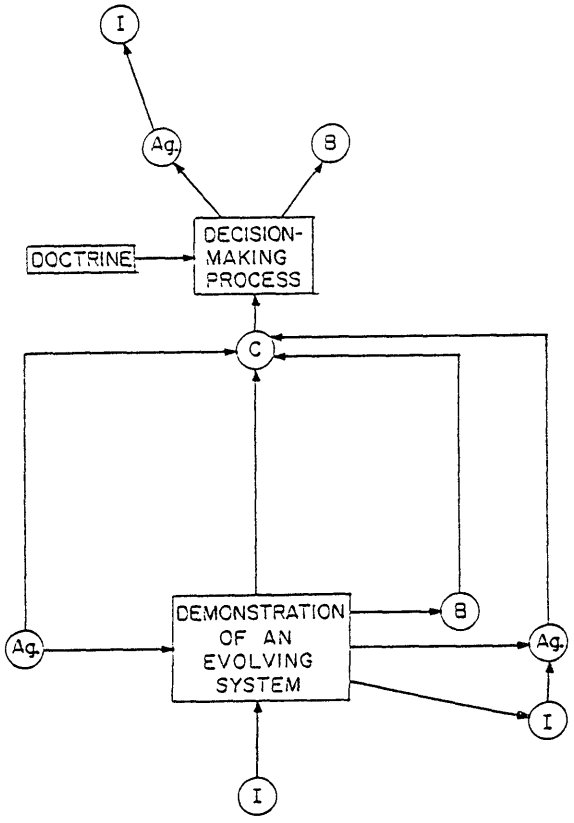


Figure 4. Organizational Interaction of Demonstration Participants

The contractors, denoted by I, provide the operational components of the system S, while the sponsor approves a scenario. All four participants observe the demonstration. The contractors report their observations and recommendations to the sponsors ($I \rightarrow A_g$). The users and the sponsor indicate their findings to the decisionmakers (group C). The sponsors, A_g, have already indicated to the decisionmakers the objectives of the demonstration. On the basis of their own observations and the inputs from the sponsoring agency and the users, the decisionmakers indicate their support for the program to the agency, and instruct the users to continue in assisting with the development and implementation of the system S.

Therefore, it is not inappropriate to express the utility of the demonstration as being that which is ultimately perceived by the decisionmakers. Indeed, the partial utilities u_A, u_B, and u_C result from the direct observation by the participants in groups A, B, and C, respectively, regardless of the interaction of those participants. After groups A and B report their observations to group C, the decisionmakers aggregate all three partial utilities in a global one. Hence, the global utility of the demonstration is an aggregation, by the decisionmakers, of the partial utilities of the developers, the system users, and the decisionmakers themselves.

$$u = u_C'(u_A, u_B, u_C) \tag{16}$$

Function u_c' can be a direct weighting of u_A, u_B, and u_C, as in expressions (14) and (15). In this case, the implication of the model is that weights a, b, and c are fixed by the decisionmakers.

3. THE DESIGN OPTIMIZATION PROBLEM

A system is most effective with regard to a mission if, operating in a given context, it is most likely to achieve those combinations of attribute values that are highly desirable; that is, if the points $\underline{x}$ for which $f(\underline{x})$ is high coincide with those for which the utility u is high. An effectiveness measure that expresses this notion is given by the expected utility, i.e.,

$$E_\pi(u) = \int_{\underline{x}} f_\pi(\underline{x})\, u(\underline{x})\, d\underline{x} \tag{17}$$

Expression (17) defines a functional which assigns a value to each useful configuration π; it is a measure of effectiveness of π with respect to the demonstration's goals. The design objective is then to maximize the effectiveness of the demonstration by selecting the appropriate configuration π:

$$E_\pi^*(u) = E^* = \max_{\pi \in \tilde{P}(t)} E_\pi(u) \tag{18}$$

The determination of π^* cannot be done analytically; each configuration must be evaluated and the corresponding values of the effectiveness measure rank ordered. The procedure is impractical, if $\tilde{P}(t)$ includes all $2^{\#T}$ configurations. However, if the design of the alternative system configurations has been carried out properly, only a few configurations need to be evaluated. The steps of the procedure for selecting the optimal configuration for the demonstration, shown in Figure 5, can be summarized as follows:

(a) For a given mission utility function u, and for the configuration π defining the probability distribution f_π, evaluate $E_\pi(u)$.

(b) Repeat step (a) for each configuration $\pi \varepsilon \tilde{P}(t)$.

(c) Rank order the configurations π in $\tilde{P}(t)$ according to the values of $E_\pi(u)$.

(d) Select the configuration that maximizes expected utility.

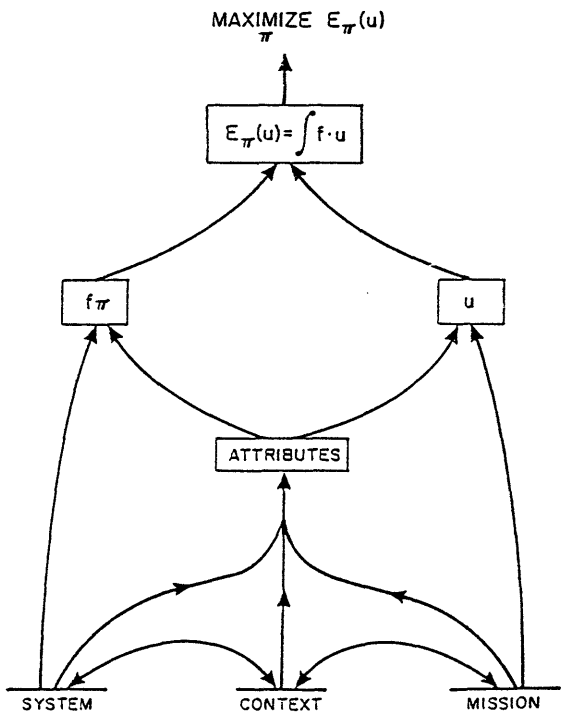


Figure 5. Methodology for Selecting the Optimal System Configuration

In order to implement this procedure, it is necessary to specify the functions v and w which define the partial utilities in Eqs. (11) to (13). These functions are given in the following form:

$$v_i(\underline{y}) = 1 - \frac{(\mathbf{1}-Y)^t\, Q_i\, (\mathbf{1}-Y)}{\mathbf{1}^t\, Q_i\, \mathbf{1}} \qquad i = A,B,C \qquad (19)$$

$$w_A(z_A) = (z_A)^\alpha \quad ; \quad w_C(z_C) = (z_C)^\gamma \qquad (20)$$

where Q_i is a matrix with all elements non negative, and

$Y^t = (y_1\ y_2\ \ldots\ y_n)$ row vector of the Type 1 attributes

$\mathbf{1}^t = (1\ 1\ \ldots\ 1)$

α and γ are real numbers between 0 and 1.

Specification of the partial utilities reduces to determining the three matrices, Q_A, Q_B, and Q_C, and the real numbers α and γ. This will be done in the context of an application - the effectiveness analysis of the METANET demonstration.

4. EFFECTIVENESS ANALYSIS OF THE METANET DEMONSTRATION

METANET can be described as a network of networks, where the objective is to demonstrate the feasibility of effective, reliable communication between a large heterogeneous set of nodes. Assume that a demonstration of some aspects of METANET is being planned. The plan is to freeze a set of components, select a set of nodes and links, and develop a scenario that will (a) demonstrate the capabilities and potential of METANET, and (b) indicate research and development needs (Mathis, 1983; Karam 1985).

4.1 The System Model

Fifteen components/technologies were frozen for use in the first demonstration of METANET; they constitute the set S(t) of operational components. These are:

Operational Technologies:

T_1 Tactical Situtation Assessment: performs part of the situation assessment function of C^2 and runs on operating system X.

T_2 Briefing Aid: allows a user to present briefings using computer graphics display hardware; runs on operating system X.

T_3 Weather Editor: allows a user to select a geographical area of the world and an environmental data field to be displayed; runs on operating system X.

T_4 Warfare Environment Simulator: provides a computer derived simulated naval war environment for both instructional and strategy testing purposes; runs on operating system X.

T_5 Local Area Network 1 (LAN1): generalized data communication network using data bus technology.

T_6 Multimedia Mail: to extend text mail, graphics, and vocoded voice; interactive interface with user connected to workstation, accessed from workstation (C^2WS).

T_7 Natural Language/Database: provides natural language access to Database (T_{10}), also includes the design and implementation of communication links among command and control workstations and Database; runs on workstation's computer.

T_8 Speech: to interface speech commands and queries to the Natural Language system, to synthesize responses from the query system into speech for the user; runs on workstation's computer.

T_9 METANET Gateway (GWY): to provide link between the workstations' local area network and other networks, including: LAN1, LAN2 (T_{12}), SANET (see T_{13}), and MILNET.

T_{10} Database: software system, allows a user to query multiple pre-existing, heterogeneous databases, using a single language and a simple integrated view of the available data.

T_{11} Data Management System (DMS): provides a graphical user interface to information, designed to be used directly by the decisionmaker; installed on board ship.

T_{12} Local Area Network 2 (LAN2): data communication network using ring technology.

T_{13} P-3C Radio Modifications: installation of a SANET (Satellite Network) node on a P-3 aircraft.

T_{14} SAT: enables linkage to SANET (see T_{13}).

T_{15} PLI: cryptographic device, enables linkage to MILNET.

Many system configurations can be obtained from these technologies, but not all are useful for the demonstration (Karam and Levis, 1984). The useful configurations are specified in conjunction with the possible scenarios in Section 4.4.

4.2 The Attributes

Six system attributes are considered relevant; they are defined so as to take values between 0 and 1. The traditional attributes are Reliability, Survivability, Input Flow, and Inverse Time Delay, and form the vector

$$\underline{y} = (y_1=R,\ y_2=S,\ y_3=F,\ y_4=\psi).$$

The novel attributes are the weighted fraction of components used and functions carried out; they form a vector

$$\underline{z} = (z_A,\ z_C).$$

Reliability denotes the capability of a network (see Section 4.4) to deliver a message from origin to destination when only the physical properties of the components are taken into account. In contrast, the attribute Survivability does not depend on the components' physical deterioration, but on the components' capabilities to resist enemy actions, e.g., jamming.

Let C be the capacity of any link in bits/sec. Assume the M/M/1 model of queueing theory and let $1/\mu$ be the mean packet size in bits/packet. If ϕ is the input flow on one link (packets/sec), then the mean time delay ξ for that link, which includes both queueing and transmission time, is:

$$\xi = \frac{1}{\mu C - \phi} \qquad (21)$$

Instead of time delay it is more convenient to consider its inverse. The scaled attributes are then:

$$\text{Input Flow:} \qquad F = \frac{\phi}{\mu C} \qquad (22)$$

$$\text{Inverse Time Delay:} \qquad \psi = \frac{1}{\mu C \xi} \qquad (23)$$

The Weighted Fraction of Technologies, z_A, and the Weighted Fraction of Functions, z_C, are given by expressions (6) and (7) where #T=15 and #F=4 (see Section 2.4).

4.3 The Mission Model

In this section, the participants in the demonstration of METANET are identified, and their expectations specified.

The Group of Developers (Group A): Six major developers can be identified (#A = 6): five system contractors and the sponsoring agency. Each developer contributed to the development of some or all the operational technologies (i.e., a subset of S(t)), and is particularly eager to see those demonstrated. This is expresed in terms of the technology by developer matrix, TA:

$$TA = \begin{bmatrix} 6/28 & 0 & 0 & 0 & 0 & 1/15 \\ 6/28 & 0 & 0 & 0 & 0 & 1/15 \\ 3/28 & 0 & 0 & 0 & 0 & 1/15 \\ 3/28 & 0 & 0 & 0 & 0 & 1/15 \\ 10/28 & 0 & 0 & 5/29 & 0 & 1/15 \\ 0 & 10/30 & 0 & 0 & 0 & 1/15 \\ 0 & 5/30 & 0 & 0 & 0 & 1/15 \\ 0 & 8/30 & 0 & 0 & 0 & 1/15 \\ 0 & 7/30 & 0 & 0 & 0 & 1/15 \\ 0 & 0 & 10/15 & 0 & 0 & 1/15 \\ 0 & 0 & 5/15 & 0 & 0 & 1/15 \\ 0 & 0 & 0 & 5/29 & 0 & 1/15 \\ 0 & 0 & 0 & 10/29 & 10/10 & 1/15 \\ 0 & 0 & 0 & 8/29 & 0 & 1/15 \\ 0 & 0 & 0 & 1/29 & 0 & 1/15 \end{bmatrix} \qquad (24)$$

Element $(TA)_{ij}$ reflects the extent to which developer j would like to see technology i demonstrated. Matrix TA was estimated by asking each developer j (contractors or the agency) to fill in column j, by rating all the technologies on a 0 to 10 scale, for example. The input data are normalized for each developer so that

$$\sum_{i=1}^{\#T} (TA)_{ij} = 1 \qquad \forall\ j = 1,\ldots,\#A \qquad (25)$$

The physical characteristic of the system's components and the context of the demonstrations dictate the following technology by attribute matrix, TY:

$$TY = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad (26)$$

Element $(TY)_{ij}$ is equal to one if the developers believe that a good performance of technology i, when used, depends on the values taken by attribute j; it is equal to zero otherwise. Developer i is concerned with the performance of attribute j insofar as attribute j is directly affected by those technologies which developer i would like to see demonstrated, and that these technologies are actually demonstrated. These ideas can be expressed by formulating the developer by attribute matrix as follows:

Let the elements $(AY)_{ij}$ be defined by

$$(AY)_{ij} = \sum_{k=1}^{\#T} \tau(T_k)\ (TA)_{ki}(TY)_{kj} \qquad (27)$$

where

$$\tau(T_k) = \begin{cases} 1 & \text{if technology k is included in the demonstration} \\ 0 & \text{otherwise} \end{cases}$$

Equation (27) can be written in matrix form

$$AY = (\underline{TA})^t\ (TY) \qquad (28)$$

where

$$(\underline{TA})_{ki} = \tau(T_k)\ (TA)_{ki} \qquad (29)$$

Thus, element $(AY)_{ij}$ denotes the degree to which developer i is concerned about the values taken by traditional attribute j. The developers' concern is contigent on the demonstration using "their" technologies. Finally, matrix Q_A in Eq. (19) can be determined by

$$Q_A = (AY)^t\ (AY) \qquad (30)$$

Parameter α in Eq. (20) is not easy to assess. In practice a parametric study would be done where α is varied from 0 to 1. This completes the specification of the utility function for Group A.

The Group of System Users (Group B): The utility of group B is a function of the Type 1 attributes only

$$u_B(\underline{x}) = v_B(\underline{y}) = 1 - \frac{(1-Y)^t\ Q_B\ (1-Y)}{1^t\ Q_B\ 1} \qquad (31)$$

The question then reduces to determining the weighting matrix Q_B. To do this, a matrix that relates system users to attributes needs to be introduced.

Let element $(BY)_{ij}$ denote the degree to which system user i is concerned with the values taken by attribute j. Matrix BY can be estimated by interviewing the system users individually. Each system user i is asked to fill in row i of matrix BY by rating all the Type 1 attributes on a scale of 0 to 10. The input data are then normalized for each system user, so that:

$$\sum_{j=1}^{\#Y} (BY)_{ij} = 1 \qquad \forall\ i = 1,\ldots,\#B \qquad (32)$$

Matrix Q_B is then equal to:

$$Q_B = (BY)^t\ (BY) \qquad (33)$$

Since the system users in this case are those persons who will use the system during the demonstration, and only those, the system user by attribute matrix BY is then

$$(BY)_{ij} = 1/4 \quad i=1,\ldots,\#B \ ; \ j=1,\ldots,4 \qquad (34)$$

i.e., all the participants in group B are equally interested in each of the four Type 1 attributes.

The Group of Decisionmakers (Group C): The decisionmakers are concerned that the demonstration "perform well"; however, their concern is conditioned by which functions are carried out. The utility of group C is

$$u_C(\underline{x}) = (z_C)^\gamma\ (1 - \frac{(1-Y)^t Q_C\ (1-Y)}{1^t\ Q_C\ 1}) \qquad (35)$$

Parameter γ is not easy to assess. In practice, a parametric study is done where γ is varied from 0 to 1. Matrix Q_C, however, can be written as

$$Q_C = (CY)^t\ (CY) \qquad (36)$$

where CY is the "decisionmaker by attribute" matrix. Element $(CY)_{ij}$ denotes the degree to which decisionmaker i is concerned about the values taken by Type 1 attribute j insofar as the functions he would like to see carried out are actually carried out, and the performance of these is contingent on the values taken by attribute j. Hence, it is not unreasonable to formulate $(CY)_{ij}$ as follows:

$$(CY)_{ij} = \sum_{k=1}^{\#F} \phi(F_k)\ (FC)_{ki}(FY)_{kj} \qquad (37)$$

where

$$\phi(F_k) = \begin{cases} 1 & \text{if function k is carried out in the demonstration} \\ 0 & \text{otherwise} \end{cases}$$

FC is called the "function by decisionmaker" matrix. Element $(FC)_{ij}$ expresses the extent to which decisionmaker j would like to see function i carried out. Matrix FC is determined by asking each decisionmaker j to fill in column j, by rating all the functions on a 0 to 10 scale. Then, the input

are normalized for each decisionmaker, so that:

$$\sum_{i=1}^{\#F} (FC)_{ij} = 1 \qquad \forall\, j = 1,\ldots,\#C \tag{38}$$

Matrix FY is called the "function by attribute" matrix. Element $(FY)_{ij}$ is equal to one if the decisionmakers believe that a good performance of function i is dependent on the values taken by Type 1 attribute j. It is equal to zero otherwise.

Equation (37) can be rewritten in matrix form

$$CY = (\underline{FC})^t\ (FY) \tag{39}$$

where

$$(\underline{FC})_{ki} = \phi(F_k)\ (FC)_{ki} \tag{40}$$

In this case, there are four decisionmakers (#C=4), while four functions can be carried out by the demonstration of METANET (#F=4). Decisionmakers 1 to 3 are commanders in the Armed Forces; they are the real system users. Decisionmaker 4 represents a decisionmaking entity. Function 1 and 3 correspond to the interactions between commanders 1 and 2, and commanders 2 and 3, respectively. Function 2 (respectively, function 4) denotes the interaction between commander 2 (respectively, commander 3) and his staff.

The function by Decisionmaker matrix FC is:

$$FC = \begin{bmatrix} 1 & 1/3 & 0 & 1/4 \\ 0 & 1/3 & 0 & 1/4 \\ 0 & 1/3 & 1/2 & 1/4 \\ 0 & 0 & 1/2 & 1/4 \end{bmatrix} \tag{41}$$

The first three columns of the matrix result directly from the interaction scheme described previously. For example, consider commander 2: he interacts with commanders 1 and 3, and also with his staff. Thus, he is eager to see how METANET will carry out functions 1, 3, and 2; hence the second column of matrix FC. Decisionmaker 4 is equally interested in seeing all four functions carried out. Hence the fourth column of matrix FC.

The decisionmakers unanimously believe that any of the four functions should be carried out with maximum reliability, survivability, and input flow of data, and with minimum time delay. The function by attribute matrix FY is then:

$$(FY)_{ij} = 1 \qquad i = 1,\ldots,4 \;\; ; \; j = 1,\ldots,4 \tag{42}$$

4.4 Scenario and Useful Configurations

Four facilities are available to house the METANET demonstration. An important set of hardware and software technologies can be made available at facilities 1 and 4. Facility 2 is the generator of weather data (DG), while facility 3 is a ship in the high seas. As it turns out, the use by the demonstration of facility 4 is a decision variable. Depending on whether three facilities (case a) or four facilities (case b) are used, the total network configuration will be slightly different (see Figures 6 and 7). It is assumed that facility 3, as well as the satellite (SANET) and P-3 nodes are in a hostile environment. Survivability is an issue for any technology using these nodes. (Hence the second column of matrix TY).

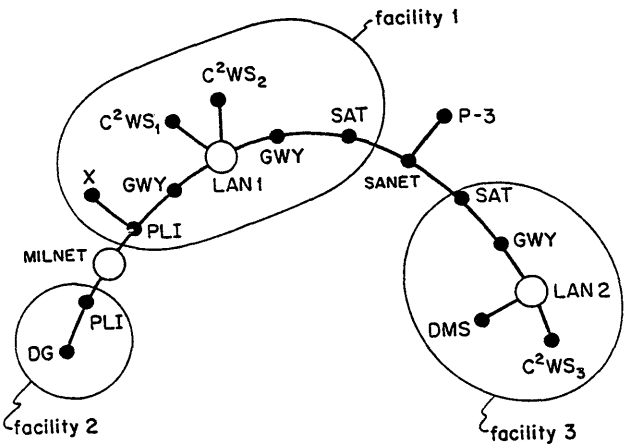


Figure 6. Total Network Configuration when Three Facilities are Used (Case a)

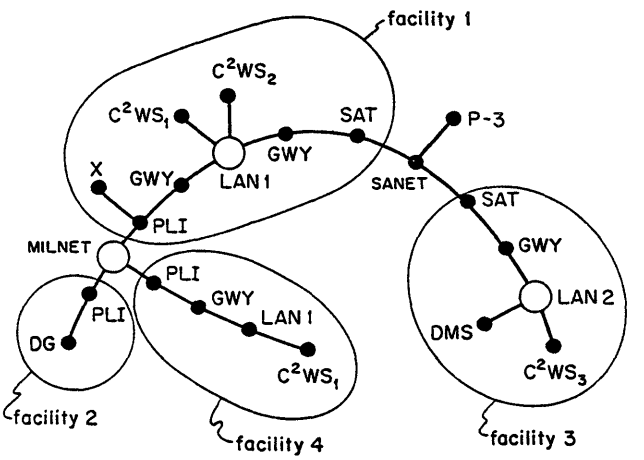


Figure 7. Total Network Configuration when Four Facilities are Used (Case b)

The scenario according to which the demonstration is run consists of several stages. An origin-destination pair, a session, is demonstrated at each stage; it performs one of the four functions described in Section 4.3. Seven sessions are identified. Session 1 is designed to carry out function 1. Sessions 2 and 3 execute function 2 each. Function 3 is carried out by session 4, while sessions 5, 6, and 7 carry out function 4.

All seven sessions do not have to be included in the demonstration: If s sessions are actually demonstrated ($1 \leq s \leq 7$), then the scenario consists of s stages. A useful system configuration corresponds to each such scenario; it includes the s origin-destination pairs. There are, $2^7-1 = 127$ (the null element ϕ is excluded) useful configurations in case a, and just as many in case b. Sessions 1 to 7 are drawn in Figure 8. Note that only session 1 has a different topology depending on whether three or four facilities are used.

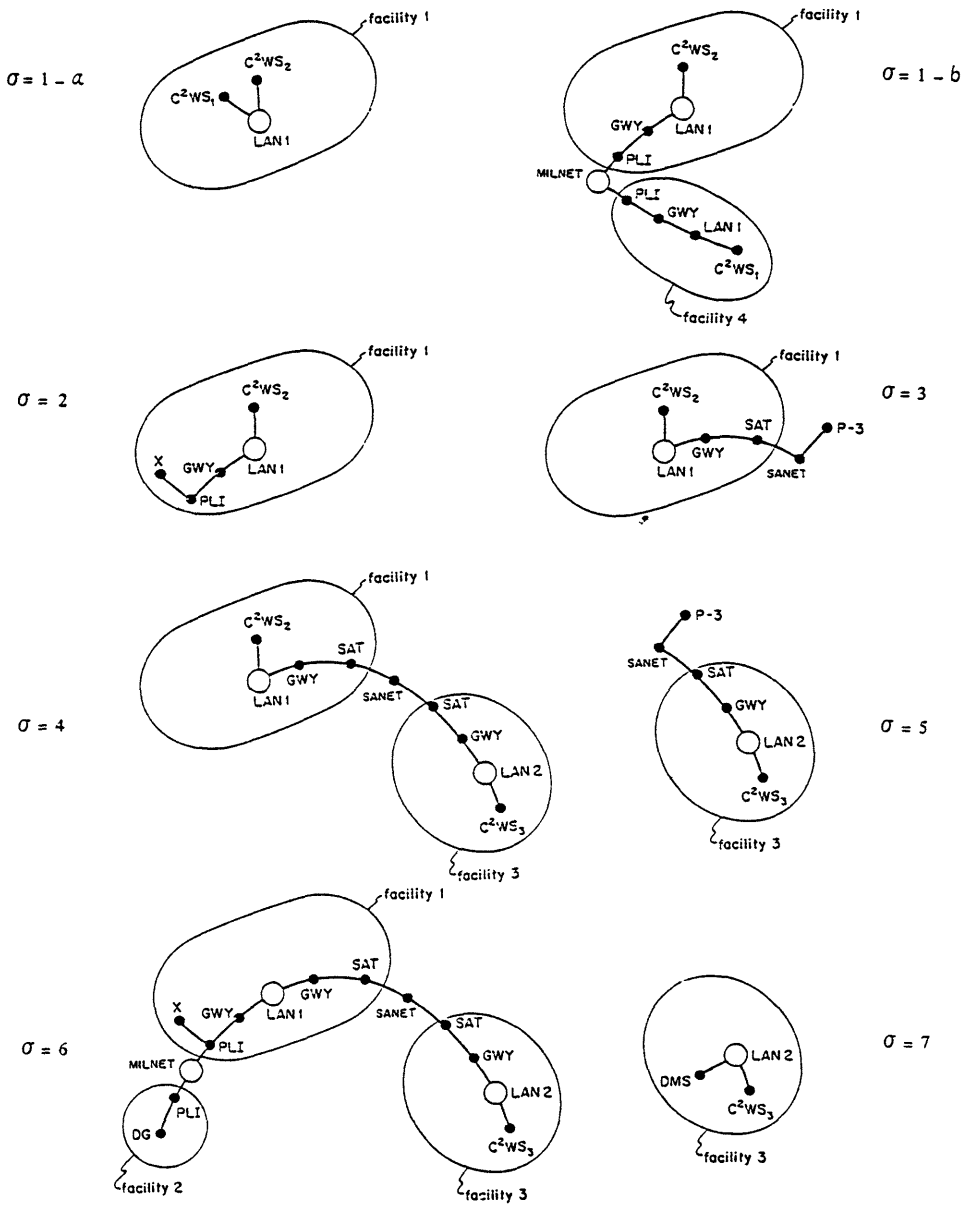


Figure 8. Topology of the Sessions

Each useful configuration π_K is characterized by the value taken by the binary variables $K(\sigma)$ for $\sigma = 1$ to 7, defined as follows:

$$K(\sigma) = \begin{cases} 1 & \text{if session } \sigma \text{ is included in configuration } \pi_K \\ 0 & \text{if it is not} \end{cases} \qquad (43)$$

For example, configuration $\pi_{(1100001)}$ is the one that includes sessions 1, 2, and 7.

The measure of effectiveness of a demonstration of METANET using π_K is:

$$E(K) = a(z_A(K))^{\alpha} \int g_{\pi_K}(\underline{y})\, v_A(\underline{y})\, d\underline{y} + b \int g_{\pi_K}(\underline{y})\, v_B(\underline{y})\, d\underline{y}$$

$$+ \; c(z_C(K))^{\gamma} \int g_{\pi_K}(\underline{y})\, v_C(\underline{y})\, d\underline{y} \qquad (44)$$

The probability distribution g is well defined when configuration π_K contains only one origin-destination pair. Let then

$$E_i(\sigma) = \int g_\sigma(\underline{y})\ v_i(\underline{y})\ d\underline{y} \qquad (45)$$

where i = A, B, or C and $\sigma = 1,\ldots,7$. For each useful configuration π_K, let $\bar{E}_i(K)$ be the average of the $E_i(\sigma)$'s for all sessions σ included in configuration π_K, i.e.,

$$\bar{E}_i(k) = \sum_{\sigma=1}^{7} K(\sigma)\ E_i(\sigma) \ / \ \sum_{\sigma=1}^{7} K(\sigma) \qquad (46)$$

Expression (46) replaces the term $\int g_\pi(\underline{y}) v_i(\underline{y}) d\underline{y}$ when configuration π_K contains more than one session. The measure of effectiveness of a demonstration of METANET using π_K is then:

$$E(K) = a\ (z_A(K))^\alpha\ \bar{E}_A(K) + b\ \bar{E}_B(K) + c\ (z_C(K)^\gamma \bar{E}_C(K) \qquad (47)$$

The design optimization problem becomes:

Maximize E(k) over k=(k(1),...,k(7) for cases a and b.

In order to solve the design problem, it is necessary to specify a number of design parameters.

Weights $\omega_A(T_i)$ and $\omega_C(F_i)$ used in expressions for the Weighted Fractions of Technologies and Functions, Eqs. (7) and (8) are given are by:

$$\omega_A(T_i) = \sum_{k=1}^{6} (TA)_{ik} \qquad i = 1,\ldots,15 \qquad (48)$$

$$\omega_C(F_j) = \sum_{k=1}^{4} (FC)_{jk} \qquad j = 1,\ldots,4 \qquad (49)$$

On the other hand, a technology i is said to be included in a configuration π_K (i.e., $\tau(T_i)=1$) whenever it is used by at least one session in that configuration. Similarly, a function j is said to be carried out by the demonstration ($\phi(F_j)=1$) if it is executed by at least one session in configuration π_K.

Reliability and Survivability depend on the probability of failure of of the components. Each failure probability is allowed to vary in a different interval of [0,1], depending on whether Reliability or Survivability is computed. Hence, for each session σ

$$R_{min}(\sigma) \leq R \leq R_{max}(\sigma) \qquad (50)$$

$$S_{min}(\sigma) \leq S \leq S_{max}(\sigma) \qquad (51)$$

For each session σ, the time delay between origin and destination is

$$\xi = \frac{L(\sigma)}{\mu C - \phi} \qquad (52)$$

where $L(\sigma)$ is the number of links in session σ between the origin and the destination. Using the scaled attributes Eq. (52) becomes

$$\mu = (1-F)/L(\sigma). \qquad (53)$$

Finally, the utility of the demonstration is assumed to be an additive average of the partial utilities, as given by Eq. (14). The Q matrices can be computed (for each session σ) by manipulating the data matrices given in this section. Parameters α and γ are set equal to 0.5, while coefficients a, b, and c in Eq. (14) are set equal to 1/3. Sensitivity analyses of the solution with respect to α, γ, a, b, and c were presented in Karam (1985) and Karam and Levis (1984).

4.5 Results

For each session σ, the quantities $E_A(\sigma)$, $E_B(\sigma)$, and $E_C(\sigma)$ were computed. The effectiveness of each configuration π_K was then computed according to Eq. (47). For each case (a or b), the configurations were then rank ordered. The results are given next.

Case a: Three Facilities

The best (first) ten configurations are listed in Table 1 in order of decreasing effectiveness. Each configuration π_K is identified by the values of the binary variables $K(\sigma)$, σ= 1 to 7. For example, the configuration that ranks first includes all sessions but sessions 5 and 6, has a measure of effectiveness of 0.799, and a z_A and z_C equal to 0.98 and 1, respectively. Table 1 gives also the values of the system attributes z_A and z_C. Several remarks can be said about the results shown in this table.

Table 1. The First Ten Configurations (Case a)

Rank	Configuration π_K σ = 1 2 3 4 5 6 7	Effectiveness	z_A	z_C
1	1 1 1 1 0 0 1	0.799	0.98	1
2	1 1 0 1 1 0 1	0.793	0.98	1
3	1 1 1 1 1 0 1	0.787	0.98	1
4	1 0 1 1 0 0 1	0.785	0.83	1
5	1 1 1 1 0 1 1	0.782	1	1
6	1 1 0 1 1 1 1	0.778	1	1
7	1 1 0 1 0 0 1	0.777	0.75	1
8	1 1 1 0 0 0 1	0.777	0.98	0.73
9	1 1 1 1 1 1 1	0.775	1	1
10	1 0 1 0 0 0 1	0.771	0.83	0.73

First, the configuration including all sessions (K=(1 1 ... 1)) is not the optimal one, it ranks #9. The interpretation is the following: some sessions had better be ignored altogether in the first demonstration of METANET if they are not adequately developed, specially if they do not execute an additional function. It can be noted, with this respect, that the first seven configurations carry all four functions (z_C=1). However, configuration #8 has a z_C of 0.85: there is at least one function which is carried out by none of the sessions included in this configuration. Configuration #8 carries out fewer functions than configuration #9 (smaller z_C) and includes fewer technologies (smaller z_A); neverthe-

less, it is more effective for the first demonstration of METANET. In fact, all first four configurations have a z_A smaller than 1; i.e., none of them includes all fifteen technologies.

Case b: Four Facilities

The same type of results is obtained when four facilities are used, and hence the same conclusions can be drawn. Table 2 shows the first ten configurations, together with their effectiveness measure, and the values of system attributes z_A and z_C.

Table 2. The First Ten Configurations (Case b)

Rank	Configuration π_K σ = 1 2 3 4 5 6 7	Effectiveness	z_A	z_C
1	1 1 1 1 0 0 1	0.775	0.98	1
2	1 1 0 1 1 0 1	0.770	0.98	1
3	1 1 1 1 1 0 1	0.768	0.98	1
4	1 1 1 1 0 1 1	0.762	1	1
5	1 1 1 1 1 1 1	0.758	1	1
6	1 1 0 1 1 1 1	0.758	1	1
7	1 0 1 1 0 0 1	0.757	0.83	1
8	1 1 0 1 0 0 1	0.750	0.75	1
9	1 1 1 0 0 0 1	0.749	0.98	0.73
10	1 0 1 1 1 0 1	0.748	0.83	1

Note that the configuration including all sessions now ranks fifth, and that its effectiveness is reduced compared to case a. In fact, all configurations that include the first session ($\sigma = 1$) have their effectiveness reduced, if four facilities are used rather than three. When the mission primitives were given the following extreme values

$$a = 1 \quad , \quad b = c = 0$$

$$\alpha = 1 \quad , \quad \gamma = 1$$

this basic result remained unchanged: the top ranking configurations were still more effective when three facilities are used rather than four. The conclusion is then the following: given the values of the system primitives, the model predicts that, for the first demonstration of METANET, it will always be more effective to use three facilities, and sessions 1, 2, 3, 4, and 7 with the corresponding scenario.

It can be inferred from these results that showing an additional technology or carrying out an additional function at the time of the METANET demonstration may be at the expense of the overall effectiveness of such a demonstration. The model developed in this paper does not explicitly address the issue of optimally designing the series of demonstrations to come. It is expected, however, that the results obtained here will be more useful, if the next to the first demonstration were considered in the model.

5. CONCLUSIONS

A methodology for effectiveness analysis of an evolving system has been presented. It requires the explicit specification of candidate technologies and the consideration of the utilities of the various groups involved in developing the system. The context in which the methodology was formulated is that of a demonstration aimed at showing the progress achieved in developing the system as well as the capabilities of the latter. The methodology provides the decisionmaker with a powerful tool that can be applied systematically to quantifying the progress made in developing a system, the expectations of the various participant groups, and finally the global effectiveness of the system at each point in time.

6. REFERENCES

Bouthonnier, V. and A. H. Levis, (1984) "Effectiveness Analsyis of C^3 Systems," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-14, No. 1.

Mathis, J. E., Ed. (1983) "Presentations From the NAVELEX Technology Meeting," NAVELEX Command Systems Note #1, SRI International, Menlo Park, CA.

Karam, J., and A. H. Levis, (1984) "Effectiveness Assessment of the METANET Demonstration," Proc. of the 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Karam, J. G., (1985) "Effectiveness Analysis of Evolving Systems," M.S. Thesis, LIDS-TH-1431, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

AN EXPERIMENTAL PLAN FOR STUDYING DISTRIBUTED TACTICAL DECISIONMAKING

Daniel Serfaty and David L. Kleinman
CYBERLAB

Department of Electrical Engineering and Computer Science
The University of Connecticut
Storrs, CT 06268

ABSTRACT

As a part of a comprehensive effort to find a new framework for multihuman decisionmaking problems, we have developed a novel experimental research paradigm involving human teams in decisionmaking tasks. The paradigm focuses on the problems of distributed resource management and task processing in an uncertain dynamic environment. The task environment is an abstraction of a Naval Battle Group Command, Control and Communications (C^3) system in which a number of geographically separated commanders must make coherent decisions based on decentralized information. The paradigm is flexible enough to be tested across a large range of experimental conditions in which the main independent variables are: the team configuration, the team information and communication structure, the uncertainty level in both inputs and consequences of action, the level of expertise and functional overlapping between the different decisionmakers. The Distributed Dynamic Decision (D^3) paradigm although primarily a tool to measure team performance and effectiveness, will also be used to study various cognitive factors that have found empirical evidence in the literature. Attempts to construct parts of an integrated model with ideas from queueing networks, team theory, distributed estimation and decentralized resource management are described. Future development of these models of human team behavior depends strongly on the availability of data to be provided by the paradigm.

I. INTRODUCTION

The study of distributed information processing and decisionmaking is presently hampered by two factors: (i) The inherent complexity of the mathematical formulation of decentralized problems (control, detection, data fusion, etc.) has prevented the development of efficient and practical theoretical models that could be used to predict actual performance in a distributed environment [1], [2]. (ii) The lack of comprehensive scientific empirical data on human team decisionmaking has hindered the development of significant descriptive models. Most of the organizational behavior and applied psychology research in the field focuses on centralized group decisionmaking rather than on team decisionmaking in which the element of decentralization is essential [3], [4].

A purely normative approach to the development of models of team information processing and decisionmaking would further have the disadvantage of not representing actual human performance. A purely descriptive approach would have the disadvantage of not providing a predictive capability for hypothetical situations in which there exist no directly applicable data. A normative - descriptive approach, constraining the normative solution by empirically determined cognitive and perceptual characteristics, aims to provide realistic predictions of human performance. There are two integrated facets to this approach. The main emphasis of the descriptive portion of the research methodology is placed upon characterizing and interpreting the constraints imposed by the decisionmaker (DM). The normative portion of the research provides a baseline for analytical modeling, with the optimal solution being a starting point rather than a goal. It is this iterative approach to model-building via extension that imparts a deductive science to normative-descriptive modeling. However, one should note that model-building need not await the "optimal" solution. Cosntraints may be introduced and their effects on performance may be systematically investigated at any point, even before discrepancies between actual and ideal performances are detected. Indeed, in many situations, solving the initial "optimization" problem with psychologically interpretable mathematical constraints may not only result in a simpler mathematical formulation, but also reduce the gap between what human decisionmakers "do" with what they "should do". Such an approach to human decisionmaking modeling has been successfully developed for the Dynamic Decision Model (DDM) by Pattipati and Kleinman [5]. The DDM is a model for decision problems involving task selection and sequencing and has been tested with success across a large range of experimental conditions.

The type of problems for which we now seek to apply the normative-descriptive modeling methodology are those of distributed Naval Tactical Decisionmaking. These problems involve a team of geographically separated commanders who must make coherent decisions by managing and assigning resources to process tasks in a dynamic, uncertain environment. This complex team decision problem has some features that were not incorporated in the DDM or any other decision model. (i) The task selection and sequencing problem is a distributed decisionmaking problem involving multihuman interactions (ii) In view of the input uncertainty associated to the task environment, human inference limitations in assessing probabilities of events are an inherent part of the information processing problem. (iii) Due to the limitation on resources available, a dynamic resource management problem is created in which human real time planning limitations are a central element. Thus, the approach taken must be multidisciplinary, combining new results from 1) the behavioral sciences, 2) the systems sciences with 3) the empirical results that emerge from experimental research. The normative-descriptive research that we propose integrates all three areas in an iterative manner, wherein mathematical models are developed, tested against data, and refined through adjustment/ modification of their descriptive elements [Fig. 1]. Relevant descriptive features to be included in the model are described in the next section. The experimental plan and the paradigm that has been developed to support it are the subject of section III. Elements of the normative-descriptive Distributed Dynamic Decision model are briefly described in section IV.

II. DESCRIPTIVE ELEMENTS

Our analysis of the literature in cognitive psychology, behavioral sciences, group psychology etc., indicates that the human cognitive characteristics of major concern fall loosely into either the categories of information processing (IP) or decisionmaking (DM). Virtually all of these characteristics describe different (although not necessarily independent) facets of

human response under uncertainty. The IP limitations deal with input uncertainty (e.g., judgement mechanisms in analyzing and interpreting information), while the DM limitations deal with consequence-of-action uncertainty (e.g., outcome evaluations and choice mechanisms). These categories may be further delineated as dealing with individuals or with a team of individuals. In the latter case our concern is on how individual judgements and choices are modified by the presence of co-acting humans.

1. Human Information Processing Characteristics

A large number of contemporary studies in cognitive psychology have uncovered various heuristics and biases (i.e., deficiencies) that single DMs apply in interpreting and aggregating information. (See Slovic, Fishoff, and Lichtenstein [7], Sage and White [8] for comprehensive reviews.) Of these, the judgemental deficiencies of misperception, representativeness and availability appear to be the most prominent for probability assessment. Conservatism/recency along with anchoring and adjustment are most prominent for human revision of opinion.

MISPERCEPTION: This generally refers to the mismatch between subjective and actual probability distributions, misrepresentations of base-rate, and the tendency to place greater belief in values closer to the mean.

REPRESENTATIVENESS: This heuristic, as advanced by Tversky and Kahneman [9], refers to the human's inclination to evaluate the probability of an event on the basis of the degree of similarity between the event and the evidence they have examined to date.

AVAILABILITY: This heuristic pertains to the finding that humans evaluate the probability of an event on the basis of the ease with which instances or occurrences involving such events can be recalled or imagined.

CONSERVATISM AND RECENCY: Conservatism, as advanced primarily by Edwards and his associates [10], refers to the nonoptimal sequential revision of subjective probabilities; whereby new information is not given as much credibility as Bayesian decision theory would predict. That is, the posterior probabilities estimated by subjects are generally (but by no means universally) nearer to the prior probabilities than those obtained via Bayes' rule. Recency refers to the human's tendency to put more credibility on new information than is estimated by theory.

ANCHORING AND ADJUSTMENT: With this heuristic, an initial value or anchor is used as a first approximation to the judgement. The initial value is then adjusted according to the information provided. Humans tend to overestimate the probability of conjuctive events and underestimate the probability of disjunctive events.

Of the above human deficiencies, only conservatism/recency and misperception are amenable to quantitative modeling at present. The modeling can generally be accomplished by modifying the objective probability distributions as perceived by the subject, by use of a modified Bayes' rule or Kalman filter, etc.

TEAM ASSESSMENT OF PROBABILITIES: The information processing characteristics of groups describe how individual judgements become modified by information transmitted to (or received from) other numbers of the group. Studies [3] have shown a few interesting phenomena: "group think" or overconfidence in the group opinion, "conformity" or decrease in initial confidence is individual judgement, group polarization, etc.

MUTUAL ASSESSMENT OF OPINION: The way in which a DM interprets and utilizes the stated judgements of another team member is discussed in Roby [4]. The evidence shows that the relative weight on the information depends upon the partner's expertise in the area that the data is associated with, the intrateam trust, the specific judgemental process of the partner in a similar task, and the partner's expressed confidence in his judgement. How well groups can identify and weight their "best" members is a fundamental question that involves the pooling of individual biases as examined by Einhorn et al., [11].

2. Human Decisionmaking Characteristics

The primary constraints that affect single human decisionmaking are due in large part to his limited combinatorial capabilities and inherent randomness or subjectivity in the interpretation of value and/or success probabilities.

MYOPIA: This involves the inability of the human DM to project the effects of a potential decision far into the future. Thus, options are evaluated using only a short time horizon which generally does not go beyond the next expected event. The mathematical treatment of a myopic decision strategy can be handled by the imposition of a finite (M-step) horizon in a normative model [5], or via the use of a discounting function.

CONSTRAINED BEHAVIOR: The hypothesis is that the human is not able, due to his inherent processing limitations, to evaluate all of the alternatives and thus selects the (first) one that will satisfy some minimal acceptance threshold. The observed result is generally referred to as "satisficing" behavior [12]. In a normative modeling context, these limitations can be treated as constraints on the solution process, or as constraints on the decision set.

DECISION RANDOMNESS: Humans fluctuate in their response selection to the same stimulus, even when there are no changes in their information or resources. Randomness in choice can arise because the human is unable to discriminate precisely among utilities or values or somewhat equally attractive alternatives.

BIAS IN SUCCESS PROBABILITY: Humans tend to bias their estimated chance of success or failure. The factors that contribute to this are much the same as those discussed under information processing, and deal with the strategies people use when having to assign probabilities to uncertain outcomes.

TEAM UNCERTAINTY AVOIDANCE AND PLANNING: [3], [4]. Team members will avoid planning (behave myopically) if the plans depend on predictions of uncertain future events, but will emphasize planning if the plans can be made self confirming through action outcomes. From a modeling viewpoint, these limitations/characteristics seem to involve a link between short-term planning (use of limited planning horizons), and the issues of probing (information seeking) and caution (risk avoidance).

RESOURCE SHARING/TRANSFERRING: Three main characteristics are displayed by humans with respect to their use of resources. [4]: (i) ownership: decisionmakers are more inclined to keep their assets for their own tasks rather than to distribute them to other DMs in the team. (ii) recall: DMs tend to forget the resource requests of others if they are not responded to immediately (they exhibit 50% more recall on their own requests!) and (iii) over-request: humans tend to over-request by asking more than they need or by overusing communications in asking several team members simultaneously.

III. EXPERIMENTAL PROGRAM

The major goals of the experimental program carried by the Distributed Dynamic Decision paradigm include the generation of data for the following purposes.

Model Development. The process of building any model for describing human/team performance requires data to substantiate assumptions, suggest probable cause-and-effect relations, and test hypotheses. This is the major impetus that drives our experimental plan.

Quantifying Descriptive Features. A rich set of measured dependent variables (DVs) will provide data not only on performance, but on how this performance was attained. The parameters in submodels for various human/team limitations, such as those discussed in the previous subsection, can be identified, in principle, from this information.

Model Validation. The normative-descriptive models that

are developed should be validated to the extent possible given the experimental data. Generally, this will mean comparing model predictions versus experimental results using data that is different from that which was used to develop the model.

The research paradigm has been described in detail in [13]. Briefly, the paradigm is an abstraction of the Naval Battle Force/Battle Group Tactical decisionmaking environment in which team members (commanders) may be geographically distributed but must coordinate their management of limited resources for situation assessment and threat prosecution under uncertainty. The task environment on enemy actions is presented to each decision maker through graphical and alphanumerical displays providing data on task status, task attributes, resources, opportunity windows and processing time for each task. Decision aids as well as mailbox-type communication capabilities are also available to the decisionmakers. The DMs can use their resource for task prosecution or for information gathering (uncertainty reduction) and under certain experimental conditions may request or send resources to other team members.

There are a large number of independent variables associated with the paradigm that can be categorized as either structural or parametric variables. The structural variables deal with team organizational structure and are the ones that relate most closely to the fundamental issues in distributed C^3 system design. The parametric variables are all of those that enter into the various sub-models within the paradigm (communication delay, task variables, numbers and types of resources, etc.). Clearly, an experimental program that was to examine the effects of all IVs upon system performance would be infeasible. Therefore, we have greatly narrowed the set of experimental IVs to those that we believe are most critical to advance our understanding of distributed decisionmaking in a Naval BG/BF context. They are primarily of the structural variety.

TEAM STRUCTURE: This relates to the number of team members and their organizational/authority arrangement. A review of the literature in group dynamics (for example see [3]), has shown that organization within a group, or team structures, is a major factor influencing intra-team communication patterns. Three levels are suggested - Dyadic: two-person teams where both members are equal in status or authority; - Hierarchical: three-person teams where there is one designated leader (or commander) and two team members having equal authorithy; - Parallel: three-person teams where all members have equal authority. The ordering of these levels is by increasing complexity in terms of the communication patterns that are likely to occur, and the decision-making procedures that the team might adopt. The hierarchical case has an inherent mechanism for conflict resolution that is not present in the dyadic case. The parallel structure permits self-organizing and adaptive responses more so than do the others, but with a potential for greater conflict.

INFORMATION STRUCTURE: The information that team members obtain, either through their data-gathering network, or via communication with other team members, is the most single important factor that influences team performance. Data about the environment can range from decentralized (local data) to centralized (shared, common data). Levels between these two extremes are possible, e.g., local data augmented with an aggregation or partial set (i.e., overlap) of other team member's data. At the same time, direct communication among team members can be possible or impossible.

Five levels of information structure are suggested. - Decentralized data, no communication; - Centralized data, no communication; - Partially overlapping data, no communication; - centralized data, limited communication; - Partially overlapping data, limited communication.

DIVISION OF RESPONSIBILITY: This variable refers to the way that responsibility for task prosecution can be distributed among the team members. The major effects of this variable will be manifested in the coordination and bargaining that are required by the organization to resolve conflicts. Responsibility can be assigned in many ways.

The most interesting cases occur when different decisionmakers have overlapping responsibility on certain tasks while being able to process then with different levels of expertise or efficiency.

INPUT UNCERTAINTY/DIFFICULTY: Input uncertainty is manipulated in our experimental design via the choice of input scenario, and the selection of parameters in the various probability distributions for generating task information, etc. The input scenarios represents our simulation of the "enemy", and is the driving stimulus to the team. The input scenarios provide the means with which to compare and evaluate different combinations of the above structural independent variables. As such, we desire a set of stimuli that represent a range of input uncertainty (as well as task distribution, tempo, etc.). The set would be applied across all experimental conditions in order to maintain a common reference when discussing measured performance. The selection of a set of 10 to 15 scenarios is done empirically by varying, among other things: the mix of task type, the proportion of neutral task, the probability distribution of the task attributes and their overlapping, the proportion of unknown task types and unknown attributes, the measurement errors on attributes and processing times, the tempo of the task arrivals and the distribution of the tasks information among the decisionmakers.

A brief list of the set of dependent variables that will be collected in the various experiments is given in Table 1. These measures will be further aggregated into global measures of performance (MOP) for the team.

TABLE 1. DEPENDENT VARIABLES

Task Volume Measures
Number of tasks performed
Number of tasks left unperformed
Number of tasks successfully performed
Number of tasks unsuccessfully performed
Number of tasks acted on with partial information
Reward Measures
Team Score
Subject score
Number of assists
Rate at which points are lost
Reaction Time Measures
Initial reaction time
Attack reaction time
Information-attack onset
Safety margin
Average time taken to do a task
Time to resolve unknown tasks
Resource Measures
Over/under assignment of resources
Amount of resources requested/transferred
Resource utilization rate
Frequency of conflicts
Frequence of conflict resolution
Communication Measures
Number of communications per subject
Rate of communications
Communication cost (usage)
Resource/message

IV. SOME MODELING ASPECTS

The empirical research discussed in the previous section will provide the data for model building and validation. The modeling process proceeds in an iterative manner, wherein mathematical models are developed, tested against data, and then refined through adjustment of their descriptive elements. The mathematical research issues associated with modeling the class of problems captured in the Distributed Dynamic Decision paradigm are highly nontrivial. Each organizational structure (i.e., combination of team structure, information structure and division of responsibility) results in a different normative decision problem, and hence, with the addition of human cognitive characteristics, a different normative-descriptive decision model.

The first simplification of model building for the Distributed Dynamic Decision problem is to decompose it into two processes: (i) information processing (hypothesis generation and evaluation) and (ii) decision-making (options evaluation and selection). This dichotomy is a natural approach to interface results from behavioral decision theory and normative decision models and has been successfully used in the DDM. Since model development is still in its on-going stages and will be the subject of subsequent papers, we will not describe here the detailed aspect of the model. Subsequent research will show how various submodels should merge into an integrated Distributed Dynamic Decision model for prediction of Human Team Performance.

V. CONCLUSION

Problems associated with distributed decision-making and information processing are inherently complex. The analytical instruments used in model building come from a large array of disciplines and a major effort is required to integrate them into a comprehensive Distributed Dynamic Decision model. Moreover, superimposing the human team aspects is itself an ambitious task, since few results are available on the behavior and performance of human decisionmakers in a decentralized uncertain environment. Nevertheless, it is believed that the normative-descriptive approach outlined here is the key to gain insight into this complex issue. The Distributed Dynamic Decision (D^3) paradigm we have developed is a novel multifaceted experimental and analytical tool designed to investigate team behavior and evaluate team performance across a large range of experimental conditions. It clearly focuses on Naval Tactical C^3 - motivated problems of distributed task processing and resource management under uncertainty. By appropriately manipulating variables such as team topology, information structures, communications, expertise and environmental uncertainty we can hope to gain more insight into team behavioral aspects such as planning, risk taking, information resource sharing, communication use, and the synergistic mechanisms by which a team of experts merge into an expert team of experts [14]. The paradigm is a research tool that helps to focus mathematical efforts, provides an experimental media for examining the descriptive and performance aspects of team response and generate data for model building and validation. It is flexible, generalizable, manageable and meets the normative - descriptive requirement of analytic tractability.

REFERENCES

1. Sandell, N.R., Jr., P. Varaiya, M. Athans, and M.G. Safonov, "Survey of Decentralized Control Methods for Large Scale Systems," IEEE Transactions on Automatic Control, Volume AC-23, 1978, pp. 108-128.
2. Tsitsiklis, J.N. and M. Athans, "On the Complexity of Decentralized Decision Making and Detection Problems," IEEE Transactions on Automatic Control, Volume AC-30, 1985, pp. 440-446.
3. Shaw, M.E., Group Dynamics: The Psychology of Small Group Behavior, McGraw-Hill eds., 1976.
4. Roby, T.B., Small Group Performance, Rand McNally and Co., 1968.
5. Pattipati, K.R., D.L. Kleinman, and A.E. Ephrath, "A Dynamic Decision Model of Human Task Selection Performance", IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-13, 1983, pp. 145-166.
6. Serfaty, D. and D.L. Kleinman, "Multi Human Decision Making: A Control Theoretic Approach", Proceedings of the IEEE Conference on Decision and Control, December 1982, pp. 561-566.
7. Slovic, P., B. Fishoff, and S. Lichtenstein, "Behavioral Decision Theory", Annual Review of Psychology, Volume 28, 1977, pp. 1-39.
8. Sage, A.P. and E.P. White, "Methodology for Risk and Hazard Assessment", IEEE Transactions on Systems, Man, and Cybernetics, Volume 10, No. 8, August 1980, pp. 425-446.
9. Tversky, A. and D. Kahneman, "Judgement Under Uncertainty: Heuristics and Biases", in Utility, Probability, and Human Decisionmaking, Wendt and Vlek eds., Boston, Massachusetts: Reidel, 1975, pp. 141-162.
10. Edwards, W., "Conservatism in Human Information Processing", in Formal Representations of Human Judgement, B. Klunmuntz eds., Wiley, New York, 1968, pp. 17-52.
11. Einhorn, H.J., R.M. Hogarth, and E. Klempner, "Quality of Group Judgement", Psychological Bulletin, Volume 84, No. 1, 1977, pp. 158-172.
12. Simon, H.A., "A Behavioral Model of Rational Choice", in Models of Man, Wiley, 1957, pp. 241-260.
13. Kleinman, D.L., D. Serfaty and P.B. Luh, "A Research Paradigm for Multi Human Decision Making", Proceedings of the American Control Conference, San Diego 1984, pp. 6-11.
14. Athans, M., "The Expert Team of Experts Approach to Commanded Control (C^2) Organization", IEEE Control Systems Magazine, Volume 2, No. 3, September 1982, pp. 30-38.

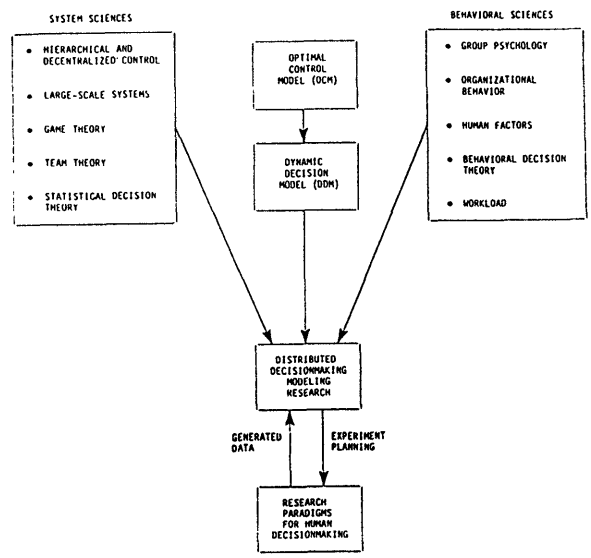


Figure 1. Disciplines Involved in the Development of a Modeling Framework for Multihuman Dynamic Decisionmaking

ON THE ANALYSIS AND DESIGN OF HUMAN INFORMATION PROCESSING ORGANIZATIONS†

Kevin L. Boettcher and Robert R. Tenney
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

The design of human organizations where members perform routine tasks under the pressure of time is considered. A three-phase approach is outlined. In the first phase, normative decision rules that specify ideal human behavior are obtained. In the second phase, implementations of these decision rules are devised, and descriptions of actual human behavior and workload are developed. Finally a third phase integrates design elements by placing parameters of the implementations for best organization performance, subject to individual member workload limitations. To illustrate the approach, a specific design problem is considered.

I. INTRODUCTION

To accomplish tasks that are too complex for individuals, humans have devised and evolved a variety of organizational structures. Despite their proliferation, however, organizations have not readily yielded to the development of rigorous methods of analysis and design. This is due in part to the inherent complexity of situations where individuals are required to coordinate their efforts so that some overall objective is achieved. Another factor is the necessity to assess whether individuals within the organization are capable of doing their assigned jobs; that is, whether their induced workload is within their limits. This paper presents an approach to organization analysis and design that is appropriate for a particular class of organizations. Specifically, consideration is focused on those organizations that (a) involve routine human information processing tasks, (b) incorporate a well-defined organizational goal that is held by all members (i.e. the organization is a team), and (c) have a short amount of time available for individual information processing tasks (e.g. a few seconds or minutes). Organizations that are of this class can be found in tactical command and control situations.

While the approach is believed to be generally applicable to members of above class, it is limited to those organizations for which (1) tractable analytic models exist and (2) related descriptive data exists. These two conditions are currently very restrictive.

The paper is organized as follows. The next section discusses a three-phase approach to the design of organizations. In the third section, a specific task is presented for which an organization is desired. The design approach is then used to develop an organization to accomplish the task. Section four presents results of tests of the design, and a fifth section summarizes and concludes the paper.

II. DESIGN METHOD

The approach to organization design used in this paper focuses on where and how in the design process to include consideration of human characteristics and limitations. With this in mind, an approach with three distinct parts, or phases, is pursued. Figure 1 shows the relationship of these phases. Given a (possibly general) statement of the task for which an organization

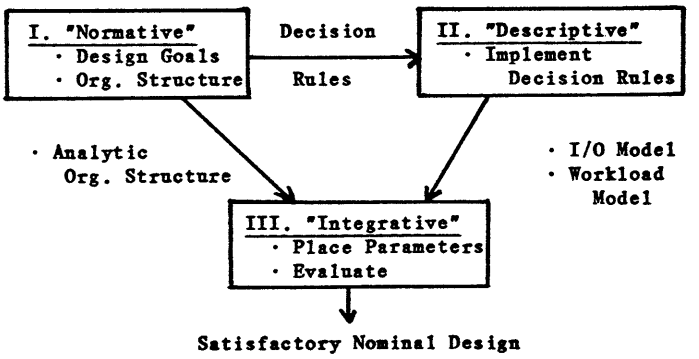


Figure 1 Organization Design Approach

is desired, the first phase in the process establishes a basic organization structure, which is expressed in analytic terms. The initial step in doing this includes the specification of the number of members, their interconnection, and their protocols for communication. It also includes the expression of design goals in terms of an objective function, as well as the delineation of the possible inputs and outputs to each organization member. In other words, everything about the organization structure is specified in analytical terms, except the mapping from inputs to outputs to be made by each member. Phase I is completed by solving an optimization problem that determines what these mappings should be. The resulting decision rules represent the desired behavior of organization members. As such they are job descriptions that are to be realized as closely as possible by actual human behavior in the organization. Execution of Phase I is thus normative in nature. In this context, and also in view of the class of organizations under consideration, models and results from the mathematical investigation of teams [1] are complementary to the issues that are addressed in Phase I of the organization design process.

Having determined, in the form of a decision rule, the information processing that each member is to perform, a second phase of the design begins in which decision rules are implemented. "Implementation" includes the specification of a collection of physical equipment, such as displays and response mechanisms, that the human is to use in order to accomplish the processing required by the decision rule. Also included is the specification of how the human is to use this equipment to perform his assigned task. Given the physical set-up and the directions for using it, a model is then developed that describes the organization member's behavior as the task is executed. This model has two components. The first is a description of the actual input/output behavior realized. The second is a measure of the workload induced by task execution. Both descriptions will in general depend on settings of parameters that are part of the physical task set-up, and also on variables that relate to how the organization member chooses to perform his task. Furthermore, since human information processing is subject to limitations, there will in general exist a maximum value of workload against which to compare the workload induced by the task. Thus Phase II of the design process is one that involves human modeling, but is such that a focus exists, in the form of a job description, for the tasks that are to be implemented. Techniques and models from human factors analysis, man-machine

†This research was supported by the Office of Naval Research under grants ONR/N00014-77-C-0532 (NR 041-519) and ONR/N00014-84-K-0519 (NR 649-003).

systems investigations, and cognitive psychology can be brought to bear to accomplish Phase II of the design.

The first two phases of design result in related, but distinct, design elements. On the one hand is an analytic organization structure that has been developed assuming ideal human behavior. On the other is a set of decision rule implementations that have been constructed so that actual human behavior can match, as closely as possible, that which is desired. The match is not necessarily perfect, however, particularly given human errors and workload limitations. Thus a third phase is necessary to integrate design elements in order to complete the organization design. In this phase the descriptions of actual input/output behavior are substituted for the decision rules in the analytic organization structure and the structure itself is augmented with the workload models. Then a constrained optimization problem is formulated to place parameters of task set-ups and parameters that relate to information processing choices available to members. The problem seeks to optimize organization performance, but does so in view of workload-related limitations of individual members. The solution to this problem is a nominal organization design that can be evaluated with respect to design goals.

Operation of the organization as designed requires that parameters of the physical task set-ups be set to the values obtained from solution of the constrained optimization problem. In addition, values obtained for information processing parameters can be interpreted either as prescriptions for how a member should be trained to exercise his information processing options, or as predictions for how he will. Successful completion of Phase III terminates the design process, although it may require several iterations on previous design steps before a satisfactory nominal design is obtained. The next section illustrates the design approach by applying it to a specific problem.

III. DESIGN EXAMPLE

Problem Statement

Suppose that the situation illustrated in Figure 2

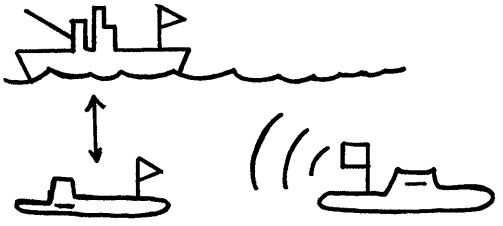


Figure 2 Illustration of Design Problem

is presented as a design problem. Two platforms, one surface and one submerged, are to perform, in a coordinated manner, a detection task regarding the presence or absence of another submerged target. Observations distinct to each platform are available each τ_0 time units, and it is required that repeated detection decisions be made at this rate, with a maximum delay of τ_d time units between a pair of observations and the detection decision associated with that pair. Furthermore, there is to be limited communication between platforms. It is desired to minimize the probability of error in detection, but in any case to make it less than the fraction $\bar{J}_0$. For this set of conditions, an organization is to be designed. To do this, the approach discussed in the previous section will be used.

Phase I

To begin the design process, a basic organization structure must be specified. In the present situation, it is natural to assume a two-member organization. Moreover, the task itself is one that falls within the class of distributed detection networks [2]. Thus a tandem structure is assumed for the organization, as shown in Figure 3. The presence or absence of a target

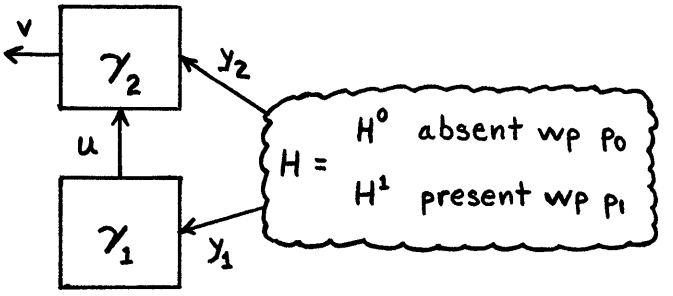


Figure 3 Organization Structure

is modeled as two hypotheses H, where $H \in \{H^0, H^1\}$. $H = H^k$ with a priori likelihood p_k. Observations by each platform are assumed to be conditionally gaussian, with $p(y_i | H=H^k) \sim N(m_{ik}, \sigma_i^2)$ $(i = 1,2;\ k = 0,1)$. Furthermore, observations are presumed to be related to incoming signal energy, which implies that $m_{i0} < m_{i1}$. For the particular situation under consideration, $p_0 = 0.4$ and $(m_{11} - m_{10})/\sigma_1 = 2$ and $(m_{21} - m_{20})/\sigma_2 = 2.6$, arbitrarily. Based on the observation received by the submerged platform (y_1), a value of u is selected and communicated to the surface platform. To incorporate the limited communication condition into the organization structure, u is restricted to two values: $u \in \{0,1\}$. Thus the first member provides only an indication regarding the target's presence or absence. The surface platform uses this indication together with the observation y_2 to decide a value of v, where $v \in \{0,1\}$. The latter is the overall detection decision of the organization. This process is to be repeated over and over as each new set of observations $\{y_1, y_2\}$ arrives every τ_0 time units.

In the structure described above, everything has been specified in analytic terms except how values of u should be determined from observations y_1 and how values v should be determined from y_2 and u. These unspecified elements are the decision rules for each organization member. To determine what they should be, an optimization problem is formulated to find the set of decision rules $\{\gamma_i^*\}$ that minimizes organization detection error. γ_1^* and γ_2^* are known be threshold tests:

$$\gamma_1^*: \begin{cases} \text{if } y_1 \geq t_1^* & u = 1 \\ \text{else} & u = 0 \end{cases}$$

$$\gamma_2^*: \text{ if } u = j \text{ and } \begin{cases} y_2 \geq t_{2j}^* & v = 1 \\ y_2 < t_{2j}^* & v = 0 \end{cases} \qquad j = 0,1$$

The values of t_1^*, t_{20}^*, and t_{21}^* depend on the relative quality of each member's observations and on the a priori likelihood of H. Basically, the first member selects the second member's threshold and in so doing biases the decision of the second member. Furthermore, it happens that $t_{20}^* > t_{21}^*$, so that the direction of the basis is consistent with the first member's indication.

Phase I of the design process is now complete. An analytic organization structure exists that reflects the conditions of the problem. Decision rules for each member have been determined that represent the ideal behavior of organization members. Attention is now focused on implementing these decision rules so that the

human organization members can attempt to realize their desired behavior.

Phase II

Implementation of the decision rules γ_i^* requires the specification of how each member's observations are to be presented so that the proper threshold comparison test can be made. It is also necessary to provide a mechanism for recording each member's response to a particular observation. Furthermore, for each of the two physical task set-ups, a description of human behavior at that task is needed. This includes a model of the member's performance at making threshold comparison tests. It also includes a model for the workload of the task. In this design situation, processing time will be used to derive a measure of workload.

Given the overall limits on processing time for the organization (τ_o and τ_d), implementation of the decision rules will begin by allocating this time between organization members. The first member will be required to process observations at the same rate that they arrive. That is, on the average, he must make a threshold comparison test every τ_o time units, where

$$\tau_1 = \tau_o \qquad (1)$$

This leaves $\tau_d - \tau_o$ time units for passing the message u between members and for the second member to respond. Communication between members is assumed to take negligible time. Thus the second member is allocated τ_2 time units, where

$$\tau_2 = \tau_d - \tau_o \qquad (2)$$

By contrast with the first member, however, τ_2 will be regarded as a deadline. The notion is that each decision by the second member will be constrained to take no more than τ_2 time units. In practice, τ_2 will be interpreted as the maximum average response time, assuming a narrow distribution of response time values.

This allocation of time is a design choice, and other choices are made later. In practical applications, good engineering practices will dominate these choices. In this research context, choices have been made to illuminate interesting aspects of organization behavior, often at the expense of pragmatic considerations.

Consider now the task of the first member. He is to compare an observation y_1 with a threshold t_1 and then decide a value of u. One way for him to do this is shown in the upper part of Figure 4. Observations y_1 are presented visually in the form of a horizontal "crossbar pattern", where the midpoint of the pattern is the value of y_1 observed. The member then decides whether the pattern midpoint is left or right of the vertical threshold and responds by depressing one of two mechanical buttons. The vertical threshold is positioned according to the value of t_1. In the lower part of Figure 4, the distribution on observations y_1, i.e. pattern midpoints, is shown (solid) as the weighted sum of two conditional distributions $p(y_1|H^k)$.

The first member is constrained to process patterns at a rate of τ_1 time units per pattern. The dial at the top of the display indicates to the member how many patterns are waiting to be processed. It advances clockwise as patterns join the queue. Since the patterns are to be processed at the same rate they arrive, the member must maintain the dial position at or near vertical in order to meet his processing rate constraint. Whether or not this is possible depends on the average time required to process patterns. The time to view and respond to a pattern, i.e. to make a

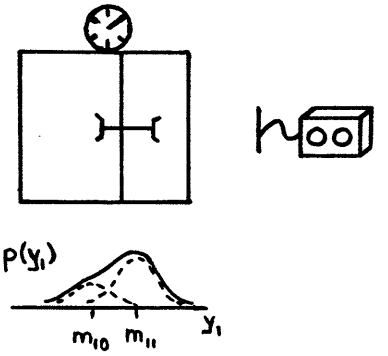


Figure 4 First Member's Task

"stimulus-controlled response (SCR)", varies with the position of the threshold t_1, however. Figure 5 shows

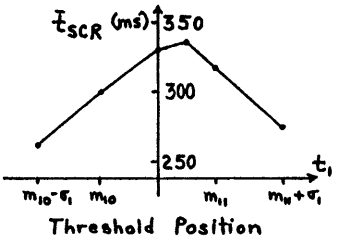


Figure 5 Average Response Time For First Member

experimentally observed variation of average SCR time with respect to the threshold position for one subject; the results are representative of those obtained from other subjects as well.

Given enough time, i.e. if $\bar{t}_{SCR} < \tau_1$, the subject is able to decide left or right of the threshold with near perfect accuracy. For example, at $t_1 = t_1^*$, $\bar{t}_{SCR}$ is approximately 330 ms. If $\tau_1 > 330$, then the member will virtually realize the ideal behavior determined by γ_1^*. However, as τ_1 decreases below $\bar{t}_{SCR}$, the member is observed to make errors as he tries to maintain the required rate.

Rather than incur SCR errors, an alternative processing option will be provided to the member: the option to "fast guess (FG)". Fast guessing means that the member ignores the pattern presented and responds arbitrarily. This takes about t_{FG} = 180 ms, which is considerably less time than a stimulus-controlled response. Thus the member can presumably fast guess enough times to meet the rate constraint, and can carefully process patterns the remainder of the time. It is assumed that a 50/50 bias is used by the member when fast guessing. This is enforced experimentally by having the subject depress both buttons when choosing to fast guess. These responses are then assigned a 0 or 1 value with equal likelihood before being passed to the second member. The investigation of how bias in guessing affects the organization's operation is of interest. A companion paper in this volume [3] discusses such effects.

Thus the model of the first member's behavior at his task is as follows. Let $\bar{k}$ designate a conditional distribution on outputs, given a particular input. The overall input/output conditional distribution for the first member, $\bar{k}_1$, is determined as a combination of the conditional distributions corresponding to the individual options:

$$\bar{k}_1 = (1-q_1)\,\bar{k}_{SCR} + q_1 \cdot \bar{k}_{FG} \qquad (3)$$

In eq.(3), q_1 is the fraction of fast guessing.

Similarly, the overall average response time T_{p1} is a combination of the individual option response times:

$$T_{p1} = (1-q_1)\cdot\bar{t}_{SCR} + q_1\cdot\bar{t}_{FG} \quad (4)$$

The model given by eq.(3)-(4) is essentially the Fast Guess model of Yellot [4], which is one of the mechanisms by which humans can trade speed for accuracy. Note that this model has two parameters: the threshold position t_1, and the fraction of fast guessing q_1. Determination of values of these parameters is made at a later stage in the design process and is done with respect to overall organization performance.

For the second organization member, an implementation for the decision rule γ_2^* is chosen as shown in Figure 6. Depending on the signal from the

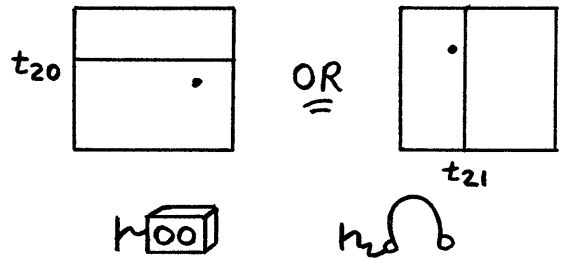


Figure 6 Second Member's Task

first member, threshold t_{20} or t_{21} is selected to be used by the second member. If it is the former, t_{20} is displayed as a horizontal line and the observation y_2 is displayed as a vertical displacement. If t_{21} is selected, the threshold is displayed vertically and y_2 is a horizontal displacement. Two horizontally arranged mechanical buttons are used to record responses. The left button is used if y_2 is left (t_{21}) or down (t_{20}) and the right button is used in the complementary situations. Recall that the second member is viewed as subject to deadline on each response. An auditory mechanism has been used to indicate that the deadline has passed, which is represented by the headphones in Figure 6.

As with the first member's implementation, if the second member has enough time, he can perform his task flawlessly. The (average) processing time required for the task, denoted T_{p2}, depends on the amount of threshold switching. Denote by q_0 the quantity p(u=0). Figure 7 shows one subject's observed processing time

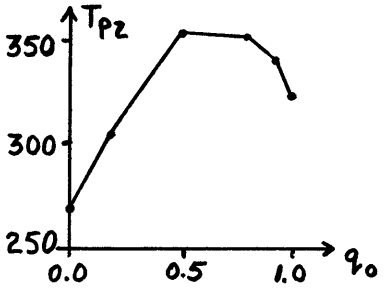


Figure 7 Second Member Processing Time

versus q_0, which is the fraction of threshold t_{20}'s use. A considerable overhead for switching is evident, as well as a difference in processing time for horizontally and vertically displayed thresholds. So long as τ_2 is greater than the time required, however, actual input-/output behavior can be expected to match desired behavior.

If the time required (T_{p2}) is less than the time allowed (t_d), errors will be made as the member is forced to trade accuracy for speed. One representation of this tradeoff is due to Pew [5], who suggests a log-linear relationship between the odds ratio (# right divided by # wrong) and response time. Using this representation, Figure 8 gives the speed/accuracy

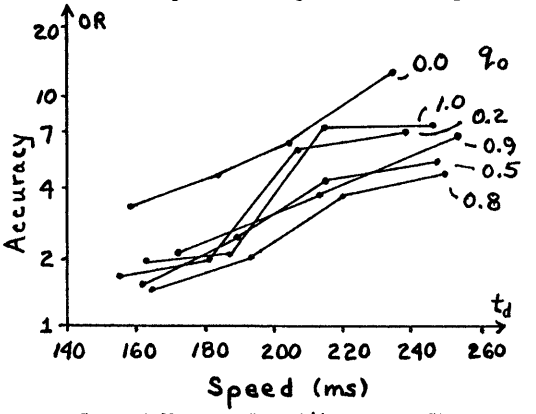


Figure 8 Second Member Speed/Accuracy Characteristic

characteristics for the second member's task as evidenced by one individual.

To obtain the results shown in the figure, several hundred responses were recorded at each q_0 condition using various deadlines. The data at each q_0 level were then rank-ordered by response time and partitioned into groups of a few hundred responses each. For each group, the average time and the odds ratio were computed. These values are the coordinates plotted in Figure 8 as representative speed/accuracy operating points.

The deadlines used were chosen such that insufficient time was available to do the task with highest accuracy. For operation in this region, it is evident that as t_d decreases there is a general decline in accuracy. Moreover, for given t_d, it is apparent that as q_0 increases from 0 up to near 0.8, accuracy decreases. This a direct result of the processing time requirements of the task as given in Figure 7. As q_0 increases still further toward 1.0, T_{p2} decreases and accuracy improves.

A model for the second member is abstracted from the data in Figure 8 as follows. Denote by f the logarithm of the odds ratio. Then a linear approximation for each speed/accuracy locus can be written in terms of f:

$$f = f_s(q_0)\cdot(t_d - t_c(q_0)) \quad (5)$$

where f_s and t_c are quantities that are chosen to best represent observed behavior. Table 1 gives the values

Table 1 Second Member Model Parameters

q_0	f_s	t_c
0.0	0.0217	117
0.2	0.0190	130
0.5	0.0152	132
0.8	0.0142	137
0.9	0.0143	119
1.0	0.0209	136

estimated from the data in Figure 8. To express the behavior represented by f in a form consistent with γ_2^*, define q_2 to be the input/output error rate. Then

$$q_2 = (1 + e^f)^{-1} \quad (6)$$

and the input/output behavior of the second member can

summarized by the conditional distribution $\bar{k}_2$:

$$\bar{k}_2: \quad \text{if } u = j \text{ and } \begin{cases} y_2 \geq t_{2j} \begin{cases} v = 1 \text{ wp } 1-q_2 \\ v = 0 \text{ wp } q_2 \end{cases} \\ y_2 < t_{2j} \begin{cases} v = 1 \text{ wp } q_2 \\ v = 0 \text{ wp } 1-q_2 \end{cases} \end{cases} \qquad j = 0,1 \qquad (7)$$

Eq.(7) implicitly assumes that the member exhibits no bias in his errors and also that changes in t_{2j} do not affect q_2. For the range of operating regions considered here, both of these assumptions are valid to first order. Note that even though t_{2j} values do not enter directly into accuracy considerations, it is not necessarily desirable to leave them at their respective decision rule values, since the effects of switching may make it advantageous to adjust t_{20} and t_{21} to relieve both members' workload, to the benefit of the overall organization.

Phase III - Integration

The elements of the organization design, which are its basic structure and the decision rule implementations, have now been established. To integrate the elements, the models of actual input/output behavior realized by organization members are substituted into the analytic organization structure for the decision rules and the structure itself is augmented with constraints derived from workload considerations. In the structure that results, there are five variables that have not been specified: the thresholds t_1, t_{20}, t_{21}; the guessing fraction q_1; and the actual deadline assigned to the second member t_d. To complete the process of integration and thereby obtain a nominal organization design, these variables are to be placed in order to optimize organization performance, but also in view of workload limitations of individual members. Specifically, the problem to be solved is stated as

Nominal Design - Constrained Organization (CNO)

min (organization detection error)

over: thresholds t_1, t_{20}, t_{21}
fast guess fraction q_1
second member deadline t_d

subject to: $T_{p1} \leq \tau_1$

$t_d \leq \tau_2$

Suppose that $\tau_0 = 260$ ms and $\tau_d = 500$. Then $\tau_1 = 260$ ms and $\tau_2 = 240$ ms. These constraints are shown in Figure 9 on the respective models of organization members. In addition, a key linkage of the members is shown, which is the amount of switching that the first member's operation imposes on the second member. For the first member, the rate constraint is such that some fast guessing will be required, except where t_1 is placed at its smallest possible value. The value of q_1 is proportional to the distance between the average SCR time and the $\tau_1 = 260$ constraint. As an example, if $t_1 = t_1^*$, $q_1 \approx 0.45$. Placement of t_1 not only establishes q_1, but it also determines the distribution on u and consequently the frequency of threshold switching by the second member. At its minimum and maximum values, t_1 induces 5% and 95% use, respectively, of the horizontally displayed threshold t_{20}. Thus operation on a particular speed/accuracy locus by the second member is dictated according to the value of t_1.

In solving Problem CNO for the present design situation, it is straightforward to show that $t_d = \tau_2 = 240$ ms, i.e. the second member should always use the entire amount of time allowed to him [3]. This means

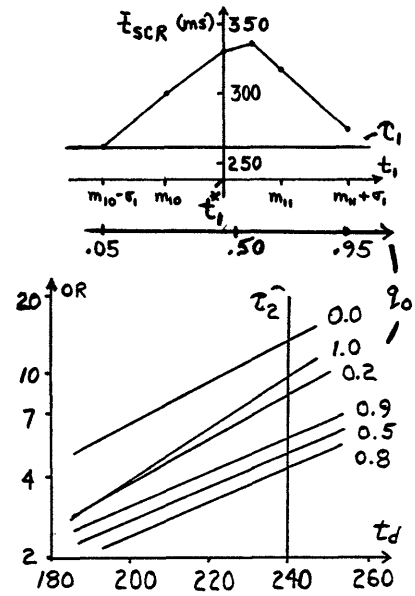


Figure 9 Illustration of Problem CNO

that the operating point of the second member will be somewhere on the vertical $\tau_2 = 240$ ms line in the speed/accuracy model. In completing the solution, a basic tradeoff must be made. At one extreme is the option to retain the first member's ideal threshold t_1^*, which gives "high quality" indications, but cannot be used all the time. If this is done, the second member uses his thresholds with nearly equal frequency. This places operation at a reduced level of input/output accuracy. At the other extreme is the option to place t_1 at its minimum value so that no fast guessing is required, but also so that all SCR responses are of lower quality. This in turn places the second member at a higher input/output accuracy operating level. Of course, there exist many other solution possibilities that are compromises between the two extremes.

The solution to Problem CNO for the values assumed in this situation, places t_1 at its minimum value in order to improve the input/output accuracy of the second member. Though not indicated explicitly in Figure 9, placing t_1 away from t_1^* also means that $t_{2j} \neq t_{2j}^*$ in the solution to Problem CNO, since overall organization performance can benefit by adjusting the second member's thresholds to compensate in part for the loss of processing quality by the first member. One interpretation for the solution outcome is based on the fact that the second member has the "last word" on the detection decision. For the benefit of the organization, it is better to have this decision be associated to the second member's observation as much as possible rather than be directly opposed to it. It is worth compromising the quality of the first member's indication to do this. Such a tradeoff is not always the outcome of Problem CNO. [3] investigates the solution characteristics of an idealized version of the problem considered here, and documents that a number of qualitatively different solutions are possible.

A final consideration in Phase III is whether the nominal design obtained satisfies original design goals. For the current design problem, one criterion for evaluation is whether the detection error probability realized, J_o, is less than that which was specified, $\bar{J}_o$. Another criterion might be whether the performance level would be maintained if the a priori likelihood of H were to change during organization operation. It is not apparent to either member what the underlying likelihood of H is. Thus little adjustment can be expected from organization members should p(H) change. It may

therefore be desirable to take this into account when specifying the organization design. Assuming that the present design is satisfactory with respect to such evaluation criteria, the design process terminates.

IV. TEST OF ORGANIZATION OPERATION

There are several characteristics of the design obtained in the previous section that suggest hypotheses about organization operation. First, if t_1 is set to its minimum value as per the design, there should be little fast guessing observed as the first member executes his task. Furthermore, there is a predicted percentage of switching that is part of the design and consequently a predicted level of input/output processing accuracy by the second member. Both of these hypotheses represent operation of organization members in regions that were previously examined when the descriptive models of their behavior were developed. Thus the predictions made are really tests of the validity of individual models.

A more interesting hypothesis about organization operation is the level of performance that will be realized. This is because the overall detection error of the organization cannot be inferred by individual members, but rather is a quantity that characterizes the organization. Furthermore, the design approach uses organization performance as the criterion for placing individual member parameters, and in effect discriminates in favor of one design solution over another based on predicted performance levels. Thus the extent to which actual performance of the organization matches that predicted represents a key test for the viablity of the design approach.

For the organization under consideration, ideal behavior, which is determined in Phase I, yields a detection error probability of 0.06. Suppose now that the normative thresholds are left in place and that the organization is operated as it has been implemented. That is, $t_1 = t_1^*$ and $t_{2j} = t_{2j}^*$. However, because individual members are constrained fast guessing is required by the first member and a lower accuracy is induced in the second member. The predicted organization performance for this operating condition, designated as condition "B", is 0.21. The solution to Problem CNO, however, predicts a performance level of 0.15. This condition is designated as "A".

Given these predictions about performance, the organization was operated at the nominal design point (A) and also with the thresholds set at their ideal values (B). Two tests were made at each operating point; observed performances for the organization are shown in Table 2, along with the predicted values. A reasonable agreement is apparent.

Table 2 Performance of Organization

	Ideal (Phase I)	Nominal Design (A)	Normative Thresholds/ Constrained Members (B)
Predicted	0.06	0.15	0.21
Test 1	-	0.15	0.20
Test 2	-	0.14	0.20

The levels of fast guessing and input/output accuracy predicted for individual members were also observed in the actual operation of the organization. In particular, no fast guessing was required in Condition A, and about 47% of the responses were fast guesses when operating under condition B. Input/output accuracy for Condition A was observed to be such that q_2 was about 0.06. For Condition B q_2 was near 0.16.

These results suggest two conclusions. First, failure to take human limitations into account can result in performance that is substantially different from that which assumes ideal human behavior. Second, there is considerable advantage to adjusting organization parameters in the situation where members are subject to workoad limitations. Finally, the experimental results and conlcusions presented here do not represent isolated behavior. [6] contains similar results using different individuals as organization members within the same basic organization structure.

V. SUMMARY

This paper has suggested an approach for the design of human information processing organizations for the situation where organization members perform routine tasks under the pressure of time. A main focus has been how and at what point in the design process to include consideration of human characteristics and limitations. A three-phase approach has been given for structuring the problem so that a balance is obtained between the complexities of considering how human behavior impacts every aspect of the organization and the hazards of neglecting consideration of human limitations in order to simplify the problem.

One of the advantages of the approach is that separation into normative and descriptive phases simplifies the design problem without greatly limiting design options. By deriving job descriptions for individual members in Phase I, a focus is provided for the execution of Phase II. A second advantage of the approach is that tradeoffs between member workload and organization performance are made apparent in the integration phase.

The design approach has been illustrated concretely by executing it on a specific design problem. The resulting organization design has been tested and found to operate as predicted. This demonstration is particularly supportive of the integrative design phase, since this phase represents a novel feature of the design approach. That there is agreement between observed and predicted operating characteristics is evidence that the approach is a valid one for organization design.

References

[1] Y-C. Ho, "Team decision theory and information structures," Proceeedings of the IEEE, Vol. 68, no. 6, pp. 644-654.

[2] L.K. Ekchian and R.R. Tenney, "Distributed detection networks," Proc of the 21st IEEE Conf on Dec and Cont, Ft. Lauderdale, FL, 1982, pp. 686-691.

[3] K.L. Boettcher and R.R. Tenney, "Distributed decisionmaking with constrained decisionmakers," Proceedings of the 8th MIT/ONR Workshop on C³ Systems, Cambridge, MA, 1985.

[4] J.I. Yellot, "Correction for Fast Guessing and the Speed-Accuracy Tradeoff in Choice Reaction Time," Journal of Math Psych, Vol.8, 1971, pp. 159-199.

[5] R.W. Pew, "The Speed-Accuracy Tradeoff," Acta Psychologica, Vol. 30, 1969, pp. 16-26.

[6] K.L. Boettcher, "A Methodology for the Analysis and Design of Human Information Processing Organizations," PhD Thesis, Department of Electrical Engineering, MIT, Cambridge, MA, 1985.

DISTRIBUTED DECISIONMAKING WITH CONSTRAINED DECISION MAKERS - A CASE STUDY†

Kevin L. Boettcher and Robert R. Tenney
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

A specific distributed decisionmaking problem is formulated that includes processing load constraints on team members. Solutions are possible that place team members in regions where random behavior is required and/or where individual errors are likely.

I. INTRODUCTION

A main goal in most distributed decisionmaking formulations, particularly team theoretic ones, is to obtain normative decision rules that represent the desired behavior of each decision agent or team member [1]. This paper considers a modified team theoretic problem that incorporates decision rules that are descriptive of actual human behavior, and furthermore takes into account the processing load incurred to execute these decision rules. The problem formulated is motivated by considerations in the design of human information processing organizations of the type where organization members perform routine processing tasks under the pressure of time. Examples of such organizations are found in air traffic control and command and control situations. In this context, the usual team problem can be taken as a model of organization structure, and decision rules as idealized behavior for organization members. When models for actual human behavior are substituted for the normative decision rules in the team structure, team behavior in general changes. Furthermore, the workload of team members may be such that desired team operation exceeds human processing limitations. Thus, given the basic team structure, a problem can be formulated to choose decision rules, to be realized by actual human behavior for best team performance, subject to their feasibility with respect to team member processing load.

The specific team structure considered is that of a two-member, tandem distributed detection network. Section II describes this structure and reviews the characteristics of theoretical team member behavior. A key feature of the decision rules is the presence of thresholds, which each member uses to make comparison tests. A model for the information processing required to execute such a test is then described, with processing time used as the measure of workload. The complete model for each member's actual behavior includes a second element, however, which accounts for behavior when processing time for threshold tests exceeds the time allowed. This element derives from human ability to trade accuracy for speed. Two different mechanisms for doing this are incorporated, one for each member. The overall actual behavior and processing load realized is parameterized by the thresholds used and other parameters that figure in the speed/accuracy tradeoff capability. The modified team theoretic problem is then to place these parameters for minimum team error, subject to processing time used being less than processing time available for each member. Section III discusses the characteristics of the problem solution. A particular consideration of interest is whether, and if so under what conditions, it remains desirable to retain the thresholds obtained in the original (unconstrained) team problem. Section IV investigates a special case of the problem, from which principles of general interest are apparent. Finally, Section V summarizes the paper.

†This research was supported by the Office of Naval Research under grants ONR/N00014-77-C-0532 (NR 041-519) and ONR/N00014-84-K-0519 (NR 649-003).

II. PROBLEM FORMULATION

A. Team Structure

Consider the two member, tandem, distributed detection network shown in Figure 1. Each team member receives a conditionally independent, gaussian observation on the presence or absence of a given phenomenon H. Based on his observation, the first member selects one of two symbols to send to the second member. The latter then incorporates his own measurement with the received symbol to make a detection decision for the network. The optimal decision rules γ_j^* for each team member that minimize the probability of error in detection are known [2]. They are threshold tests as given in (1).

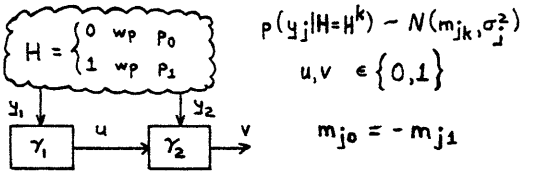


Figure 1 Team Structure

$$\gamma_1^*: \quad \text{if } y_1 \gtrless t_1^* \quad u = 1 \text{ or } 0$$

$$\gamma_2^*: \quad \text{if } u = i \ (i = 0,1) \text{ and if } y_2 \gtrless t_{2i}^* \quad v = 1 \text{ or } 0 \qquad (1)$$

Basically, the first member biases the second member's choice by selecting the latter's threshold.

B. Information Processing Models

Now consider that the threshold comparison tests in (1) are to be accomplished by humans. For example, the observation could be displayed visually as a horizontally displaced dot, with the threshold also displayed as a vertical line displaced according to its value. Viewing such a display and selecting a response takes time. Furthermore, threshold position with respect to the likely position of observations will have an effect on the time required to select a response. In particular, assume that a comparison with threshold t requires, on the average, $\bar{t}_p$ seconds to make, where

$$\bar{t}_p = \bar{t}_p(t) = a - b\cdot(t)^2 \quad a > 0,\ b \geq 0 \qquad (2)$$

Given that observations are predominantly near zero, the model in (2) reflects the observed behavior that response time decreases as the uncertainty decreases in the response required. In eq.(2), as t becomes large in absolute value ($b \neq 0$), the likelihood that observations will fall only on one side of t is high.

First Team Member

The first team member performs his task using a single threshold. The processing time required to do this test is given by eq.(2); specifically, it is $\bar{t}_{p11}(t_1) = a_1 - b_1\cdot(t_1)^2$. In addition, it is assumed that the input/output behavior realized is such that a flawless comparison can be made. Denote by $\bar{k}_{11}$ the conditional distribution $p(u|y_1)$ realized using the threshold test. The model is then that of

$$\bar{k}_{11}: \quad \text{if } y_1 \geq t_1 \quad u = 1 \ ; \quad \text{else } u = 0 \qquad (3)$$

Suppose now that the operation of the team is such that the member must complete comparison tests at the rate of

one every τ_1 seconds. If it happens that t_1 is set such that $\bar{t}_{p11}(t_1) > \tau_1$, the member will be overloaded. Therefore, an alternative processing mode is provided: an option to "guess", i.e. to essentially ignore the observation y_1 and to arbitrarily respond u = 1 with some guessing bias g_1. Input/output behavior when guessing is modeled by the conditional distribution $\bar{k}_{12}$:

$$\bar{k}_{12} : \quad u = 0 \text{ wp } 1 - g_1 \quad \text{and} \quad u = 1 \text{ wp } g_1 \tag{4}$$

To make this a viable option, assume that the time required to exercise this option, denoted by $\bar{t}_{p12}$, is less than $\bar{t}_{p11}(t_1)$ for some range of t_1 values.

Finally, because the team member has two options, there will be an additional amount of processing time required to switch between them. Switching overhead depends on switching frequency, as given by the expression

$$d_1 \cdot (1 - q_1) \cdot q_1 \tag{5}$$

where q_1 is the fraction of guessing and d_1 is a scale factor. If one option is used exclusively, (5) is zero. Thus, the first team member has an input/output behavior modeled by K_1 that requires a processing time of T_{p1}:

$$K_1 = (1 - q_1) \cdot \bar{k}_{11} + q_1 \cdot \bar{k}_{12} \tag{6}$$

$$T_{p1} = (1 - q_1) \cdot \bar{t}_{p11}(t_1) + q_1 \cdot \bar{t}_{p12} + d_1 \cdot (1-q_1) \cdot q_1 \tag{7}$$

The model given in eq.(6) and eq.(7) is basically the so-called Fast-Guess model [3], which reflects one mechanism whereby humans can trade speed for accuracy.

Second Team Member

The second team member switches between two thresholds. Assuming an overhead for switching similar to (5), the average time required to accomplish this task depends on the threshold values, and the relative frequency of using them:

$$T_{p2} = \sum_{i=0}^{1} \left[p(u = i) \cdot (a_{2i} - b_{2i} \cdot (t_{2i})^2) \right] + d_2 \cdot p(u = 0) \cdot p(u = 1) \tag{8}$$

As with the first team member, it assumed that the second member is subject to a processing time limit; in this case it is assumed to be a like a deadline constraint τ_2. So long as $T_{p2} \leq \tau_2$, the team member can accomplish this processing without error. Errors will be made, if however, if p(u), t_{20}, and t_{21} are such that $T_{p2} > \tau_2$. The likelihood of errors depends on the difference between the deadline imposed, denoted t_d, and the processing time required T_{p2}. Thus the input/output behavior of the second member, K_2, is as follows:

K_2: if u = i (i = 0,1) and (9)

$y_2 \geq t_{2i}$, then v = 1 wp $1-q_2$; v = 0 wp q_2

$y_2 < t_{2i}$, then v = 0 wp $1-q_2$; v = 1 wp q_2

where

$$q_2 = q_2(T_{p2}, t_d) = (1 + e^{f(T_{p2}, t_d)})^{-1} \tag{10}$$

and

$$f = \begin{cases} f_s \cdot (t_d - T_{p2}) + f_m & T_{p2} \geq t_d \\ f_m & T_{p2} < t_d \end{cases} \tag{11}$$

In words, the second member performs the threshold comparison test correctly a fraction $(1 - q_2)$ of the time, and makes an error on the fraction q_2 of the observations processed. For analytical convenience it is assumed that $f_m < \infty$, which effectively means that the minimum value of q_2 is non-zero. Eqs.(8)-(11) form the model of the second team member. It reflects a second mechanism of trading speed for accuracy exhibited by humans. In particular, the log-linear relationship between the "odds-ratio" $(1-q_2)/q_2$ and processing time is derived from Pew [4]. Note that values of f_s and f_m are not selected; they represent fixed human behavior.

C. Problem Statement

Five independent variables have been specified within the models of team members. They include the three comparison thresholds (t_1, t_{20}, t_{21}), the amount of guessing by the first member (q_1), and the processing time deadline for the second member (t_d). Substituting K_j for γ_j^* and adding the processing time constraints for each member, a constrained optimization problem can be formulated to minimize the detection error probability for the organization, subject to meeting the processing time limitations of each member. Denote by J_o the detection error probability. Then formally stated, the problem is as follows.

Problem A1 (Constrained Optimization Problem)

$$\min_{t_1, t_{20}, t_{21}, q_1, t_d} \quad J_o(q_1, t_1, t_{20}, t_{21}, t_d)$$

$$\text{s.t.} \quad T_{p1} \leq \tau_1 \quad ; \quad t_d \leq \tau_2$$

III. SOLUTION CHARACTERISTICS

There are several issues of interest with respect to the solution of Problem A1. One is whether it is ever to any advantage to set the deadline t_d for the second member to be strictly less than τ_2. This is shown not to be the case, due to the monotonicity of q_2 in t_d. A second issue is whether a possible solution is to leave the thresholds at their unconstrained optimal values, i.e. $t_1^*, t_{20}^*, t_{21}^*$, and to tolerate any consequent input/output errors (q_2) or guessing (q_1). At the other extreme is the possible solution of adjusting thresholds such that q_1 and q_2 are minimized. The basic consideration is one of whether it is better to absorb guesses and input/output errors some of the time in order to use quality thresholds most of the time, or to use an "inferior" set of thresholds all of the time. In Problem A1, so long as the thresholds t_{20}, t_{21} affect processing time of the second member, it is better to adjust them. However, solutions to Problem A1 do not necessarily minimize q_2 and q_1.

Examination of Problem A1 is greatly facilitated by taking advantage of the fact that the joint distribution p(u ,H) completely characterizes the analytical link between team members [2]. Thus the minimization in Problem A1 can proceed in two stages. First t_{20}, t_{21} and t_d can be placed as a function of p(u,H). Since there is a 1-1 relationship between (q_1, t_1) pairs and p(u,H) distributions, a second minimization can be performed over these distributions to place q_1 and t_1, and thereby solve Problem A1. Denote by p_{ik} the quantity $p(u=i, H=H^k)$. Then it is convenient to represent the distribution p(u,H) as a vector $\mathbb{T}$, where

$$\mathbb{T} = [p_{00}, \; p_{10}, \; p_{01}, \; p_{11}]' \tag{12}$$

Furthermore, possible $\mathbb{T}$ values depend on t_1 and q_1 according to

$$\mathbb{T} = (1-q_1) \cdot [p_{00t}(t_1), \; p_0 - p_{00t}(t_1), \; p_1 - p_{11t}(t_1), \; p_{11}]' + q_1 \cdot [(1-g_1) \cdot p_0, \; g_1 \cdot p_0, \; (1-g_1) \cdot p_1, \; g_1 \cdot p_1]' \triangleq \mathbb{T}(t_1, q_1) \tag{13}$$

where

$$p_{00t}(t_1) = \Phi\left(\frac{t_1 - m_{10}}{\sigma_1}\right) \cdot p_0 \tag{14a}$$

$$p_{11t}(t_1) = \left[1 - \bar{\Phi}\left(\frac{t_1 - m_{11}}{\sigma_1}\right)\right] \cdot p_1 \qquad (14b)$$

and $\bar{\Phi}(\cdot)$ is the unit normal cumulative distribution function. From eq.(13) it is evident that $\mathbf{T}$ is determined as a combination of two $\mathbf{T}$ vectors, one corresponding to exclusive use of the threshold and one corresponding to exclusive use of guessing.

Define

$$J(\mathbf{T}, t_{20}, t_{21}) = p_{00} \cdot \left[1 - \bar{\Phi}\left(\frac{t_{20} - m_{20}}{\sigma_2}\right)\right] + p_{01} \cdot \left[\bar{\Phi}\left(\frac{t_{20} - m_{21}}{\sigma^2}\right)\right] + p_{10} \cdot \left[1 - \bar{\Phi}\left(\frac{t_{21} - m_{20}}{\sigma_2}\right)\right] + p_{11} \cdot \left[\bar{\Phi}\left(\frac{t_{21} - m_{21}}{\sigma_2}\right)\right] \qquad (15)$$

Eq.(15) represents the detection error probability of the team as a function of $\mathbf{T}$, t_{20}, and t_{21}, assuming $q_2 = 0$. Rewriting J_0 using J and showing the decomposition by stages, Problem A1 becomes

$$\min_{t_1, q_1} \left[1 - 2 \cdot J(\mathbf{T}, t_{20}, t_{21})\right] \cdot q_2(T_{p2}, t_d) + J(\mathbf{T}, t_{20}, t_{21})$$

$$\text{s.t.} \quad T_{p1} \leq \tau_1, \quad \mathbf{T} = \mathbf{T}(t_1, q_1)$$

$$\min_{t_{20}, t_{21}, t_d} \quad \text{s.t.} \quad t_d \leq \tau_2$$

Finally, before proceeding to an analysis of solution characteristics, it is convenient to formulate a modified version of Problem A1. It is true that explicit dependence on thresholds t_{20} and t_{21} occurs in Problem A1 only in the function J and in the determination of processing time T_{p2}. Therefore it is possible to aggregate these thresholds into the single variable $\bar{T}_{p2}$ and to substitute a new function $\bar{J}$ for J, where

$$\bar{J}(\mathbf{T}, \bar{T}_{p2}) = \min_{t_{20}, t_{21}} J(\mathbf{T}, t_{20}, t_{21}) \quad \text{s.t.} \quad T_{p2} = \bar{T}_{p2} \qquad (16)$$

In other words, given a $\mathbf{T}$ and $\bar{T}_{p2}$ value, the relationship of t_{20} and t_{21} is defined (in fact they describe an ellipse). The minimization in eq.(16) generates threshold values $\bar{t}_{2i}$ that are the solutions to eq.(16) as a function of $\bar{T}_{p2}$. Using this aggregation, Problem A1 an be stated in terms of $q_1, t_1, \bar{T}_{p2}$ and t_d as:

Problem A2

$$\min_{q_1, t_1} \left[1 - 2 \cdot \bar{J}(\mathbf{T}, \bar{T}_{p2})\right] \cdot q_2(\bar{T}_{p2}, t_d) + \bar{J}(\mathbf{T}, \bar{T}_{p2})$$

$$\text{s.t.} \quad T_{p1} \leq \tau_1, \quad \mathbf{T} = \mathbf{T}(t_1, q_1)$$

$$\min_{t_d, \bar{T}_{p2}} \quad \text{s.t.} \quad t_d \leq \tau_2$$

Assigning Deadline

Consider now the inner minimization in Problem A2. For given $\bar{T}_{p2}$, necessary conditions for a solution value of t_d [5] are given by

$$\frac{\partial \bar{J}}{\partial t_d} \cdot \left[1 - 2 \cdot q_2\right] + \frac{\partial q_2}{\partial t_d} \cdot \left[1 - 2 \cdot \bar{J}\right] + \mu = 0 \qquad (17a)$$

$$\mu \cdot (t_d - \tau_2) = 0 \qquad (17b)$$

$$\mu \geq 0 \qquad (17c)$$

The first term in eq.(17a) is zero since $\bar{J}$ does not depend on t_d. The first factor in the second term is negative, since q_2 is monotonically decreasing with respect to increasing t_d. Furthermore, $\bar{J}$ is bounded above by 0.5. The latter derives from the interpretation of J as the detection error probability of the team when $q_2 = 0$. A value of $J \geq 0.5$ implies that the thresholds are being used to give observations an opposite interpretation, one which results in worse than chance behavior. Assuming that the minimization in eq.(16) assures that at least chance performance will obtain, i.e. $\bar{J} < \min(p_0, p_1) \leq 0.5$, then eq.(17) implies that $t_d = \tau_2$. That is, always place the deadline at the maximum allowable. This result is valid independent of $\bar{T}_{p2}$ and $\mathbf{T}$ values.

Using Unconstrained Optimal Thresholds

Continuing with examination of the inner minimization, consider the question of whether the unconstrained optimal thresholds can be a solution to Problem A2. Because of the reformulation in terms of $\mathbf{T}$ and the stagewise minimization structure, this question must be answered in a more general way. Whereas the minimization in eq.(16) resulted in the construction of two functions $\bar{t}_{2i}(\bar{T}_{p2}, \mathbf{T})$, performing the minimization of J without the constraint in eq.(16) results in two different functions that represent the unconstrained optimal values of t_{2i} for a given $\mathbf{T}$ value. Included in this set is the pair of thresholds that define of γ_2^*. Indeed, if the functions defined by the unconstrained minimization are denoted $t_{2i}^*(\mathbf{T})$, then

$$t_{2i}^* = t_{2i}^*(\mathbf{T}(t_1^*, 0)) \qquad (18)$$

The investigation below proceeds in terms of $\mathbf{T}$ and determines whether $t_{2i}^*(\mathbf{T})$ represent a possible solution to the inner stage minimization. Denote by $T_{p2}^*(\mathbf{T})$ the processing time required by the second member when unconstrained optimal thresholds are used. Setting $t_d = \tau_2$ in Problem A2, the inner stage minimization becomes that of finding a value of $\bar{T}_{p2}$ that solves

$$\frac{\partial \bar{J}}{\partial \bar{T}_{p2}} \cdot \left[1 - 2 \cdot q_2\right] + \frac{\partial q_2}{\partial \bar{T}_{p2}} \cdot \left[1 - 2 \cdot \bar{J}\right] = 0 \qquad (19)$$

The issue at hand is whether $T_{p2}^*(\mathbf{T})$ satisfies eq.(19). Because $\bar{T}_{p2}^*(\mathbf{T})$ represents a global minimum of $\bar{J}$, the first term in eq.(19) is zero. Now, if $T_{p2}^*(\mathbf{T}) \leq \tau_2$ the second term is also zero, since q_2 does not depend on $\bar{T}_{p2}$ in this region. Thus unconstrained optimal thresholds are solutions when the processing time they require does not exceed the deadline. This is reasonable, since any adjustment of thresholds would have no effect on input/output errors; hence the thresholds can be left at their unconstrained optimal values.

However, for $T_{p2}^*(\mathbf{T}) > \tau_2$ a different result obtains. In this situation, q_2 is monotonically increasing with $\bar{T}_{p2}$. Furthermore, since $\bar{J} < 0.5$, as discussed earlier, it is true that the second term is non-zero and hence $T_{p2}^*(\mathbf{T})$ does not solve eq.(19). This result means that if the processing time required by use of the unconstrained optimal threshold values is greater than that allowed, it is always desirable to adjust t_{20} and t_{21} to reduce T_{p2} and thereby reduce the input/output error q_2.

Minimizing Second Member Input/Output Errors

The discussion above has concluded that, when it is an issue, it is more advantageous to reduce the second member's input/output errors than to retain the best thresholds for processing observations. The question arises as to whether input/output errors should be minimized as much as possible, at the expense of the threshold settings. In terms of Problem A2, this issue is one of whether $\bar{T}_{p2} = \tau_2$ is a solution to the inner minimization, given that $T_{p2}^*(\mathbf{T}) > \tau_2$, or whether $\bar{T}_{p2} > \tau_2$ is a solution instead. Its resolution depends on how drastically the trade of speed for accuracy is made by the team member, which is modeled by the parameter f_s.

To properly investigate this issue, it is necessary to add another constraint to Problem A2 in the inner stage that restricts values of $\bar{T}_{p_2}$ according to the region of interest. The result is the problem

$$\min_{\bar{T}_{p2}} \quad \bar{J}(\mathbf{T},\bar{T}_{p2}) + \left[1 - 2\cdot\bar{J}(\mathbf{T},\bar{T}_{p2})\right]\cdot q_2(\bar{T}_{p2},\tau_2) \qquad (20)$$

$$\text{s.t. } \tau_2 \leq \bar{T}_{p2}$$

where it is assumed that $T^*_{p_2}(\mathbf{T}) > \tau_2$. The necessary conditions for a solution value of $\bar{T}_{p_2}$ are

$$\frac{\partial \bar{J}}{\partial \bar{T}_{p_2}} \cdot \left[1 - 2\cdot q_2\right] + \frac{\partial q_2}{\partial \bar{T}_{p_2}} \cdot \left[1 - 2\cdot\bar{J}\right] - \mu = 0 \qquad (21a)$$

$$\mu\cdot(\tau_2 - \bar{T}_{p_2}) = 0 \qquad (21b)$$

$$\mu \geq 0 \qquad (21c)$$

and the issue is whether $\bar{T}_{p_2} = \tau_2$ is a solution to (21). If so, $\mu > 0$. Furthermore, it must be true that the first two terms in (21a) are positive in sum. The second of the two is always positive, as discussed previously. However, the first is always negative for the region of $\bar{T}_{p_2}$ of interest. This assumes that $q_2 < 0.5$, which is again the assumption that the second member's processing behavior is better than chance level. Furthermore, in the interval where $\tau_2 \leq \bar{T}_{p_2} \leq T^*_{p_2}(\mathbf{T})$, $\bar{J}$ monotonically decreases with increasing $\bar{T}_{p_2}$. That is, as $\bar{T}_{p_2}$ forces the thresholds t_{20} and t_{21} to move away from $t^*_{2i}(\mathbf{T})$, $\bar{J}$ increases.

Thus it is unclear whether $\bar{T}_{p_2} = \tau_2$ satisfies (21a). A more specific test to resolve the ambiguity can be derived as follows. At $\bar{T}_{p_2} = \tau_2$, q_2 is at its minimum: $q_2 = (1 + \exp(f_m))^{-1} = q_{2m}$. Furthermore

$$\frac{\partial q_2(\tau_2,\tau_2)}{\partial \bar{T}_{p_2}} = f_s\cdot(e^{f_m})\cdot(q_{2m})^{-2} \qquad (22)$$

Substituting (22) into (21a) and rearranging gives

$$f_s > -(q_{2m})^2\cdot e^{-f_m}\cdot\left[\frac{1-q_{2m}}{1-\bar{J}(\mathbf{T},\tau_2)}\right]\cdot\frac{\partial \bar{J}(\mathbf{T},\tau_2)}{\partial \bar{T}_{p_2}} \triangleq F_s \qquad (23)$$

which must be satisfied if $\bar{T}_{p_2} = \tau_2$ is a solution. F_s is a non-negative quantity. The parameter f_s models the rate at which input/output errors increase as the processing time required increases beyond the deadline. If $f_s > F_s$, then the marginal increase in q_2 is great enough such that it is optimal to minimize input/output errors and to adjust thresholds accordingly. If $f_s < F_s$, then there exists a compromise between the two extremes - minimum q_2 at $\bar{T}_{p_2}$ or minimum $\bar{J}$ at $\bar{T}_{p_2} = T^*_{p_2}$ - that gives better overall team performance.

Guessing by First Member

Discussion thus far has considered solution characteristics in terms of $\mathbf{T}$, and the conclusions reached pertain to the second member. Turning now to the outer minimization in Problem A1, the question arises as to if and under what circumstances the problem solution involves guessing by the first member. This issue can be resolved by considering, in geometric terms, how feasible (t_1,q_1) values map to $\mathbf{T}$ values.

For fixed a priori probabilities on H (i.e. p_0,p_1), it is possible to characterize all $\mathbf{T}$ values in the (p_{00},p_{11}) plane as t_1 and q_1 range over their values. A region is determined typically as shown in Figure 2. The upper boundary of the region is the locus where $q_1 = 0$. Points Y and Z correspond to where $t_1 \to -\infty$ and $+\infty$, respectively. The lower boundary is the locus of points determined when $q_1 = 1$ and the guessing bias ranges from 0 to 1. Point S corresponds to $g_1 = 0.5$. When viewed as part of the lower boundary, points Y and Z correspond to g_1=1 and 0, respectively. In terms of the underlying (t_1,q_1) values, any point in the interior or on the

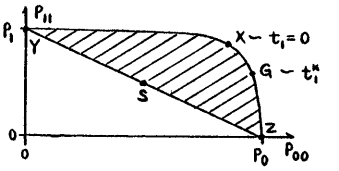


Figure 2 Typical Region of $\mathbf{T}$ Values in (p_{00},p_{11}) Plane

lower boundary represents a non-zero guessing frequency by the team member. Note that the unconstrained optimal value of $t_1=t_1^*$ is therefore on the upper boundary as illustrated. Finally, the geometric representation in Figure 2 has many properties in common with the Receiver Operating Characteristic in signal detection theory [6]. Besides the association of the lower "diagonal" to guessing, it is also the case that better team performance results when the operating point in the (p_{00},p_{11}) plane moves nearer to (p_0,p_1), where perfect discrimination between hypotheses is made (by the first member).

Consider now the outer minimization of Problem A2. While Figure 2 represents possible $\mathbf{T}$ values, not all of them will be feasible due to the constraint on T_{p_1}. Figure 3a shows typically how this constraint restricts

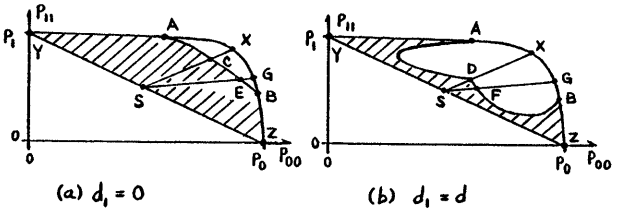


Figure 3 Constraint on T_{p_1} in (p_{00},p_{11}) Plane; g_1=0.5

$\mathbf{T}$ values for $d_1 = 0$, i.e. when the first member has no switching overhead. A guessing bias of 0.5 has been assumed. The arc ACB represents the locus where $T_{p_1} = \tau_1$, and the shaded area designates the region of feasible $\mathbf{T}$ values. A similar depiction is given in Figure 3b, except for the case where d_1 has increased from zero to a relatively significant value. Again, the arc ADB represents the locus where $T_{p_1} = \tau_1$.

The solution to Problem A2 is found by searching over regions such as those in Figure 3. It can be shown, however, that a solution to A2 is such that either $q_1 = 0$ or $T_{p_1} = \tau_1$. This means that the upper boundary of the feasible region contains the solution of Problem A2. In Figures 3a and 3b, therefore, the solution must be on the arcs YACBZ or YADBZ, respectively. In particular, it is possible that solutions will be obtained on the arcs ACB or ADB, i.e. it may be optimal to guess. This can be explained qualitatively as follows. All other things being equal (i.e. neglecting the second member), it is desired to operate in the (p_{00},p_{11}) plane as close as possible to the point where $q_1 = 0$ and $t_1 = t_1^*$. In Figure 3, neither region admits the unconstrained optimal solution as feasible. In Figure 3a, however, point E is closer than point B, where the former is such that $q_1 \neq 0$ and the latter is the nearest feasible point where $q_1 = 0$. In Figure 3b, point B is closer to the unconstrained optimal point. Thus the situation in (a) is likely to have a solution where $q_1 \neq 0$, while in (b) the solution will likely be at point B. Though shown for cases where $d_1 = 0$ or $d_1 \neq 0$, this behavior does not represent a special case, tied to the presence of switching overhead, nor is it dependent on having the bias in guessing at 0.5. Figure 4 shows the same constraints for a bias of $g_1 = 0.75$.

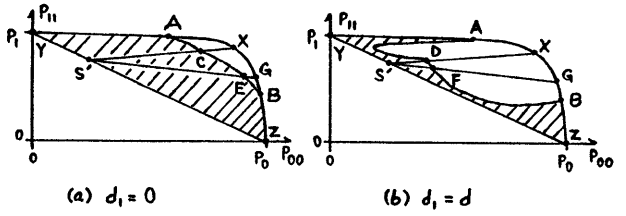


Figure 4 Constraint on T_{p1} in (p_{00}, p_{11}) Plane; g_1=0.75

V. SPECIAL CASE

To highlight particular mechanisms of how one member can affect the other and also team performance, consider the following special case. Suppose that the second member's processing time is independent of the threshold positions, but that it takes longer to use threshold t_{20} than t_{21}. Also, assume that the switching overhead for the second member is significant and that the deadline τ_2 affects the use of t_{20} but not that of t_{21}. That is, mathematically assume that

$$b_{2i} = 0; \quad a_{21} > \tau_2 > a_{20} \tag{24}$$

Also, assume that the first member is unconstrained. For this special case, Problem A1 can be summarized in terms of Figure 5. Since T_{p2} is independent of t_{2i}, its

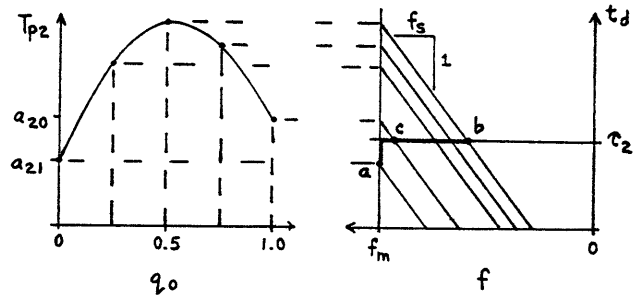


Figure 5 Illustration of Special Case Solution

variation is due entirely to variation in p(u), which is determined by the first team member through placement of t_1. The dependence of T_{p2} on p(u=0) (denoted = q_0) is shown in the left part of Figure 5. The relationship between T_{p2} and input/output errors q_2 (through f) is shown in the right part of the figure. Recall from eq.(11) that a given value of T_{p2} determines a locus of f values as a function of t_d. With $t_d = \tau_2$, a specific operating point on this locus is selected. As q_0 moves from 0 to 1, the resulting T_{p2} values trace out feasible operating points in the right part of the figure, moving from a to b and back to c. Each point on this locus has a minimum detection error probability obtained by solution of the first stage of the minimization. The overall solution thus becomes a matter of searching over t_1 (and thereby q_0) values. The interesting feature of the minimization in this special case is that the trade-off between speed and accuracy required for the second member is governed entirely by the first member. Furthermore, a reduction in T_{p2} depends mostly on reducing the switching frequency. If t_1^* is somewhere near 0, then $q_0 \approx 0.5$ and the optimization problem is essentially one that must weigh two alternatives: either degrade the first member's quality of processing by adjusting t_1 to reduce switching load of the second member and thereby reduce q_2; or accept the higher input/output error rate of the second member in favor of retaining a higher quality of processing by the first.

Once the solution is obtained, the thresholds will be set at the solution values and the team will presumably operate as modeled. By way of illustrating how processing load and performance can interrelate, suppose that after the team has been set into operation the constraint on the first member becomes binding, say due to external factors that reduce the value of τ_1. As per design, the team member can resort to guessing to meet the constraint. Figure 6 shows a trajectory in the

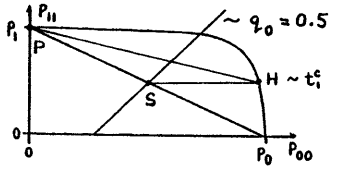


Figure 6 Illustration of Special Case Operation

(p_{00}, p_{11}) plane corresponding to increasing q_1 for two biases in guessing. Point H corresponds to the solution operating point (with $t_1 = t_1^c$). Points S and P correspond to completely random operation with guessing biases of 0.5 and 1.0, respectively. The locus of where $q_0 = 0.5$ has also been shown. As q_1 increases the operating point moves away from H to either S or P. Because the movement is toward the diagonal "guessing" line, team performance will generally be worse. However, a significant qualitative difference is apparent. Along the trajectory HS, T_{p2} is increasing and in fact comes to rest where switching frequency is at its maximum. Performance thus not only degrades because of changes in J but also because of an increase in q_2. Along the trajectory HP, however, T_{p2} first rises due to the increase in switching, but decreases as switching overhead goes to zero. In this case the contribution to performance degradation due to input/output errors is less. Within these cases are examples of increasing processing load and degrading performance, as well as decreasing processing load and degrading performance.

V. SUMMARY

The addition of processing time constraints to a team theoretic problem modifies team operation. In particular, partially random behavior by team members can be optimal, either by a member's choice, through the selection of an option to guess; or by design, through selection of thresholds such that processing time exceeds a deadline, which in turn makes processing errors more likely. Furthermore, the special case considered has demonstrated that a variety of relationships can exist between team performance and member workload. Because of this variability, a general guideline is suggested, whereby a first step toward understanding a particular structure might be to identify which of the possible relationships actually exists. The effects of switching, as seen in the special case, also suggest a principle of general interest. Given that changing tasks or procedures may require processing resources, and that the necessity to switch may be governed by another team member, the recognition of the potential for switching within a team structure may lead to a better understanding of team behavior.

References

[1] Y.C. Ho, "Team Decision Theory and Information Structures," Proc IEEE, Vol. 68, June 1980, pp. 644-654.

[2] L.K. Ekchian and R.R. Tenney, "Detection Networks," Proc 21st IEEE Conf Dec and Cont, 1982, pp.686-691.

[3] J.I. Yellot, "Correction for Fast Guessing and the Speed-Accuracy Tradeoff in Choice Reaction Time," J. Math Psych, Vol. 8, 1971, pp. 159-199.

[4] R.W. Pew, "The Speed-Accuracy Tradeoff," Acta Psych, Vol. 30, 1969, pp. 16-26.

[5] D.G. Luenberger, Intro to Linear and Nonlinear Programming, Reading, MA: Addison-Wesley, 1973.

[6] H.L. Van Trees, Detection, Estimation, and Modulation Theory, Vol. I, New York: John Wiley, 1968.

CALCULATING TIME-RELATED PERFORMANCE MEASURES OF A DISTRIBUTED TACTICAL DECISIONMAKING ORGANIZATION USING STOCHASTIC TIMED PETRI NETS[1]

R.Paul Wiley and Robert R. Tenney

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

I. INTRODUCTION

One of the key categories in which a distributed tactical decision making organization must be evaluated to assess its effectiveness is time-related performance measures. Examples of these measures include: the rates at which different tasks are completed, the probability that shared resources are available when requested, the average number of tasks waiting to be performed, the percentage of tasks successfully completed, etc. These performance measures depend strongly on the delays in decisionmaking due to organizational architecture and associated coordination protocols. If we are able to understand the effects of these architectures and protocols on the performance measures, we will be one step closer to being able to design effective distributed, real-time decisionmaking organizations.

A realistic model of distributed real-time decisionmaking organizations must incorporate at least three features: asynchronous protocols, concurrent operations, and random task-completion times. The asynchronous protocols are necessary since the groups or agents which compose the organization cannot be tightly synchronized; i.e. they do not communicate with each other at prespecified times. The concurrent operations are necessary since different parts of the organization work independently, coordinating their activities regularly to make sure the overall objective is being achieved. Finally, random task-completion times are necessary since most agents or groups perform a wide variety of tasks under many different conditions.

Stochastic Timed Petri Nets (STPNs) naturally incorporate all of these features. The basic objective of this paper is to introduce a methdology that can be used to analyze the dynamic and steady state behavior of these nets and, consequently, enhance our understanding of the time-related issues of the organizations they model.

The methodology presented in this paper can be used to analyze live and save STPNs (terms defined later). This class of STPNs can model organizations with finite queues and interesting protocols such as priorities and/or probabilistic choices. For organizations that can be modeled with this class of STPNs, the time-related performance measures listed in the first paragraph, among others, can be calculated.

So as not to lose sight of applications in the abstraction of mathematics, all steps of the methodology will be illustrated by an example. This example models the use of a shared resource by three groups. The groups could correspond to three naval battle groups, each responsible for a geographical sector, collectively in charge of defending a naval task force against submarine attacks. The shared resource in this situation could correspond to a special purpose helicopter, based on the main carrier of the task force, equipped with bouys and sonar equipment (i.e., a LAMPS helicopter). Periodically, the captain of each battle group will request the use of the helicopter.[2] We assume that the probability distribution for the time between requests for each battle group captain is known, as well as the probability distribution for the time that the helicopters will spend in each sector. In addition, we assume that the protocol or decision rule that decides which among the three captains will get use of the helicopter, in case of a conflict, is assumed (or hypothesized). Given the required information, this shared resource problem can be modeled with a STPN and such questions as the percentage of time the helicopter (the shared resource) is being used and by whom, the average amount of time that each captain must wait after he has requested use of the helicopter, the rates at which each captain will get use of the fleet, etc., can be answered.

An overview of this paper goes as follows. We start by defining STPNs and presenting the relevant concepts. We then show how the operation of a STPN can be viewed in terms of the operation of an infinite collection of Unfolded STPNs. This decomposition is then used to write state equations for the system, which in turn can be used to analyze the STPN (and the organization it models) with respect to time-related performance measures.

II. STOCHASTIC TIMED PETRI NETS

Stochastic Timed Petri Nets are graphs with two types of nodes: places (drawn as a circle and labeled p_i) and transitions (drawn as a line segment and labelled t_i), along with directed arcs going from one type of node to the other (see Figure 1). Besides

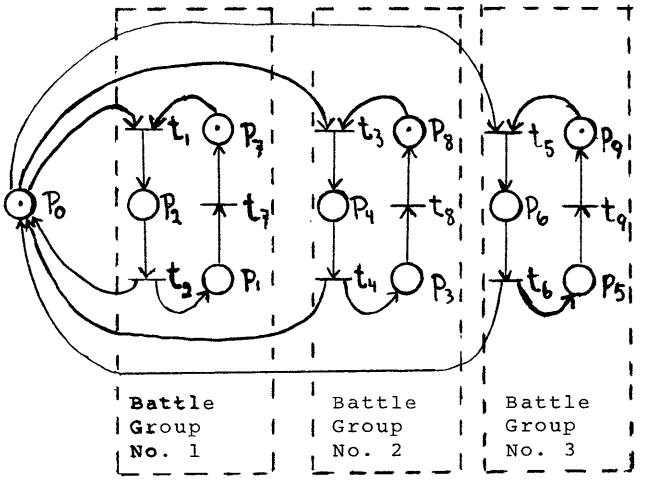


Figure 1. STPN Model of Naval Defense Shared Resource Example

nodes and arcs, a STPN assigns to every place a non-

[1]This research was supported by the Office of Naval Research under grant ONR/N00014-77-C-0532 (NR 041-519) and ONR/N00014-84-K-0519 (NR 649-003).

negative number of tokens, called the marking of the net, and a nonnegative random processing time[3]. When a token arrives at a place, it is defined to be unavailable (following Sifakis [3]). The token remains in this state until the instant when the processing has finished. At this time, the token becomes available. We assume that the processing time probability distributions for every place are given, and that they are stationary and independent from place to place.

In the STPN shown in Figure 1, the shared-resource example, a token arriving at p_1 represents the helicopter being sent to geographical sector No. 1. The token being processed in that place represents the helicopter performing its tasks in that sector. And finally, the token becoming available again represents the helicopter completing its tasks, a time after which it is available to be sent elsewhere.

Tokens move around by transition firings. A transition is enabled, that is, it may fire, only when all of its input places have available tokens. When a transition does fire, it removes a token from all of its input places and adds a token to all of its output places. If transition t_1 in Fig. 1 fires, for example, we remove a token from places p_0 and p_1 and add a token to place p_2.

The firing of the mentioned transition t_1, in terms of the naval example, represents the act of sending the LAMPS to the first geographical sector. Before the LAMPS will be dispatched, however, two facts must be true. The LAMPS (represented by a token in p_0) must be available, and the captain of the first geographical sector must request its use (request represented by a token in p_7). Thus, transition t_1 coordinates the requests from the captain of Battle Group No. 1 with the availability of the shared resource. Moreover, while the LAMPS is performing its tasks in the first geographical sector, the captains of the other geographical sectors cannot use it. In the STPN, this restriction is enforced by the fact that, during this time, no token is available in p_0 to fire transitions t_3 or t_5.

Two or more transitions are said to be in conflict if firing one will disable the others. In Fig. 1, transitions t_1, t_3 and t_5 are in conflict. If an enabled transition is not in conflict, it fires instantly. If several enabled transitions are in conflict, then a decision rule, specified a priori, selects one and that transition fires instantly. These decision rules can depend on the relative firing times of the transitions that feed tokens into the input places of the transitions which are in conflict.

In the naval defense example, this conflict represents the situation where two or more captains request the use of the LAMPS, while it is in service in the area. A decision rule must be provided to settle this conflict. This decision rule can be a function of the relative firing times of t_7, t_8, t_9 and either t_2, t_4 or t_6 (whichever transition fed the token into p_0). Possible decision rules include priorities, and the fleet assigned accordingly, or probabilistic choices, whereby the fleet is assigned according to some probabilistic rule (which might account for varying battle conditions), or some combination of the above.

In the preceding paragraphs, we have formulated the rules of operation for a STPN. In order to implement these rules, it is important to determine what information must be recorded in order to predict the future behavior of the system. That is, we need to determine the "state" of a STPN. One fact we need to know, certainly, is the position of every token (the marking of the net) since this specifies what transitions will potentially be enabled. In addition, we need to know the amount of time that each token has been in each place to determine whether it is available or not. One way of determining the amount of time a particular token has been processed is to remember the time at which the transition that created that token fired, and subtracting that time from real time. In the next section, we will take this last approach. Thus, a state for a STPN consists of the marking of the net plus the time that each token has been in each place. We can further argue that if any of this information is missing, then we do not have enough information to predict the future behavior of the system. Thus, the mentioned state is also minimal.

There is one last item that we must specify in order to follow the evolution of a STPN, and that is the initial conditions. Given the previous discussion on the state of a STPN, we assume that the initial marking and the time that each token has been in each place at time zero are given. With the given initial conditions, the known decision rules, and the processing times determined by their respective probability distributions, the STPN evolves autonomously in time. It is this autonomous evolution what we wish to study, understand, and compute performance measures for.

One can think of the operation of a STPN as a succession of markings, with the changes from one marking to another specified by the transition firings and the rules for moving tokens. If the number of tokens in any particular place can never exceed one, regardless of what processing time probability distributions and decision rules are assigned, the STPN is said to be safe. If the probability that any particular transition firing gets arbitrarily close to one if we wait long enough, regardless of what state the STPN has reached, the STPN is defined to be live. And finally, if there exists a directed path from any node (place or transition) to any other node, the STPN is strongly connected.

Throughout this paper, we will assume that the STPNs we are studying are strongly connected, live, safe, and with a finite number of nodes. The first assumption is a necessary condition for the STPN to be safe (Ramchandani [2]) and also ensures that we are not solving a problem that could be split into two or more independent problems. The second assumption, liveness, says that all parts of the system will operate regularly. If there exists a transition that is not live, then the part of the system associated with that transition should be deleted since it is not performing any activities. Liveness is also a necessary condition for convergence. The third condition, safeness, is necessary to impose ordering. If we allowed multiple tokens in a place, then they would be free to cross each other depending on their variable processing times. In order not to keep track of all possible orderings of tokens, we require safeness. Notice that this still allows us to model (bounded) finite first-in first-out queues by concatenating a series of safe places, as shown in Figure 2.

Now that all the necessary concepts have been defined, we can specify the performance measures that we are interested in obtaining. These performance measures are the average transition firing rates, the average number of tokens in each place, the average amount of time a token spends in each place, and finally, the probability that a token will enable one of several transitions which may be in conflict. The basic methodology actually obtains much more since we also obtain the probability distribution (and not just the mean value) for the time a token spends in each

[2]For simplicity, we assume that there is only one helicopter. Generalization to many is easy.

[3]Many authors, Ramchandani [2], Zuberek [5] and Molloy [1], associate processing times with transitions instead of places. The distribution makes no difference for the class of problems studied in this paper.

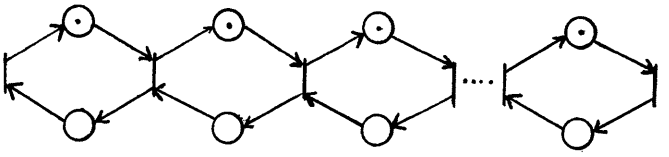


Figure 2. STPN Model of a Finite First-In First-Out Queue

place. From the list of performance measures, most, if not all, of the time-related performance measures of the original organization can be answered.

III. STATE EQUATIONS

III.a Introduction

In the previous section, we established that the state of a STPN consists of the marking of the net and the time each token has spent in each place. We also mentioned that one could deduce the time a token spends in each place by recording the firing time of the transition that fed the token into that place. In this section, we will derive equations which keep track of the transition firing times and from which the marking can be deduced. Thus, in light of the previous discussion, these equations can be used to follow the state of the STPN as it evolves.

The derivation is performed in two steps. In the first step, we show that the evolution of a STPN can be visualized in terms of the evolution of an infinite collection of Unfolded STPNs. This visualization provides an ordered index that can be used to decribe the system mathematically. In the second step, the mentioned index is used to write firing time state equations.

III.b Unfolded STPNs

One of the main difficulties of analyzing STPNs, or any other mathematical model of asynchronous concurrent systems, is obtaining an ordered index which can be used to describe the system mathematically. The usual index, time, is not appropriate since different parts of the system work independently. But STPNs are causal systems, that is, certain transitions must fire and certain tokens must be processed before other transitions can fire. The Unfolded STPNs, which we discuss in this subsection, capture this causality and provide the ordered index that we seek.

An Unfolded STPN is constructed by taking the original STPN, cutting every directed circuit at an appropriate transition, and "unfolding" the net. For example, Figure 3 shows the Unfolded STPN corresponding to the STPN of Fig. 1.

As can be seen from this example, every place and transition is labeled by the superscript k. This variable indicates to which copy of the Unfolded STPN the places and transitions belong. This labelling is important since, as we already mentioned, the operation of the regular STPN can be viewed in terms of the operation of an infinite collection of Unfolded STPNs. Also observe that the Unfolded STPN begins and ends with the same set of transitions, each transition in the first set labeled with the variable k and each transition in the second set labeled with the variable k+1. This fact does not mean that subsequent copies of the Unfolded STPN are connected. Rather, it means that there are two copies of each of these transitions. The reason we need duplicate copies of these transitions will become obvious later.

As mentioned previously, the operation of the regular STPN can be viewed in terms of the operation of

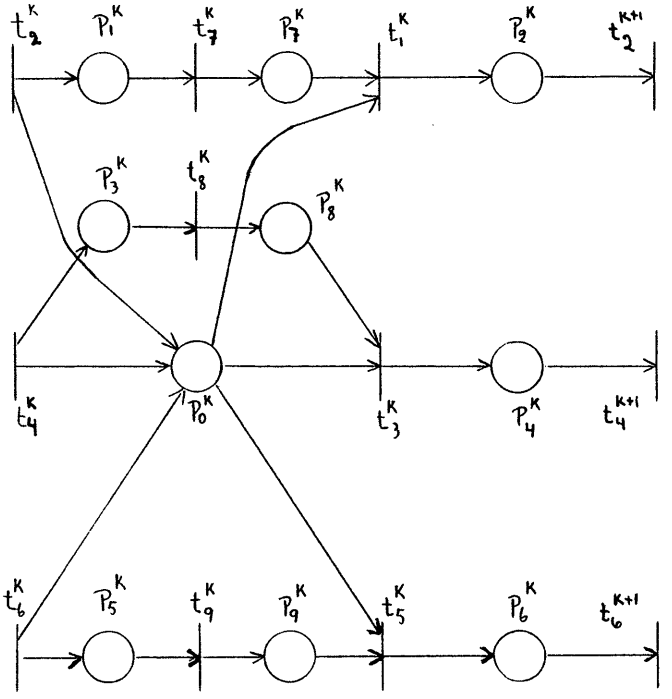


Figure 3. Unfolded STPN Corresponding to Shared Resource STPN of Fig. 1

an infinite collection of Unfolded STPNs. Since the latter form a disjoint collection, we must specify what we mean by the operation of the collection Unfolded STPNs. We do this now.

Imagine having an infinite collection of Unfolded STPNs numbered k=1,2,3,... The k^{th} copy of this collection is referred to as the k^{th} stage. Now suppose we place a token in every place of stage 1 for which the corresponding place in the regular STPN contains a token in the initial marking. Also suppose that the time each of these tokens has been processed at time 0 is initialized to be the same as the initial times specified for the regular STPN. Then we can implement the rules of operation that we describe in the previous section for stage 1. Since the Unfolded STPN contains no directed circuits, two things can happen to the tokens. The first possibility is that they will leave the stage by having some of the transitions labelled with the superscript 2 fire. The other possibility is that the tokens will be deadlocked.

As an illustration, consider the STPN of Fig. 1 along with its corresponding Unfolded STPN in Fig. 3. Initially, stage 1 contains a token in p_0^1, p_7^1, p_8^1 and p_9^1. Also initialized are the firing times for transitions t_2^1, t_7^1, t_8^1, and t_9^1 (this is equivalent to initializing the times each token has been processed at time 0). Now, suppose the given decision rule picks t_1^1 as the transition to fire (since t_1^1, t_3^1 and t_3^1 are in conflict). This firing takes a token from p_0^1 and p_1^1 and places a token into p_2^1, where it is processed. After that token is processed, transition t_2^2 fires, and that token leaves stage 1. Meanwhile, the tokens in p_3^1 and p_5^1 are deadlocked since the output of those transitions will never fire.

So far we have explained how stage 1 of an Unfolded STPN will operate. To initialize stage 2, we must remember which transitions fired at the end of stage 1. A token is placed in the output place of

every transition that fired. Thus, for the example that we have been considering, a token is placed of p_0^2 and p_1^2. We also place a token in every place of stage 2 whose corresponding place in stage 1 contains a deadlocked token. In the example, this corresponds to placing a token in p_8^2 and p_9^2. The transition firing times that originally fed these tokens are also recorded in stage 2. In the example, the firing time of $t_4^2(t_6^2)$ is set to be the same as the firing time of $t_4^1(t_6^1)$. Now, stage 2 has all the necessary state variables initialized and we can implement the rules of operation for it.

Thus, the collection of Unfolded STPNs operates one stage at a time. Each stage evolves according to the rules of operation until every token has left the stage or is deadlocked. Then the following stage is initialized according to the discussion given above, and the collection of Unfolded STPNs keeps operating in this manner forever.

It is not hard to see that the operation of the Unfolded STPNs reflects the operation of the original STPN. In mathematical terms, there exists a one to one correspondence between the sample paths of the Unfolded STPNs and the sample paths of the regular STPN (see [4] for proof). This correspondence allows us to study the evolution of the regular STPN by studying the evolution of the corresponding Unfolded STPNs. This approach is exactly the one we shall take.

III.c Transition Firing Times

The evolution of the Unfolded STPNs can be divided into two separate processes. The first process keeps track of the evolution from one marking to the next which results from transition firings. The second process keeps track of the transition firing times and the processing of tokens. The two processes are connected since the first process, through its markings, determines which transitions will potentially be enabled while the second process actually picks the transition which fires next. The rules for keeping track of markings have already been given in Section II. Therefore, in this section, we will derive equations which will enable us to keep track of the transition firing times.

In order to perform this derivation, consider the STPN shown in Figure 4, which we assume is part of a larger Unfolded STPN. Let $x_i(k)$ be the firing time for

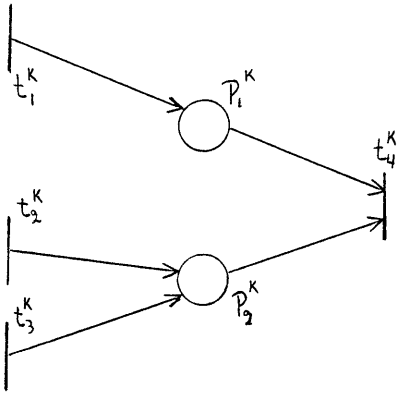


Figure 4: STPN for Derivation of State Equations

transition t_i^k and $z_i(k)$ be the processing time for place p_i^k. Our objective is to obtain an equation which will give us the numerical value of $x_4(k)$ given $x_1(k)$, $x_2(k)$, $x_3(k)$, $z_1(k)$ and $z_2(k)$. In order to do this, we must know at what times the tokens entered p_1^k and p_2^k. The time the token entered p_1^k is given by $x_1(k)$, and the time the token entered p_2^k is given by $\max[x_2(k), x_3(k)]$. We can deduce this last fact since we know only t_2^k or t_3^k fired, and the latest firing is the one which fed the token into p_2^k. After each token enters their respective places, they are processed. When both tokens have finished being processed, transition t_4^k fires. We conclude that

$$x_4(k) = \max\{z_1(k) + x_1(k),$$

$$z_2(k) + \max[x_2(k), x_3(k)]\}$$

If transition t_4^k had contained more than two input places, then the main maximization operator in this last equation would have contained more terms. Similarly, more input transitions to places p_2^k or p_3^k could also be easily handled.

Firing time equations can be written in the manner prescribed above for all transitions that fire in the k^{th} stage. Which transitions fire in each stage depend on the initial marking of that stage and the decision rules. As specified previously, the evolution of the k^{th} stage of the Unfolded STPN will terminate when transitions corresponding to the next stage fire or when the tokens become deadlocked. To initialize the calculations in the next stage, the deadlocked tokens must be "transported" to the corresponding places in the next stage and the firing times of the transitions that fed these tokens must also be initialized.

Let us illustrate this discussion with the shared resource example. Suppose the initial marking of stage k places a token in p_0^k, p_1^k, p_8^k and p_9^k, and that the known firing times are $x_2(k)$, $x_8(k)$, and $x_9(k)$. Further assume that t_2^k is the transition that fed the token into p_0^k. The only transition that can fire from this state is t_7, and

$$x_7(k) = z_1(k) + x_2(k).$$

Now let the token in p_7^k be processed. Once this has been accomplished, the decision rule picks t_1^k, t_3^k or t_5^k to fire. Suppose it picks t_3^k. As a result of that decision t_4^{k+1} will also fire. The equations are

$$x_3(k) = \max\{z_8(k) + x_8(k), z_0(k) + x_2(k)\}$$

and

$$x_4(k+1) = z_4(k) + x_3(k).$$

As a result of all these firings, a token is placed in p_3^{k+1} and p_0^{k+1}. Meanwhile, the tokens in p_7^k and p_9^k are deadlocked. Therefore, these tokens must be moved to p_7^{k+1} and p_9^{k+1}, respectively, and the firing times of t_7^{k+1} and t_9^{k+1} initialized as follows:

$$x_7(k+1) = x_7(k)$$

and

$$x_9(k+1) = x_9(k)$$

Thus, we started with a state in stage k and ended up with a state in stage k+1. The state equations can be used in this manner to follow the evolution of the Unfolded STPNs, and of the regular STPN that it is derived from.

III.d Basic Methodology

Similar state equations to the ones just discussed can be derived for any live and safe STPN. Once these equations are obtained, they can be used to analyze the STPN with respect to the time related performance

measures cited previously. The basic methodology uses the state equations to recursively calculate the probability distributions of the states as a function of k, the ordered index that tells us what stage the evolution of the Unfolded STPNs is going through, starting from the initial state. The details, algorithms, and conditions necessary for convergence and uniqueness are given in [4].

The state equations that we derive for a STPN could also be used to simulate the system by using random draws to determine the processing times and the probabilistic decisions. The shared resource example that we discuss in the next section was solved by using both methods: analysis and simulation. As they should, the values of performance measures obtained by both programs agree.

IV. COMPUTER EXAMPLE

Throughout this paper, we have used the naval defense shared resource example to motivate the study of STPNs and explain the steps of the methodology. In this section, we present the results of a computer program that implements this methodology and interpret some of the results in terms of the naval defense example.

As noted in Section I, inputs to the methodology are the probability distributions for the processing times of every place, and the decision rule which determines which battle group will get use of the LAMPS helicopter in case of conflicting demands. The processing time probability distributions are shown in Fig. 5.

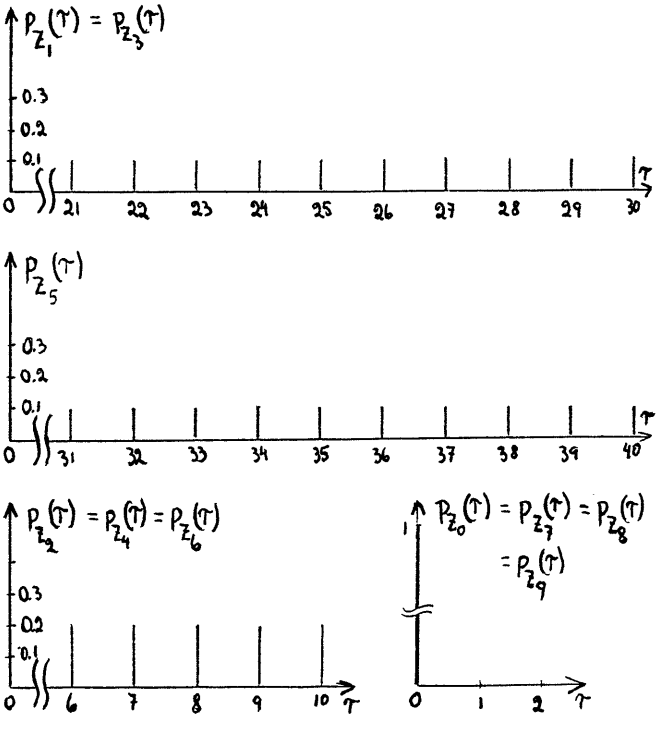


Figure 5. Processing time Probability Distributions.

The decision rule that the program uses is assigning equal priorities to Battle Groups Nos. 1 and 2 and sending the LAMPS helicopter to whichever has been waiting the longest. (In case of a tie, the helicopter is sent to the Battle Group No. 1.) Battle Group No. 3 is given lowest priority. The LAMPS will be sent there only if the captains of the other battle groups have not requested it. The decision rule just described might be reasonable if the submarine attack is most likely to come from the geographical regions protected by Battle Groups Nos. 1 and 2, and is less likely to come from the geographical region protected by Battle Group No. 3.

The results of the program are listed in the following table. From these results, we can deduce

Place Number	Expected Delay	Expected Number of Tokens
0	4.7874e+00	3.7438e-01
1	2.5500e+01	7.2613e-01
2	7.9999e+00	2.2780e-01
3	2.5500e+01	7.2354e-01
4	7.9999e+00	2.2699e-01
5	3.5500e+01	7.5795e-01
6	7.9999e+00	1.7080e-01
7	1.6173e+00	4.6055e-02
8	1.7429e+00	4.9454e-02
9	3.3363e+00	7.1233e-02

Transition Number	Expected Rate	Probability of Decision
1	2.8475e-02	3.6424e-01
2	2.8475e-02	1.0000e+00
3	2.8374e-02	3.6272e-01
4	2.8374e-02	1.0000e+00
5	2.1350e-02	2.7302e-01
6	2.1350e-02	1.0000e+00
7	2.8475e-02	1.0000e+00
8	2.8374e-02	1.0000e+00
9	2.1350e-02	1.0000e+00

Table. Output of Computer Programs

that the LAMPS is idle 37% of the time (average number of tokens in p_0), that the probability of the LAMPS being sent to Battle Group No. 2, once it is available is .36 (probability of decision t_3), that the captain of Battle Group No. 3 must wait, on the average, 3.3 units of time after a request for the LAMPS (average time token spends in p_9), that the LAMPS is sent out at an average rate of .078 (sum of t_1, t_3 and t_5 firing rates), etc. These and similar measures give a complete picture of the situation with respect to time-related performance measures.

V. CONCLUSIONS AND FUTURE RESEARCH

We have shown, how STPNs can be used to model distributed tactical decisionmaking organizations. We then outlined the methodology by which STPNs can be analyzed with respect to time-related performance measures. Finally, the modeling and methodology were applied to a simple, yet interesting, naval defense shared resource example. The results clearly indicate that the study of STPNs will help us understand the dynamic and steady state behavior of the organizations they are capable of modeling.

Future research directions are many. One of these is modeling of more complex organizations using STPNs.

In the naval defense example, for instance, we could have modeled the asynchronous protocols and delays of each battle group's own operations. These models could then be substituted for places p_1, p_3, and p_5, and the resulting analytical program would provide a more accurate assessment of the situation. Another area of research is performing a sensitivity analysis to small changes in processing times. This analysis could help an organization designer understand what changes could be made to improve the organization. And finally, a third area of research is finding alternate, faster ways of using the state equations to find the steady state probability distributions. The methodology outlined in this paper calculates these distributions recursively, which can converge rather slowly.

References

[1] M.K. Molloy, "Performance Analysis Using Stochastic Petri Nets", IEEE Transactions on Computers, Vol. (AS-24, No. 7, (July 1977), pp. 400-405.

[2] C. Ramchandani, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Technical Report 120, Lab. for Computer Science, M.I.T., Cambridge, MA (1974).

[3] J. Sifakis, "Use of Petri Nets for Performance Evaluation". Measuring, Modeling and Evluating Computer Systems, H. Beilner and E. Gelenbe, eds., North-Holland, pp. 75-93.

[4] R.P. Wiley, "Performance Analysis of Stochastic Timed Petri Nets", Ph.D. Thesis, Electrical Engineering and Computer Science Department, M.I.T., Cambridge, MA (expected December 1985).

[5] W.M. Zuberek, "Timed Petri Nets and Preliminary Performance Evaluation", The 7th Annual Symposium on Computer Architecture Conference Proceedings (May 6-8, 1980), pp. 88-96.

COMPUTATION OF DELAYS IN ACYCLICAL DISTRIBUTED DECISIONMAKING ORGANIZATIONS*

Victoria Yu-yu Jin
Alexander H. Levis

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

An algorithm for computing time delays in a distributed decisionmaking system is developed. Starting with a matrix representation of the organizational structure all possible information processing paths are scanned and the time delay associated with each one is computed. When the decision strategies are known, the expected delay of the overall system can be obtained.

1. INTRODUCTION

Decisionmakers in a distributed system have access to specified information and control specified resources. Usually, even in a simple organization, there is more than one path through which information can be processed. Decisionmakers can choose the path. There is no general rule for predicting which path will be chosen, because each individual decisionmaker has a different personality, different skills and reacts differently to different circumstances.

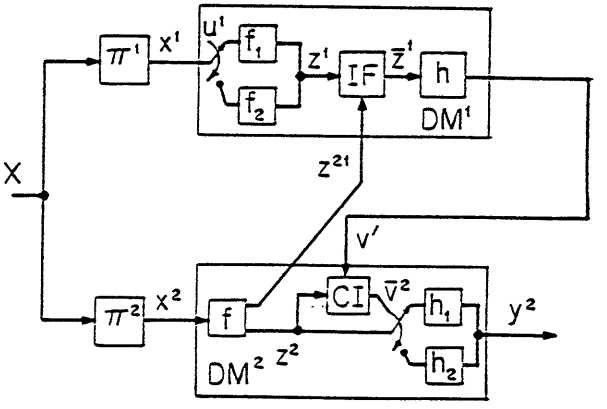
There are many measurements of performance of distributed decisionmaking systems (DDMs). One of the most important is the time interval from the moment a stimulus is received by a system to the moment a response is made. This time delay is one indicator of a system's ability to respond to events in a timely manner (see Cothier and Levis, 1985).

To evaluate the time delay in DDM systems, all possible information-processing paths must be identified and then the time delay associated with each path computed. A DDM system is often a large-scale system which contains many decisionmakers and decision support systems with complicated interconnections. For these systems, scanning all possible paths and computing time delays can become difficult or even impossible. An algorithm is required to solve the problem. Then, protocols that reduce the time delay in the operation can be designed, so that the effectiveness of the system can be improved [Jin, 1985].

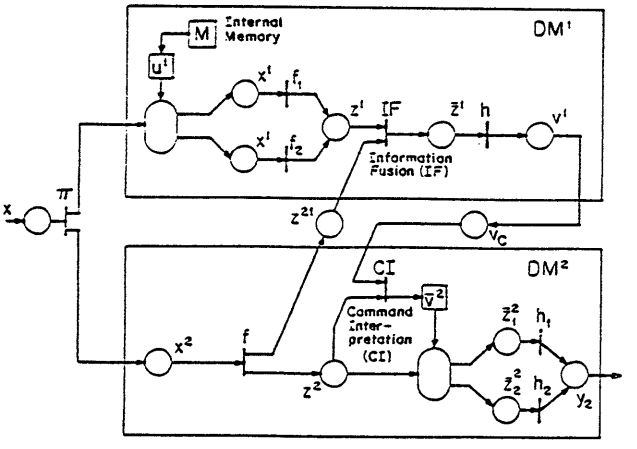
In this paper, such an algorithm for computing time delay of DDM systems is developed, which scans all paths and computes conditional probabilities and time delays associated with these paths. From these results, a tree can be established to show all possible paths explicitly, the probability that each possible path occurs is easily calculated, and the expected delay of the overall system can be obtained.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under Contracts N00014-83-K-0185 (NR 247-349) and N00014-84-K-0519 (NR 649-003).

The algorithm is developed by using the Petri Net representation of decisionmaking organizations [Tabak and Levis, 1985] which shows explicitly the interactions between DMs and the sequence of operations in the system. Figure 1 shows the Petri Net representation of a two decisionmaker system; details of the procedure for constructing the Petri Net can be found in Tabak and Levis (1985).



(a) A block diagram representation



(b) An equivalent Petri Net representation

Figure 1. An Example of a Two Decisionmaker System

2. COMPUTATION OF TIME DELAYS

2.1 DEVELOPMENT OF AN INTERCONNECTION MATRIX

The information contained in a Petri Net can be summarized in the system matrix A_S, which is relatively complicated because of its compact form.

Therefore, it is not convenient for scanning for all possible paths. An interconnection matrix, C_s, is needed that indicates the inter-connections between DMs in an explicit way so that all possible paths can be found using a simple algorithm. The interconnection matrix is obtained by scanning the system matrix and storing the relevant interconnection information in a new format. The interconnection matrix indicates whether the components are connected (+1, -1) or not connected (0); and, if connected, how they are connected.

The elements of the interconnection matrix, C_s, are defined as follows:

1. C_s has a dimension of m x n, where m is the total number of arcs (or links) in a Petri Net and n is the total number of transitions.

2. An element of C_s, $\{c_{ij}\}$, gives the connection status of the j-th transition to the i-th arc;

3. The element c_{ij} can take the values of -1, 0, +1:

$$c_{ij} = \begin{cases} -1, & \text{when there is an output from the j-th transition to the i-th link;} \\ 0, & \text{when there is no connection between the j-th transition and the i-th link;} \\ +1, & \text{when there is an input from the i-th link to the j-th transition.} \end{cases}$$

As an example, Table 1 shows the system matrix A_s for the two-decisionmaker organization of Figure 1. In the system matrix, each decisionmaker (DM) is considered as a subsystem. The information source (AIN) and the response link (AOU) are also subsystems. Therefore, the total number of subsystems in a DDM system with n DMs is n+2. Each transition of a Petri Net is modeled as a column in a submatrix, which contains the input and output information for the transition.

Table 1. The System Matrix of the Two DM System

$A_s =$

AIN	1	0	0	0
	0	0	0	0
	0	0	0	0
	1111	0	0	0
	2	0	0	0
	1	0	0	0
	1.12	0	0	0
	2	0	0	0
	1	0	0	0
A1	1	1	1	1
	3333	3333	2	0
	1	1	1	0
	0	0	0	2
	0	0	0	2
A2	1	1	1	1
	3333	1	0	0
	1	4	0	0
	1	0	4444	4444
	3	0.12	1	1
AOU	1	0	0	0
	2	0	0	0
	3.12	0	0	0
	0	0	0	0
	0	0	0	0

The elements of each column indicate the origins of the inputs to that transition and the destination of its outputs. For example, column 1 in AIN (Table 1) indicates that the delay of this transition is one unit; the second element shows that there are no inputs and hence the third, which would otherwise indicate the source, is also zero. The fourth entry, 1111, is a code indicating multiple outputs. There are 2 outputs (fifth entry); the first output is routed to decisionmaker #1 (1 in position 6), to his first transition which is a two-way switch (the seventh entry, 1.12). The other output goes to decisionmaker #2 and, specifically, to his first transition (eight and ninth entries). For details see Tabak and Levis (1985). The corresponding interconnection matrix is given in Table 2.

Table 2. The Interconnection Matrix for the Two DM System

$C_s =$

-1	1	0	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0	0	0
-1	0	0	1	0	0	0	0	0	0
0	-1	0	0	1	0	0	0	0	0
0	0	-1	0	1	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0
0	0	0	0	0	-1	1	0	0	0
0	0	0	-1	1	0	0	0	0	0
0	0	0	-1	0	0	1	0	0	0
0	0	0	0	0	0	-1	1	0	0
0	0	0	0	0	0	-1	0	1	0
0	0	0	0	0	0	0	-1	0	1
0	0	0	0	0	0	0	0	-1	1

After C_s is established, it is necessary to check whether the sum in each row is zero. If it is not, there must be an error because each row of C_s stands for only one link which connects two vertices in a certain direction. Therefore, there must be a -1 to indicate that link is an output of one of the vertices and a +1 to indicate that it is an input of the other vertex.

2.2 SCANNING ALL POSSIBLE PATHS

The scanning problem is formulated as finding all possible paths from the vertex that represents the input source to the vertex that is the output sink. The paths form a tree with the input source as the main root of the tree. Every path is a branch of the tree.

The Algorithm

Let P(m,z) be the z-th subpath ending at the m-th vertex and D_{mz} be the time delay associated with this subpath. The elements of C_s are partitioned into four subsets: S_1 and $\bar{S}_1 = C_s - S_1$, S_2 and $\bar{S}_2 = C_s - S_2$ with $S_1 = \{1\}$ and $S_2 = \{-1\}$.

The elements of C_s have the following properties:

(1) If $c_{ij} = 1$ and $c_{ik} = -1$, vertices j and k are connected and V_j precedes V_k.

(2) If there are more than one (-1) in column j of C_s, vertex V_j is a root or a subroot.

(3) If there are n (+1) in column j, then n paths converge into the same path after they reach vertex V_j.

Scanning is done backwards, that is, it starts from the last vertex of the output sink, V_i, in which

there are usually more than one input. The first positive one (+1) in the i-th column of C_s is the first input to V_i, and it is processed first. The processing stops when a multi-input vertex, V_j, is found, i.e., there are several paths converging into transition V_j. To avoid iterative computation, V_j is stored as a subroot and is marked as the end of some subpaths. Then scanning goes back to the second input of V_i. The previous procedure is repeated until a new convergent vertex, say, V_k, is found. After all inputs of V_i are processed, the same procedure is repeated for all the subroots. When the subpaths of the last subroot end with V_1, which is the first transition of the source, scanning is completed.

After all subpaths are found, they are assembled into paths by matching the last vertex V_k in the subpath P(i,j) to the first vertex V_k in P(k,z). When the last vertex of a subpath is 1, a path is completed.

The algorithm depends on the following rule.

Let $c_{ik} \in S_1$ and $c_{hj} \in S_2$, i.e., $c_{ik} = +1$ and $c_{hj} = -1$.

If $i = h$

then there is a path from V_j to V_k, i.e.,

$$P(k,z) = V_j \rightarrow V_k$$

and the delay associated with this path, D_{jz}, is

$$D_{kz} = D_j + D_k,$$

the sum of the delays of the two vertices in the path.

One important rule of the algorithm is that no loop is allowed in any path. If a vertex appears in one path more than once, scanning stops. An error message is given.

Consider System A (Fig. 1); its system matrix A_s and interconection matrix C_s were given in Tables 1 and 2. There are 10 vertices and 13 edges. All paths are scanned using the algorithm. Related subpaths are then joined together to complete possible paths. The subpaths ending with the n-th vertex are connected to the subpaths starting on the n-th vertex to form intermediate paths. Table 3 lists all subpaths and intermediate paths for this example.

Often some possible paths are active simultaneously, i.e., in parallel. Therefore, to avoid confusion, intermediate paths are defined as follows:

> An intermediate path is a single path which starts from the source vertex and ends at the sink vertex.

Then, a possible path can be represented by a "sequence" of intermediate paths. Figure 2 shows a tree which displays all possible paths as sequences of intermediate paths. Notice that even though intermediate paths are shown in sequence, this does not mean that they occur one after another, but instead, they may be simultaneous. The tree representation shows all possible paths explicitly.

Table 3. All Subpaths and Intermediate Paths of System A

Subpaths	Index of Vertex
P(10,1)	10, 9, 7
P(10,2)	10, 8, 7
P(7,1)	7, 4, 1
P(7,2)	7, 6, 5
P(5,1)	5, 4, 1
P(5,2)	5, 3, 1
P(5,3)	5, 2, 1

Intermediate Paths	Index of Vertex
P(1)	10, 9, 7, 4, 1
P(2)	10, 8, 7, 4, 1
P(3)	10, 9, 7, 6, 5, 4, 1
P(4)	10, 9, 8, 6, 5, 4, 1
P(5)	10, 9, 7, 6, 5, 2, 1
P(6)	10, 9, 7, 6, 5, 3, 1
P(7)	10, 8, 7, 6, 5, 3, 1
P(8)	10, 8, 7, 6, 5, 2, 1

2.3 COMPUTATION OF TIME DELAY FOR ALL PATHS

To compute the delay for each path, the only calculation needed is to add the delays associated with the vertices which constitute a path. The algorithm for the computation of delays is:

(a) Let n be the number of transitions in the j-th subpath ending at vertex i, P(i,j). Assume that a transition has a delay of t_k. Then the delay of P(i,j) is

$$D_{ij} = \sum_{k=1}^{n} t_k \qquad (1)$$

(b) Let D_{i1}, D_{i2}, D_{i3} be time delays associated with three subpaths P(i,1), P(i,2), P(i,3). Then the delay of all subpaths with the end vertex V_i is

$$D(i) = \max(D_{i1}, D_{i2}, D_{i3}) \qquad (2)$$

For instance, consider the example in Table 3. There are four distinct subpaths ending at V_7:

P(7,1) = 7,6,5,2,1

P(7,2) = 7,4,1

P(7,3) = 7,6,5,3,1

P(7,4) = 7,6,5,4,1

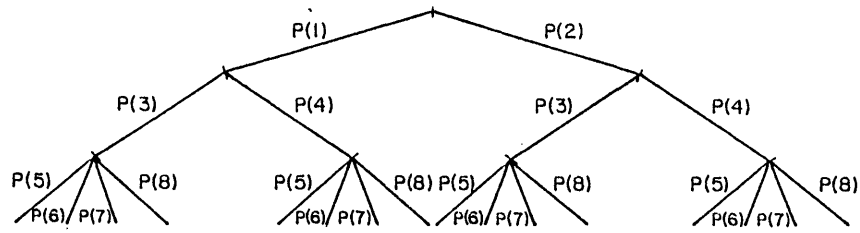


Figure 2. A Tree Showing All Possible Paths of System

The associated delays are D_{71}, = 5, D_{72} = 3, D_{73} = 5, D_{74} = 5. Then the delay from V_1 to V_7 is

$$D(7) = \max(D_{71}, D_{72}, D_{73}, D_{74}) = 5 \quad (3)$$

For an intermediate path containing n subpaths, the delay is the sum of delays associated with the n subpaths. For example, in Table 3, intermediate path P(1) contains subpaths P(10,1) and P(7,1). Then, the delay of P(1) is D(1)= 3 + 5 - 1 = 7 where 3 is the delay in subpath P(10,1); 5 is the delay in subpath P(7,1) which is calculated above; and 1 is subtracted because V_7 is counted twice.

For a possible path, because all the intermediate paths are simultaneous, the time delay is the maximal delay of the intermediate paths. For example, possible path 1 in Fig. 2 consists of intermediate path P(1), P(3), P(5). The maximal delay is 7, so the delay associated with path 1 is 7.

After all delays are computed, a shortest path with the minimal time delay and a longest path with maximal delay can be found. For analyzing overall system performance, it may be desired to compute the expected delay of the system.

2.4 EXPECTED DELAY OF A SYSTEM

To calculate expected delay, probabilities associated with each path need to be calculted first. Then the expected delay of a system can be computed.

Usually, for a system model, probabilities are given as conditional probabilities associated with each transition. If transition V_i has only one input from the previous transition, V_j, then the conditional probability $p(V_i/V_j)$ is 1. If V_i is a transition of a decision switch, a conditional probability $p(V_i/V_j) \leq 1$ will be assigned. For a n-way switch, the sum of n conditional probabilities should be equal to 1, that is

$$\sum_{i=1}^{n} p(V_i/V_j) = 1 \quad (4)$$

The probability that information processing will follow a certain intermediate path X with n transitions is given by

$$p(X) = p(V_1) \prod_{i=1}^{m-1} p_i(Z/V) \quad (5)$$

where X is the path number; Z is a transition which is on the path and V is the transition preceding Z. Table 4 shows the conditional probability matrix P_s for the example of section 2.1. Table 5 shows the probabilities associated with intermediate paths of System A.

Table 4. Probability Matrix P_s of 2 DM System

$$P_s = \begin{bmatrix} 0.0 & 0.6 & 0.4 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

The expected delay can be calculated by the following equation:

$$E_r = \sum_{i=1}^{r} p_i D(i) \quad (6)$$

where p_i and D(i) are the probability and time delay associated to the i-th possible path; r is the total number of possible paths in a system.

Table 5. Conditional Probabilities Associated with Each Intermediate Path of 2 DM System

Intermediate Path	Conditional Probability
P(1)	1 * 1 * 0.7 * 1 = 0.7
P(2)	1 * 1 * 0.3 * 1 = 0.3
P(3)	1 * 1 * 1 * 1 * 0.7 * 1 = 0.7
P(4)	1 * 1 * 1 * 1 * 0.3 * 1 = 0.3
P(5)	0.6 * 1 * 1 * 1 * 0.7 * 1 = 0.42
P(6)	0.4 * 1 * 1 * 1 * 0.7 * 1 = 0.28
P(7)	0.4 * 1 * 1 * 1 * 0.3 * 1 = 0.12
P(8)	0.6 * 1 * 1 * 1 * 0.3 * 1 = 0.18

For example, in system A, there are 16 possible paths (Figure 2). Path 1 has delay of 7 (Section 2.4)

and probability of 0.206. Then the first term of E_n is 0.206 · 7. In this particular example, because the delay, D, of all possible paths is 7,

$$E_{16} = D \cdot \sum_{i=1}^{16} p_i = 7. \qquad (7)$$

3. APPLICATION

The delays in two organizations, each one consisting of three decisionmakers, will be determined using the algorithm described in Section 2. The application is an abstracted and very simplified version of an air defense problem. In the parallel organization (Fig. 3), the airspace has been divided into three sectors, with each decisionmaker assigned to one sector. Each DM can observe and engage threats in his sector. However, threats may move between sectors; therefore, there is need for communication -- information sharing -- between decisionmakers with adjacent sectors. In the hierarchical organization (Fig. 4), the airspace is divided into two sectors, with each one assigned to a single DM.

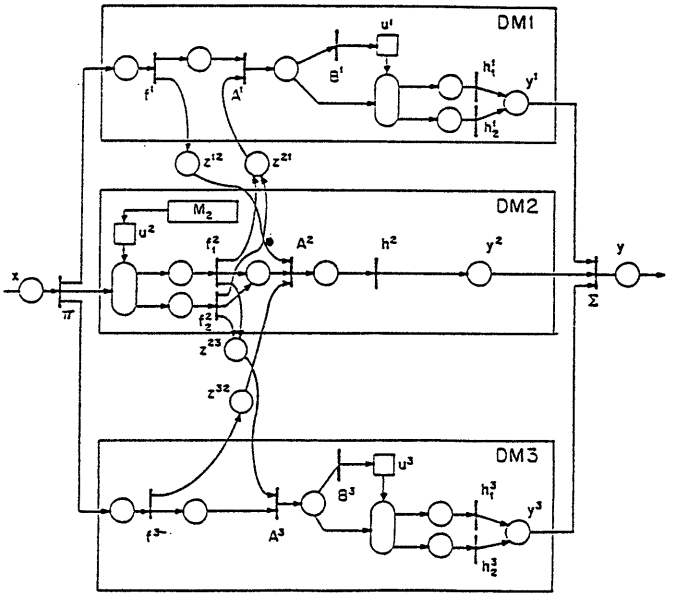


Figure 3. A Parallel Organization

Since, the workload will be high for each DM, a central region is defined that stradles the two sectors. A supervisor is introduced who does not observe the airspace directly, but receives information about threats in the central region from the two DMs. He then processes the data and allocates threats in the central region (command inputs) to either one of the DMs depending on the trajectory of the threat.

Parallel Organization

Using the algorithm interconnection matrix, allintermediate paths, the conditional probability andtime delay associated with each intermediate path, and the expected delay of overall system are computed. The interconnection matrix and labels of transitions or vertices are shown in Table 6.

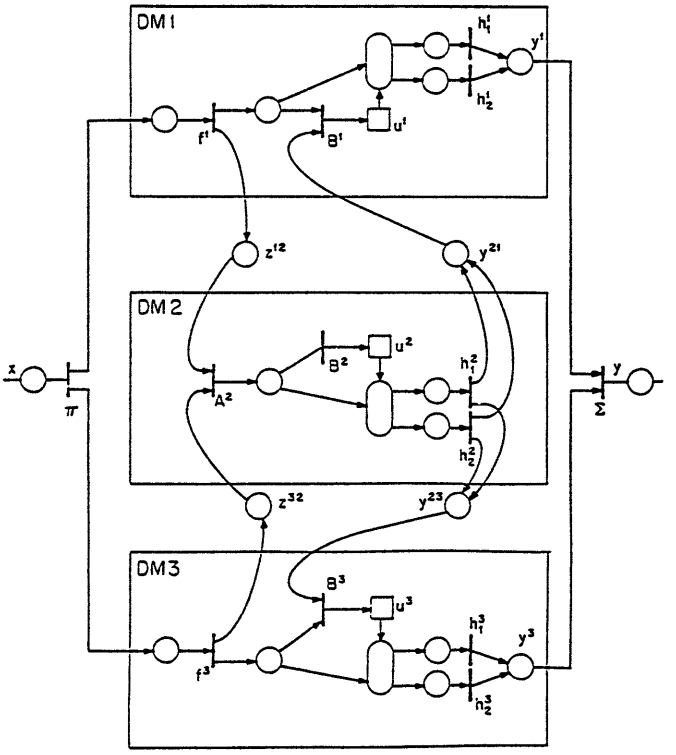


Figure 4. A Hierarchical Organization

In the results, Table 7, intermediate paths are indicated by the sequence of indices of the vertices which represent transitions. The symbol "SW" denotes that the following vertex (transition) belongs to a decision switch. The symbol "/ /" indicates that the following subpath is parallel to other subpath(s).

Table 7, shows that there are sixteen intermediate paths. Some of these paths are in parallel. For example, P(1) is parallel to both P(2) and P(3): they have the same subpath after V_7. Only decision switches create different paths. Because some intermediate paths occur simultaneously, that is, are parallel, consequently, the resulting tree has 128 possible paths.

A delay of unity in each transition is assumed during the computation. The expected delay of this system is 6 units.

Hierarchical Organization

The interconnection matrix and the transition labels for the hierarchical organization are shown in Table 8. All intermediate paths, their conditional probabilities and their delays are shown in Table 9. There are twenty intermediate paths. Each path has a delay of eight units, because for a set of parallel paths the delay of the set is the maximum path delay.

The resulting tree has 64 different possible paths, and the expected delay is 8 units.

Table 6. Interconnection Matrix for Parallel Organization

INTERCONNECTION MATRIX : COLUMN=TRANSITION ROW=LINK

-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	-1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0
0	0	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0
0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	-1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1

Table 7. Paths for Parallel Organization

INTERMEDIATE PATH	CONDITIONAL PROBABILITY	DELAY
P(1)= 10<- SW-0 9<-0 8<-0 7<-//<-0 2<-0 1	0.300	6
P(2)= 10<- SW-0 9<-0 8<-0 7<- SW-0 3<-0 1	0.120	6
P(3)= 10<- SW-0 9<-0 8<-0 7<- SW-0 4<-0 1	0.180	6
P(4)= 10<- SW- 11<-0 8<-0 7<-//<-0 2<-0 1	0.700	6
P(5)= 10<- SW- 11<-0 8<-0 7<- SW-0 3<-0 1	0.280	6
P(6)= 10<- SW- 11<-0 8<-0 7<- SW-0 4<-0 1	0.420	6
P(7)= 10<-//<- 13<-0 6<-//<-0 2<-0 1	1.000	5
P(8)= 10<-//<- 13<-0 6<- SW-0 4<-0 1	0.600	5
P(9)= 10<-//<- 13<-0 6<- SW-0 3<-0 1	0.400	5
P(10)= 10<-//<- 13<-0 6<-//<-0 5<-0 1	1.000	5
P(11)= 10<- SW- 15<- 14<- 12<- SW-0 3<-0 1	0.200	6
P(12)= 10<- SW- 15<- 14<- 12<- SW-0 4<-0 1	0.300	6
P(13)= 10<- SW- 15<- 14<- 12<-//<-0 5<-0 1	0.500	6
P(14)= 10<- SW- 16<- 14<- 12<- SW-0 3<-0 1	0.200	6
P(15)= 10<- SW- 16<- 14<- 12<- SW-0 4<-0 1	0.300	6
P(16)= 10<- SW- 16<- 14<- 12<-//<-0 5<-0 1	0.500	6

EXPECTED DELAY OF THE SYSTEM IS 6

Table 8. Interconnection Matrix for Hierarchical Organization

INTERCONNECTION MATRIX : COLUMN=TRANSITION ROW=LINK

-1	1	0	0	0	0	0	0	0	0	0	0	0	0
-1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	1	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	1	0	0	0	0	0	0	0
0	0	0	0	-1	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	-1	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	-1	1	0	0	0	0
0	0	0	0	1	0	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	-1	1	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	1	-1	0	0
0	0	-1	1	0	0	0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	-1	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	-1	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	-1

Table 9. Paths for Hierarchical Organization

INTERMEDIATE PATH	CONDITIONAL PROBABILITY	DELAY
P(1)=0 7<- SW-0 6<-0 5<-//<-0 2<-0 1	0.400	8
P(2)=0 7<- SW-0 6<-0 5<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.280	8
P(3)=0 7<- SW-0 6<-0 5<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.280	8
P(4)=0 7<- SW-0 6<-0 5<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.120	8
P(5)=0 7<- SW-0 6<-0 5<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.120	8
P(6)=0 7<- SW-0 8<-0 5<-//<-0 2<-0 1	0.600	8
P(7)=0 7<- SW-0 8<-0 5<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.420	8
P(8)=0 7<- SW-0 8<-0 5<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.420	8
P(9)=0 7<- SW-0 8<-0 5<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.180	8
P(10)=0 7<- SW-0 8<-0 5<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.180	8
P(11)=0 7<- SW- 13<- 11<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.350	8
P(12)=0 7<- SW- 13<- 11<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.350	8
P(13)=0 7<- SW- 13<- 11<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.150	8
P(14)=0 7<- SW- 13<- 11<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.150	8
P(15)=0 7<- SW- 13<- 11<-//<-0 3<-0 1	0.500	8
P(16)=0 7<- SW- 14<- 11<- SW- 10<-0 9<-0 4<-//<-0 2<-0 1	0.350	8
P(17)=0 7<- SW- 14<- 11<- SW- 10<-0 9<-0 4<-//<-0 3<-0 1	0.350	8
P(18)=0 7<- SW- 14<- 11<- SW- 12<-0 9<-0 4<-//<-0 2<-0 1	0.150	8
P(19)=0 7<- SW- 14<- 11<- SW- 12<-0 9<-0 4<-//<-0 3<-0 1	0.150	8
P(20)=0 7<- SW- 14<- 11<-//<-0 3<-0 1	0.500	8

EXPECTED DELAY OF THE SYSTEM IS 8

4. CONCLUSIONS

An algorithm has been developed for computing time delays in DDM systems. From the system matrix, an interconnection matrix is created, which consolidates all the information about connections between the system's components. Scanning of the interconnection matrix results in a set of intermediate paths. Then, all possible paths can be constructed by concatenating intermediate paths. The time delay associated which each intermediate path is calculated by summing the delays of the transitions contained in that path. A possible path is composed of several intermediate paths which are active simultaneously. Then, the time delay associated with a possible path is the maximal delay of the intermediate paths contained in this possible path. After all possible paths and associated delays are found, the expected delay for the overall decisionmaking system can be calculated. The expected delay provides an indication of the speed of response of the system. The algorithm has been applied to compute the delays in parallel and hierarchical organizations.

5. REFERENCES

Tabak, D., and A. H. Levis, "Petri Net Representation of Decision Models," to appear IEEE Trans. on Systems Man and Cybernetics, November 1985.

Cothier, P. H., and A. H. Levis, "Assessment of Timeliness in Command and Control," LIDS-P-1454, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, April 1985.

Jin, V. Y-Y, "Delays for Distributed Decisionmaking Organizations," MS Thesis, LIDS-TH-1459, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, May 1985.

OPTIMAL TASK ALLOCATION FOR A TEAM OF TWO DECISION MAKERS WITH THREE CLASSES OF IMPATIENT TASKS

Zhen-Jiang Wu, Peter B. Luh, Shi-Chung Chang
Dept. of Elec. Eng. & Comp. Sci., Univ. of Connecticut
Box U-157, Storrs, CT 06268

David A. Castanon
Alphatech Inc., 111 Middlesex Turnpike
Burlington, MA 01803

ABSTRACT

Motivated by decision making problems in scenarios such as Naval Battle Force/Battle Group operations and alike, we study task selection/allocation for a team of two decision makers. The two decision makers are modeled as two service stations with overlapping processing responsibilities, different processing rates, and common information. By using stochastic dynamic programming, a functional equation for the optimal, state-dependent priority assignment policy is derived. Properties of the optimal cost-to-go fuctions and the optimal policy are established through a set of inductive proofs. It is shown that the optimal policy is governed by two switching surfaces in the three dimensional state space. For the infinite time horizon case, the optimal policy turns out to be stationary, and numerical studies are done by using the over-relaxed Gauss-Seidal method. The switching surfaces are obtained for several sets of system parameters. These numerical results support analytical findings and also provide further insights to the problem.

1. INTRODUCTION

Computer monitoring and decision aiding with human supervision has been the trend for systems automation. An important class of decision making problems encountered involves the assignment of priorities among tasks that arrive dynamically, with random processing time requirements, random deadlines and different rewards. There can be heterogeneous decision makers (computers and/or human operators) in the system, each having different capability, expertise, imformation, responsibility, and/or objectives. As an example, consider the Naval Battle Force/Battle Group (BF/BG) operations which are directed by a team of commanders located at geographically separated platforms. The commanders could have different areas and levels of expertise. Each may have access to a portion of the information because of limited surveilance and communications. For potential threats such as air, surface or subsurface targets, imformation has to be collected and prosecution must be done before those targets penetrate the defense. The problem is then how these commanders as a team should divide and sequence the randomly arriving tasks to achieve the team objective (survivability, rewards, etc.). Problems of this nature also arise in production scheduling of manufacturing systems, air traffic control, etc.

In the man-machine study literature, there have been a few articles using queueing models to explore the priority assignment problems, e.g., (ROU80), (WAL78), (CAR66). Their emphases were on the modeling and analysis of factors in human decision making. Since analytical solutions were not available for the systems considered, most of the results were obtained through simulation.

On the other hand, optimal queueing control approach has been used by some researchers to tackle generic problems in simpler setups. (HAR75) and (PAT81) worked on cases involving single decision maker and multiple classes of tasks. (ROS82), (LIN84) and (HAJ84) dealt with various cases on two service stations and single class of tasks. Markov or semi-Markov model were established and stochastic dynamic programming was commonly used to find closed-loop optimal policy.

The paper (PAT81) started with a general formulation and tried to incorporate balking and reneging into the model for handling the random deadline issue. Difficulty in solving it was disscussed. The authors then reduced their scope to a problem with Markovian assumptions, and developed a solution methodology including a recursive algorithm and approximations under heavy traffic environment. (HAJ84) presented a Markov network problem with two service stations and linear costs. The two service stations have quite general interacting structure in between. Decisions in the problem are task routing and service priorities. The author considered both finite horizon and long run average cost cases, and showed the existence of switching curves for the optimal control policy in the two-dimensional state space.

Motivated by decision making problems in scenarios such as Naval BF/BG operations (KLE84) and alike, we in this paper extend (HAJ84)'s queueing network model to study multi-class tasks selection /allocation among a team of two decision makers. The two decision makers are modeled as two service stations with different processing capabilities (responsibility), different processing rates (expertise), and common information. Our purpose is not to model and solve a realistic military command, control and communication (C^3) problem. Rather, the study is an exploration into multi-person decision making issues in a dynamic, discrete event environment. It serves as part of the effort in developing a normative-descriptive model for the Distributed Dynamic Decision

Making experimental paradiagm as described in (KLE84).

The network considered is Markovian with three classes of tasks and linear cost. By using stochastic dynammic programming and inductive proofs, we show that the optimal policy is governed by two switching surfaces in the three dimensional state space. Both finite horizon and infinite horizon cases are considerd. For the infinite horizon case, the optimal policy is shown to be stationary, and numerical studies are made by using the over-relaxed Gauss-Seidel method. Optimal cost-to-go functions and switching surfaces are obtained numerically and shown pictorially. Variations of the optimal policy as a function of a few system parameters also provide us with insights to the problem.

The organization of this paper is as follows. The two-server queueing problem is formulated in section 2, and from it a semi-Markov decision model is developed. By applying stochastic dynamic programming, optimality conditions for the control policy are obtained. In section 3, we use mathematical induction procedure to establish properties of the optimal policy. The core of the proofs are on proving properties of the optimal cost-to-go functions. We also consider the effects of having Poisson reneging of tasks, which models sudden leaving or penetration of tasks. Section 4 discusses numerical testings and results. Finally, concluding remarks are given in section 5.

2. PROBLEM FORMULATION

Consider the queueing network model shown in Fig. 2.1. Three classes of tasks A, B, and C arrive in Poisson streams with rates λ_a, λ_b and λ_c, respectively. The two servers (DM1 and DM2) have exponentially distributed service rates. DM1 is capable of processing A and C classes of tasks with mean rates μ_{1a} and μ_{1c}, and DM2 is capable of processing classes B and C at mean rates μ_{2b} and μ_{2c}. Classes A and B tasks leave the system after being processed. Class C tasks are the so called "unknown tasks". A class C task requires "identification" by either DM1 or DM2 to find out whether it belongs to class A or class B, or it is a task that needs no futher processing. If it turns out to be a class A (B) task, it is sent to queue A (B) for further processing. If it needs no further processing, it is sent out of the system. Let P_{1a} (P_{1b}) be the probability that a class C task is identified to be of class A (B) by DM1. p_{2a} and P_{2b} are similarly defined for DM2. We assume that

$$\mu_{1c} > \mu_{1a} \text{ and } \mu_{2c} > \mu_{2b},$$

i.e., identification is faster than processing.

Denote $x = (x_a, x_b, x_c) \in Z_+^3$ as the state of the system, where x_a, x_b, and x_c are, respectively, the number of class A, B and C tasks in the system. Both DMs have complete information about the current state x and have state-dependent controls over the system. The controls are u= (e, f), where e: $Z_+^3 \to [0, 1]$ is the probability that DM1 selects the next task to be processed from queue A. (1-e) is therefore the probability of selecting a class C task. The mapping f : $Z_+^3 \to [0, 1]$ is similarly defined for DM2. Following the modeling procedure of (HAJ84), we shall in the sequal set up a semi-Markov decision model for this problem.

In the formulation, state transitions include arrivals, potential departures and potential trasfers between queues. For i= a, b or c, let A_i and D_i denote, respectively, an arrival and a potential departure of a class i task, and R_{ca} (R_{cb}) a potential transfer of a class C task to class A (to class B). For example,

$$A_a x = (x_a+1, x_b, x_c),$$

$$D_a x = ((x_a-1)^+, x_b, x_c),$$

$$R_{ca} x = (x_a+1, x_b, (x_c-1)^+),$$

where $(x_i-1)^+ = x_i-1$ if $x_i > 1$, and 0 otherwise.

The actual total event rate is

$$r' \triangleq \lambda_a + \lambda_b + \lambda_c + e\,\mu_{1a} + (1-e)\,\mu_{1c} + f\,\mu_{2b} + (1-f)\,\mu_{2c}, \tag{2.1}$$

which is state-dependent since e and f are functons of the state. To bypass the difficulty of state dependence, we utilize the "pseudo epoch" concept of (LIP75) and define a constant total event rate r as follows:

$$r = \lambda_a + \lambda_b + \lambda_c + \mu_{1c} + \mu_{2c} > r'$$

For a given u= (e, f), the transition probability function P(./.,u) is defined on $Z_+^3 \times Z_+^3$ by

$$\begin{aligned} P(y/x,u) = r^{-1}\{ & \lambda_a I(y=A_a x) + \lambda_b I(y=A_b x) \\ & + \lambda_c I(y=A_c x) + e\,\mu_{1a} I(y=D_a x) \\ & + f\,\mu_{2b} I(y=D_b x) + [(1-e)(1-P_{1a}-P_{1b})\mu_{1c} \\ & + (1-f)(1-P_{2a}-P_{2b})\mu_{2c}] I(y=D_c x) \\ & + [(1-e)P_{1a}\mu_{1c} + (1-f)P_{2a}\mu_{2c}]\, I(y=R_{ca} x) \\ & + [(1-e)P_{1b}\mu_{1c} + (1-f)P_{2b}\mu_{2c}]\, I(y=R_{cb} x)\} \\ & + (1- r'/r)\, I(y=x) \end{aligned} \tag{2.2}$$

Since controls are state dependent and are made at instants of event epochs, we denote U the sequence $(u_0, u_1, \ldots)$ where u_k is the control at the k-th transition. Given U and an initial state i_0 in Z_+^3, a semi-Markov decision process (x(t), t ≥ 0) with jump rate r and imbedded transition probability P is defined as

$$x(t) = x_{n(t)},$$

where $X = (x_0, x_1, \ldots)$ is a sequence of random variables with $P(x_0 = i_0) = 1$ and

$$P(x_{k+1} = j / x_k = i_k, \ldots x_0 = i_0)$$

$$= P(j/\, i_k,\, u_k(i_0,\ldots i_k)), \tag{2.3}$$

and $(n(t):\ t>0)$ is a rate r Poisson process independent of x.

The instantaneous cost of the system is a linear function of the number of customers in the system, i.e., $C'x = c_a x_a + c_b x_b + c_c x_c$, where c_i, $i = a, b$ and c are nonnegative constants. Let τ_n be the random time at which nth event happens. Then the cost for a control policy U over $[0, \tau_n)$ is

$$E_x^U \int_0^{\tau_n} e^{-\alpha t} C'x(t)dt$$

where α is a nonegative discount constant, and E_x^U denotes the expectation with respect to $(x(t))$. Following (ROS82), this cost can be shown to be equal to

$$(\alpha + r)^{-1} E_x^U \sum_{K=0}^{n-1} \beta^K C'x_k, \tag{2.4}$$

where $\beta = r/(r+\alpha) < 1$. In view of (2.3), (2.4) can be interpreted as the cost over n time steps for a discrete time decision process with discount factor β. Ignoring the constant factor $(\alpha + r)^{-1}$, let us define the n-stage optimal cost-to-go for a given initial state x as

$$V_n^\beta(x) = \min_U E_x^U \sum_{K=0}^{n-1} \beta^K C'x_k, \quad n \leq \infty, \text{ and}$$

$$V_0^\beta(x) = 0 \text{ by convention.}$$

Following (HAJ84) and (SCH75) it can be shown that

$$\lim_{n\to\infty} V_n^\beta(x) = V_\infty^\beta(x).$$

Then the dynamic programming equation leads to the following optimality conditions:

$$\begin{aligned} V_{n+1}^\beta(x) = C'x + \beta \min_{e,f} \{ & \lambda_a V_n^\beta(A_a x) \\ & + \lambda_b V_n^\beta(A_b x) + \lambda_c V_n^\beta(A_c x) \\ & + eE_1(x) + (1-e)E_2(x) \\ & + fF_1(x) + (1-f)F_2(x)\} \triangleq TV_n^\beta(x) \end{aligned} \tag{2.5}$$

where T denotes the dynamic programming operator, and

$$\begin{aligned} E_1(x) &\triangleq \mu_{1a} V_n^\beta(D_a x) + (\mu_{1c} - \mu_{1a}) V_n^\beta(x), \\ E_2(x) &\triangleq \mu_{1c}[(1-P_{1a}-P_{1b})V_n^\beta(D_c x) + P_{1a}V_n^\beta(R_{ca}x) \\ &\quad + P_{1b}V_n^\beta(R_{cb}x)], \\ F_1(x) &\triangleq \mu_{2b} V_n^\beta(D_b x) + (\mu_{2c} - \mu_{2b}) V_n^\beta(x), \text{ and} \\ F_2(x) &\triangleq \mu_{2c}[(1-P_{2a}-P_{2b})V_n^\beta(D_c x) + P_{2a}V_n^\beta(R_{ca}x) \\ &\quad + P_{2b}V_n^\beta(R_{cb}x)]. \end{aligned} \tag{2.6}$$

The optimal control $u = (e, f) \in [0, 1]^2$ for the current state x with n steps to go is therefore determined by

$$e = \begin{cases} 1 & \text{if } E_1(x) \leq E_2(x), \\ 0 & \text{if } E_1(x) > E_2(x), \text{ and} \end{cases}$$

$$f = \begin{cases} 1 & \text{if } F_1(x) \leq F_2(x), \\ 0 & \text{if } F_1(x) > F_2(x), \end{cases} \tag{2.7}$$

This is a bang-bang type of control.

3. STRUCTURE AND PROPERTIES OF OPTIMAL CONTROL POLICY

In this section, the inductive approach of (HAJ84) is extended to show that the optimal control policy is of switching type. The derivation, however, are more complicated than that of (HAJ84) because of the existence of multiple classes of tasks and the specific routing pattern considered. We also show that for the infinite horizon case with $\beta < 1$, the policy is stationary.

The core of the proof is to establish by induction the following properties for $V_n^\beta(\cdot)$.

(P1): $V_n^\beta(x)$ is increasing in x_a, x_b, and x_c.

(P2): $V_n^\beta(x+y) - V_n^\beta(x)$ is increasing in x_a, in x_b and in x_c for each fixed y in Z_+^3.

(P3): $V_n^\beta(x) - G_{in}(x)$ is increasing in x_a, in x_b, and in x_c, where

$$\begin{aligned} G_{in}(x) \triangleq\ & (1 - P_{ia} - P_{ib}) V_n^\beta(D_c x) + \\ & P_{ia} V_n^\beta(R_{ca}x) + P_{ib} V_n^\beta(R_{cb}x), \quad i = 1, 2. \end{aligned} \tag{3.1}$$

(P4): $E_1(x) - E_2(x)$ and $F_1(x) - F_2(x)$ are increasing in x_c, and decreasing in x_a and in x_b, where $E_1(x)$, $E_2(x)$, $F_1(x)$, and $F_2(x)$ are defined by (2.5).

For n= 1, $V_1^\beta(x) = C'x = c_a x_a + c_b x_b + c_c x_c$. It can be easily checked that (P1)-(P4) are satisfied.

Assume that V_n^β satisfies (P1)-(P4) for $x \in Z_+^3$. After lengthy derivation, it can be shown that V_{n+1}^β also satisfies (P1)-(P4) (the proofs are provided in a detailed version of the paper under preparation). Thus by mathematical induction we conclude that

Theorem 1:

$V_n^\beta(x)$ satisfies (P1)-(P4) for $\forall\, n \in N$, $x \in Z_+^3$ and $\beta < 1$.

After exploring the properties of the optimal cost-to-go function $V_n^\beta(X)$, we now show the existence of switching surfaces for e and f.

Define two switching functions

$$s_{1n}(x_b, x_c) = \min\{x_a: E_1(x) - E_2(x) \leq 0\} \text{ and}$$

$$s_{2n}(x_a, x_c) = \min\{x_b: F_1(x) - F_2(x) \leq 0\},$$

and their associated regions

$$S_{1n} = \{x \in Z_+^3: x_a > s_{1n}(x_b, x_c)\},$$

$$S_{2n} = \{x \in Z_+^3: x_b > s_{2n}(x_a, x_c)\}.$$

Theorem 2:

The switching functions s_{1n} and s_{2n} define two switching surfaces in the state space x. When there are n steps to go, the optimal decision is given by

$e_n=1 \iff x \in S_{1n}$, and

$f_n=1 \iff x \in S_{2n}$.

Proof:

This assertion is a consequence of (2.6) and (P4).

For the infinite horizon case ($n=+\infty$), by following Theorem 1 of (LIP73), we have

Theorem 3:

$$V^{\beta}(x)=\lim_{n\to\infty} \min_{U} \sum_{k=0}^{n} \beta^k C'x_k, \quad \beta < 1,$$

is achieved by a stationary policy.

For a system with reneging tasks, the analysis is similar. Suppose the reneging is a poisson process with mean reneging rates α_i, i= a, b, and c. With state x, the system reneging rate is $\alpha_a x_a + \alpha_b x_b + \alpha_c x_c$. Suppose that under normal operation, the system reneging rate is bounded above by a constant M. Then the total event rate can be redefined as

$$r= \lambda_a + \lambda_b + \lambda_c + \mu_{1c} + \mu_{2c} + M.$$

The transition probability function (2.2) can be modified accordingly. Let the linear instantaneous reneging cost be

$$c_{ra}\alpha_a x_a + c_{rb}\alpha_b x_b + c_{rc}\alpha_c x_c.$$

Then the dynamic programming equation becomes

$$\begin{aligned} V^{\beta}_{n+1}(x)= \bar{C}'x &+ \beta r^{-1} \min_{e,f} \{\lambda_a V^{\beta}_n(A_a x) \\ &+ \lambda_b V^{\beta}_n(A_b x) + \lambda_c V^{\beta}_n(A_c x) + \alpha_a x_a V^{\beta}_n(D_a x) \\ &+ \alpha_b x_b V^{\beta}_n(D_b x) + \alpha_c x_c V^{\beta}_n(D_c x) \\ &+ (M- (\alpha_a x_a + \alpha_b x_b + \alpha_c x_c)) V^{\beta}_n(x) \\ &+ eE_1(x) + (1-e)E_2(x) \\ &+ fF_1(x) + (1-f)F_2(x)\} \triangleq TV^{\beta}_n(x), \end{aligned}$$

where $\bar{C}'x \triangleq (c_a + c_{ra}\alpha_a)x_a + (c_b + c_{rb}\alpha_b)x_b + (c_c + c_{rc}\alpha_c)x_c$, and E_1, E_2, F_1, F_2 and the optimality conditions remain the same as in (2.6) and (2.7). Consequently all the previous results apply.

4. NUMERICAL RESULTS

In this section, infinite horizon problems are considered for numerical study. Though the previous results are developed under the assumption of infinite queue size, finite queue size for each class of tasks is assumed for this numerical study. Let N be the size of each queue. If N is large, the finite queue size model provides a good approximation to the original infinite queue size problem, except for regions close to the queue capacity. An iterative algorithm based on the over-relaxed Gauss-Seidel method (PIZ75) is developed to solve the optimal cost-to-go equation. With the optimal cost-to-go determined for each state, analytic results such as (P1)-(P4) are checked and switching surfaces are obtained. We also performed studies on the optimal control policy as a function of a few system parameters such as cost coefficients, arrival/service rates, class C task feedback probabilities, discount factor, etc.

For an infinite horizon problem, the dynamic programming equation (2.5) can be written as

$$V^{\beta}(i,j,k)= C(i,j,k)+ \beta r^{-1} PV(i,j,k), \quad 0\leq i,j,k \leq N, \qquad (4.1)$$

where $i= x_a$, $j= x_b$, $k= x_c$, $C(i,j,k)$ is the instantaneous cost at state (i,j,k), PV is the expected optimal cost-to-go after one state transition from (i,j,k), and N is the queue size for each class. An over-relaxed Gauss-Seidel method is adopted to solve (4.1). The solution procedure is to iterate coordinate by coordinate and find the fixed point $V^{\beta*}$ of (4.1).

In the numerical study, N is set to be 15, which implies that the dimension of (4.1) is $16^3= 4096$. The over-relaxed Gauss-Seidel algorithm is implemented in FORTRAN on IBM 3081. For all the cases we tested, the algorithm converged, and the number of iterations varies from 6 to 90. Except for those states in the neighborhood of queue capacity, numerical results obtained do verify the properties established in section 3.

The switching surfaces of DM1 and DM2 are depicted in Figures 4.1 and 4.2, respectively, for a case with the following parameters:

$$c_a= c_b= c_c= 0.8,\ \lambda_a= \lambda_b = \lambda_c= 1,$$

$$\mu_{1a}= 4,\quad \mu_{1c}= 6,\quad \mu_{2b}=5,\quad \mu_{2c}=7,$$

$$P_{1a}= P_{1b}= P_{2a}= P_{2c}= 0.1 \text{ and } \beta = 0.94.$$

It is of our interest to investigate the sensitivity of the optimal policy with recpect to system parameters. In doing this, the previous example is used as a basis to carry out the comparisons. A few observations from the study are as follows.

(1). If a class C task is identified to be class A or B, it is to be routed to the proper queue for processing. The increases in class C task feedback probabilities P_{ij}, i=1, 2, j= a, b, indirectly result in increases of arrivals of classes A and B tasks. The control policy of DM1 (DM2) therefore shifts in favor of processing classes A and B tasks. Figure 4.3 shows this observation. It illustrates the variations of the optimal control e with respect to feedback probabilities. The curves are obtained with $x_b= 3$.

(2). The discount factor β implies an effective look ahead time of $(1-\beta)^{-1}$ stages.

When β is small, $V^{\beta}(X)$ is dominated by the linear stage-wise cost, thus it is approximately linear in x. In this case, the constants $c_a \mu_{1a}$ and $c_c \mu_{1c}$ reflect the average gain of DM1 for processing classes A and C tasks. DM1 would therefore process the class of tasks with the larger gain until that queue is depleted. This in turn implies that switching surface for DM1 is either the $x_a=0$ plane or $x_c=1$ plane. Similar statement holds for DM2. When β is large, $V^{\beta}(x)$ becomes strictly convex. The switching surfaces are no long degenerate. Figure 4.4 demonstrates the variations of e for several values of β .

(3).Variations of e for several values of arrival rates are shown in Figure 4.5. The insensitivity of the control policy with respect to arrival rate variations for $\lambda > 3$ is due to the finite population assumption.

5. CONCLUDING REMARKS

In this paper, the problem of multi-class task selection/allocation for a team of two decision makers was modelled in the framework of priority queueing network. Centralized information structure was assumed for decision makers and the resultant control policy fell into the category of "centralized design with decentralized implementation" (FIN80). By using stochastic dynamic programming, a functional equation for the optimal, state-dependent priority assignment policy was derived. Properties of the optimal cost-to-go function and the optimal policy were established through a set of inductive proofs. An iterative algorithm was developed for the numerical computation of the optimal function equation. Numerical results from testing on finite population problems supported our analytic findings and also provided us with further insights to the problem.

Two shortcomings of the present model are in the objective function and the handling of task reneging. The linear instantaneous cost associated with system population gave us analytic convenience but it might not be realistic for practical problems. Important performance measures such as timeliness and survivability are not well captured. The reneging of tasks is used to model sudden leaving or penetration of tasks. The memoriless property of the Poisson reneging process maintain the manageability of the problem. For more realistic situations, once a task appears, its due time is more or less determined. Decisions will naturally be based on the time available and the time required for each task. The Poisson reneging process is not a proper model for this consideration. However, a general reneging process is known difficult to handle in the realm of queueing control. Further evaluation of potential resolution approaches are needed.

For multi-task, multi-decision maker systems, there are many other challenging aspects, especially the study of distributed decision making. In this setup, tasks arrive randomly at individual processors. The decision maker at a processor has to determine local routing policy and local priority policy. The issues of information pattern at each decision-maker, information flow among decision-makers, dynamic resource allocation, the installment of a coordinator, etc. are critical and pose various levels of complexity to the problem.

Acknowledgement: The research works reported here was supported in part by Alphatech Inc. under contract SC-000192-01. The authors appreciate valuable inputs from Professor D. L. Kleinman and Mr. D. Serfaty in formulating the problem and many many stimulating discussions.

REFERENCES

(ROU80) W. B. Rouse, System Engineering Models of Human-Machine Interaction, Series Vol.6, North Holland,1980.

(WAL78) R. S. Walden, "A Queueing Model of Pilot Decision-making in a Multitask Flight Management Situation," IEEE Trans. on System, Man and Cybernetics, Vol. SMC-8, pp. 867-875, Dec. 1978.

(CAR66) Carbonell, J. R. "A Queueing Model of Many Instrument Visual Sampling," IEEE Trans. Hum. Factors Electron., Vol. HFE-7, pp. 154-164, Dec. 1966.

(FIN80) W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, A. Wozniak, Control and Coordination in Hierarchical Systems, Wiley, 1980.

(HAR75) J. M. Harrison, "Dynamic Scheduling of a Multi-class Queue: Discount Optimality," Oper. Res., Vol.23, No.2, pp. 270-282, March/April, 1975.

(HAJ84) B. Hajek, "Optimal Control of Two Interacting Service Stations," IEEE Transactions on Automatic Control, Vol. AC-29, No.6, June 1984, pp. 491-499.

(KLE84) D. L. Kleinman, D. Serfaty, P. B. Luh, "A Research Paradigm for Multi-human Decision Making," Proceedings of the 1984 American Control Conference, Dan Diego, CA, June 1984, pp. 6-11.

(LIP73) S. A. Lippman, "Semi-Markov Decision Processes with Unbounded Rewards," Management Sci., Vol. 19, No.7, 1973, pp 717-731.

(ROS82) Z. Rosberg, P. P. Varaiya, J. C. Walrand, "Optimal Control of Service in Tandem Queues," IEEE Transactions on Automatic Control, Vol. AC-27, No.3, June 1982, pp. 600-610.

(PIZ75) S. M. Pizer, Numerical computing and Numerical Analysis, Science Research Associates, Inc., California, 1975.

(PAT81) K. R. Pattipati and D. L. Kleinman, "Priority Assignment Using Dynamic Programming for a Class of Queueing Systems," IEEE Transctions on Automatic control, Vol. AC-26, No 5, Oct., 1981, pp 1095-1106.

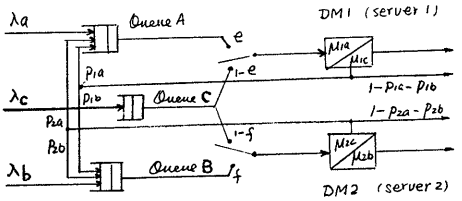


Fig. 2.1.

SWITCHING SURFACE (E)
FOR DM1

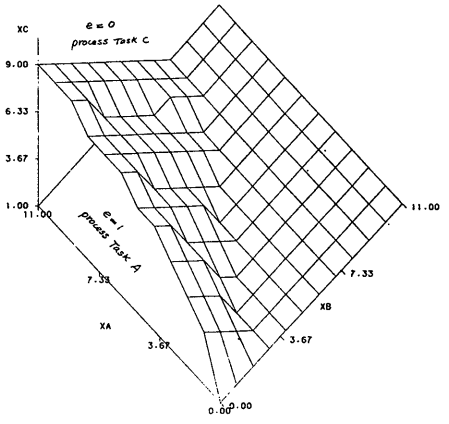


Fig. 4.1

SWITCHING SURFACE (F)
FOR DM2

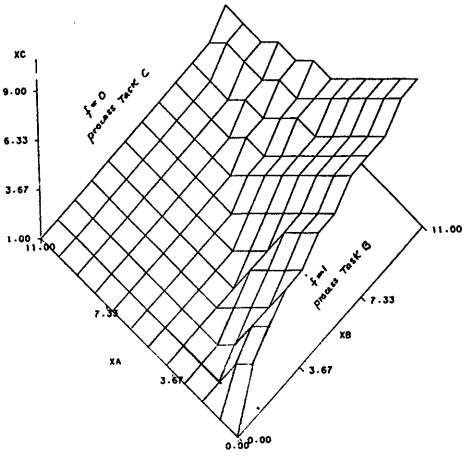


Fig. 4.2

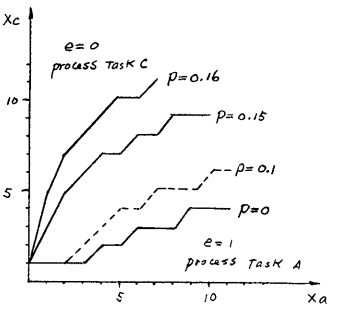


Fig. 4.3
for fixed $x_b = 3$
$p = p_{1a} = p_{1b} = p_{2a} = p_{2b}$ varies

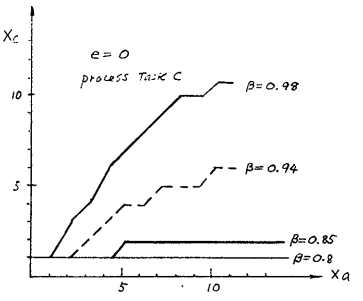


Fig 4.4
for fixed $x_b = 3$
β varies

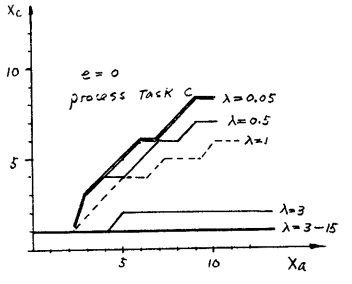


Fig. 4.5
for fixed $x_b = 3$
$\lambda = \lambda_a = \lambda_b = \lambda_c$ varies

INFORMATION STORAGE AND ACCESS IN DECISIONMAKING ORGANIZATIONS*

Ghassan J. Bejjani
Morgan Stanley, 55 Water Street, New York, NY 10041

Alexander H. Levis
Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139

ABSTRACT

Information storage and access in decisionmaking organizations is modeled using a Petri Net representation. A centralized and a decentralized database configuration are analyzed and their impacts on the decisionmakers' workload assessed. Organizational protocols are defined and their criteria of acceptability presented. Minimum allowable input interarrival time and response time are determined for two organizational structures: parallel and hierarchical. A numerical example suggests the use of timeliness as a third organizational attribute — the first two being workload and peformance. It also demonstrates the importance of updating coordination in evaluating the organization's performance.

1. INTRODUCTION

During the past decade, information theory has been applied to the analysis and evaluation of organizations. First developed by Shannon (Shannon and Weaver, 1949), information theory matured into a mathematical theory in its own right, and was applied to the study of various communications systems (Gallager, 1968). It was then used as a basic tool for modeling human decisionmaking (see Sheridan and Ferrel, 1974, and Drenick, 1975). The Partition Law of Information (Conant, 1976) provided a physical interpretation of the mathematical expressions derived by using the n-dimensional version of the theory.

A two-stage information theoretic model of the decisionmaker was introduced by Boettcher and Levis (1982). Quantitative means for measuring the human decisionmakers' workload and the organization's performance were designed under a set of restrictive assumptions. Subsequent research effort (Hall and Levis, 1984; Chyen and Levis, 1985; Tomovic and Levis, 1984) was oriented towards relaxing some of those assumptions and resolving more complex issues related to a realistic use of the decisionmaking model.

This paper addresses the issue that decisionmakers are not memoryless (an assumption in the original model) and that information storage and acccess devices are actually put to service in most modern organizations. The study of databases in acyclical organizations is approched along two directions: (a) computation of modified activity terms that represent the decisionmaker's workload and (b) consideration of time and delays in the normal functioning of an organization. The two directions are developed separately but are brought together in the illustrative example of the last section.

Figure 1 shows the Petri Net representation (Tabak and Levis, 1984) of the two-stage model of the n^{th} member of an organization. His input x^n is a component of a single vector source distributed by a set of partitioning matrices among all the decisionmakers (Stabile and Levis, 1984). The decisionmaker processes this input in the situation assessment (SA^n) stage to determine or select a particular value of the variable z^n that denotes the situation. He may communicate his assessment of the situation to other members (z^{no}) and he may receive their assessments in return (z^{on}). This supplementary information may be used to modify his assessment, i.e., it may lead to a different value of z^n denoted by $z^{n'}$. Possible alternatives of action are evaluated in the response selection (RS^n) stage. The outcome of this process is the selection of a local action or decision y^n that may be communicated to other team members or may form all or part of the organization's response. A command input from other decisionmakers, v^{on}, may affect the selection process.

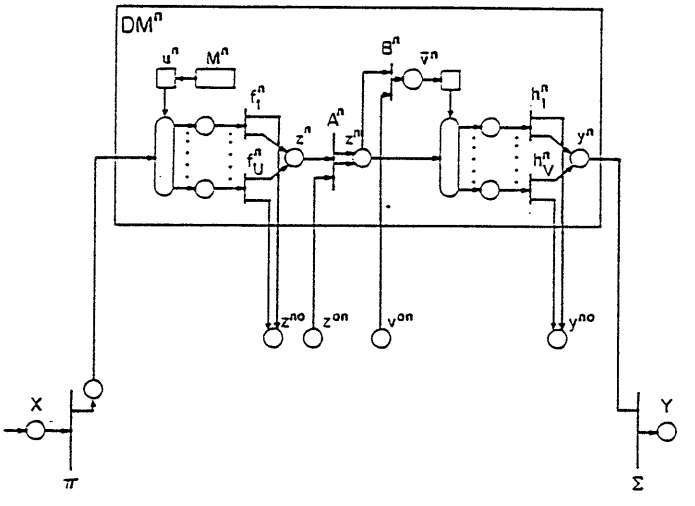


Figure 1. Petri Net Representation of the n^{th} Decisionmaker of the Organization

The situation assessment stage consists of U algorithms (f_i^n, i=1,...,U). The value taken by the variable u^n determines the algorithm to be used, and is chosen according to the probability distribution $p(u^n)$. Similarly, the choice of an algorithm in the

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided in part by the Office of Naval Research under Contracts N00014-83-K-0185 (NR 274-349) and N00014-84-K-0519 (NR 649-003) and in part by the US Army Research Institute for the Behavioral and Social Sciences under Contract No. MDA903-83-C-0196.

RS stage is determined by the variable $\bar{v}^n$, with probability distribution $p(\bar{v}^n|z^n)$.

As a response to the need for memory and information handling in today's organizations, the concept of Decision Aids first appeared a litle more than a decade ago. These devices are evolving into well-integrated Decision Support Systems (DSS) (Keen, 1981). The database is one of the three main parts of a Decision Support System. The other two are an information management program, and a machine-user interface (computer terminal) (Sprague, 1980; Sprague and Carlson, 1982). This paper will address the database and decisionmaker/machine interface issues from an information theoretic point of view. The database's storage and access procedures, and their impact on the decisionmaker's workload and performance levels, will be described.

2. THE GENERAL DATABASE MODEL

The database model developed in this paper conforms to the traditional definition of an information storage device: it can receive information from an external source, it stores it adequately, and it delivers this information, or part of it, whenever accessed by its users. The Petri Net model adopted here consists of two stages (see Fig. 2). The first stage, transition C, receives an input from the decisionmaker who requests access to the data. This input represents the situation in which the user is. Transition C determines then the nature of the information needed to cope with that situation, and sends a query to the next stage, D. Transition D performs the actual search, and delivers the data to the decisionmaker at a predetermined stage of his internal decisionmaking process.

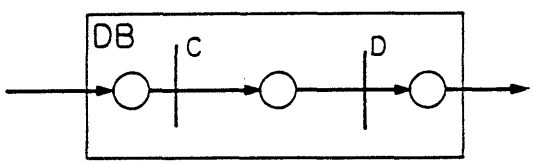


Figure 2. Petri Net Representation of the General Database Model

Databases can be used in either a centralized or a decentralized configuration. Decentralized databases are defined here as individual storage units, accessed exclusively by one decisionmaker, and holding and delivering information relevant to this decisionmaker's task only. It was proved (Bejjani, 1985) that the increase in activity due to a centralized or decentralized configuration were similar. However, there are important differences. First, the time associated with the query process is much shorter when the database is an individual one than when it is centralized. In effect, in the former case, no irrelevant information is to be scanned and then discarded, which happens in the latter case, and the system's answer to its stimuli is more timely. However, an advantage of a centralized database structure is that it allows for more convenient updating. It can be updated in one operation, providing all the decisionmakers with equally recent information, whereas decentralized databases require a much greater updating effort to obtain the same result. This paper will develop information theoretic aspects of the centralized databases and discuss the decentralized case briefly (for a comprehensive comparison of the two configurations, the reader is referred to Bejjani, 1985).

2.1 Centralized Databases

A centralized database is a database shared by all members of the organization. It is physically located in one place, and individual terminals allow the decisionmakers to access it independently. In the Petri Net representation, a centralized database is modeled as one unit, comprising several transition C/transition D sequences. There are two such databases, one for the SA stage, called DBSA, and one for the RS stage, DBRS. The inputs to transition C^n in DBSA are the inputs to the n^{th} decisionmaker, x^n, and the variable u^n indicating the SA algorithm he is about to use (see Fig. 3). Transition C^n emits then a message towards transition D^n that carried a query for the information needed for DM^n to process x^n through the selected SA algorithm. D^n in turn delivers the requested data, d^n_{SA}, to the decisionmaker, who receives it as an input to the algorithm he is using. The usage of DBRS follows a similar rationale applied to the RS stage.

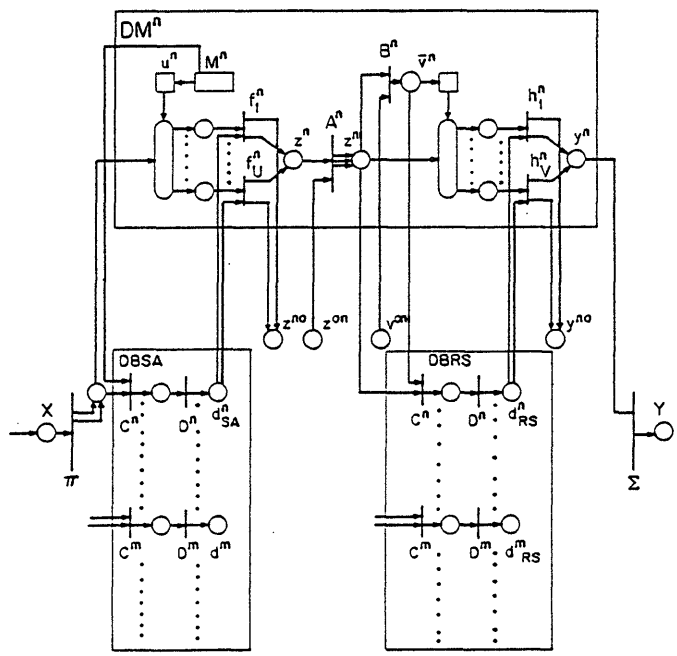


Figure 3. Petri Net Representation of DM^n Using Centralized Databases

The use of databases has a significant impact on the decisionmaker's workload, as can be seen in the following development. Activity rate terms are derived by applying the Partition Law of Information Rates (Conant, 1976) to the decisionmaking model used here. For a more complete description of the calculations, the reader is referred to Bejjani (1985). The modifications to the basic model are due to the presence of two supplementary variables, d^n_{SA} and d^n_{RS}, and to their relationship with the existing structure. For simplicity, the superscript n will be omitted in the following equations whenever confusion may not arise.

Throughput Rate:

$$F_t = \bar{T}(x, d_{SA}, z^{on}, v^{on}, d_{RS}: u, z^{no}, \bar{z}, \bar{v}, y^{no}, y) \quad (1)$$

Blockage Rate:

$$F_b = \bar{H}(x, d_{SA}, z^{on}, v^{on}, d_{RS}) - F_t \quad (2)$$

Noise Rate:

$$F_n = \bar{H}(u) + \bar{H}_{\bar{z}}(v) \quad (3)$$

Coordination Rate:

$$F_c = \sum_{i=1}^{U} \left(\bar{g}_c^i\,(p(x, d_{SA})) + \frac{\alpha_i}{\tau_{SA}} H(p_i)\right) + \bar{H}(z)$$

$$+ \bar{g}_c^A\,(p(z, z^{on})) + \bar{g}_c^B\,(p(\bar{z}, v^{on}))$$

$$+ \sum_{j=1}^{V} \left(\bar{g}_c^{U+j}\,(p(\bar{z}, d_{RS})) + \frac{U+j^{\alpha}}{\tau_{RS}} H(p_j)\right) + \bar{H}(y)$$

$$+ \bar{H}(z) + \bar{H}(\bar{z}) + \bar{H}(\bar{z}, \bar{v}) + \bar{T}\,(x, d^{on}: z^{on})$$

$$+ \bar{T}_{\bar{z}}(x, d_{SA}, z^{on}: v^{on}) + \bar{T}_{\bar{z},\bar{v}}(x, d_{SA}, z^{on}, v^{on}: d_{RS}) \quad (4)$$

where

$$p_i = p(u=i) \quad ; \quad p_j = p(v=j) \quad (5)$$

$$H(p) = p\log_2 p + (1-p)\log_2(1-p) \quad (6)$$

and α_i is the number of variables of the algorithm i that are reinitialized at each iteration. The symbol τ_{SA} designates the mean interarrival time of the input to the SA stage. τ_{RS} has an equivalent meaning with respect to the RS stage. The mean input interarrival time can be used in the equations, if the interarrival time is not constant, by regulating the source (Hall, 1982). The functions $\bar{g}_c^i$, $\bar{g}_c^{U+j}$, $\bar{g}_c^A$ and $\bar{g}_c^B$ are the individual coordination rate functions of the SA, A, B, and RS algorithms, and are of the following form:

$$\bar{g}_c^i = \sum_{j=1}^{\alpha_i} \bar{H}_u\,(w_j^i) - \bar{H}_u(W^i) \quad (7)$$

The terms $\bar{H}(z)$, $\bar{H}(\bar{z})$, $\bar{H}(\bar{z},\bar{v})$ in (4) can be interpreted to represent the direct coordination rate between subsystems, through the fact that one subsystem's output is another's input. However, indirect coordination between the subsystems is accounted for by the transmission rate terms. $\bar{T}_z(x, d_{SA}: z^{on})$ represents the coordination rate that is due to the relationship between x and d_{SA}, and z^{on}. Indeed, if the inputs to DM^n and those to the rest of the organization (RO) are related, or if d_{SA}^n and d_{SA}^m, $m \neq n$, are not totally independent, due to the structure of the storage or the updating process in the centralized database, then z^{on} can bring to S^A information about the inputs to the system that is not contained in z. Similar interpretations hold for the other two transmission rate terms. The term $\bar{T}_{\bar{z},\bar{v}}(x, z^{on}, d_{SA}, v^{on}: d_{RS})$ raises the question of the relationship between d_{SA} and d_{RS}, i.e., whether the situation assessment database (DBSA) and the response selection one (DBRS) are related.

2.2 Decentralized Databases

A decentralized database structure is shown in Figure 4. The only difference with respect to Figure 3 is the presence of only one transition C^n/transition D^n sequence per database, which models the exclusive use of each database by a single decisionmaker. Apart from that, decentralized databases are assumed to function in exactly the same manner as centralized ones.

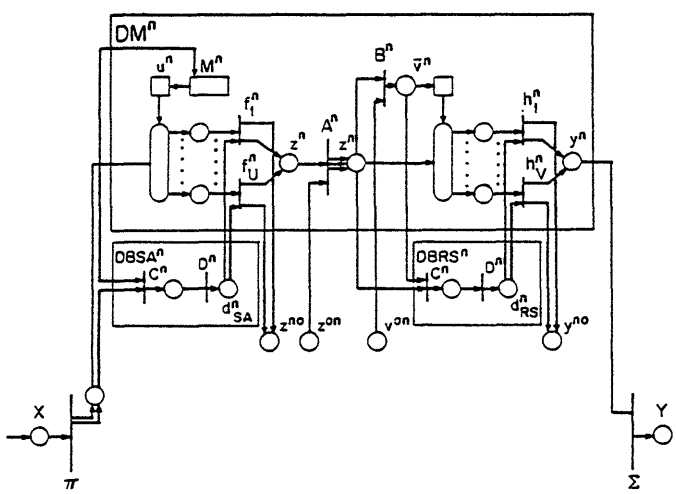


Figure 4. Petri Net Representation of DM^n Using Decentralized Databases

2.3 Fixed Databases and the Memoryless Model

The results in section 2.1 were derived assuming the data d_{SA} and d_{RS} to be variable quantities. However, it might very well be the case that d_{SA} and d_{RS} are fixed, either because the databases are never updated or because the values taken by d_{SA} and d_{RS} remain valid during a very long time, compared to the mean input interarrival time. In this simple case, the database's direct contribution to the decisionmaker's activity rate is null, and the expressions developed above become similar to those derived in the basic memoryless decisionmaker case. They are derived by simply eliminating the variables d_{SA} and d_{RS} and the input variables to the databases from the equations, which shows that the reduction from the database-equipped model to the memoryless one is consistent.

3. PROTOCOLS AND THEIR APPLICATION TO ORGANIZATIONAL STRUCTURES

3.1 Definition of Protocols and Determination of Their Key Variables

A protocol is the description of the chronological order in which elementary tasks have to be performed within one decisionmaker as well as between two or more of them. Determination of procotols is a fundamental design problem for organizations in general, and of updatable database-equipped ones in particular. Indeed, if the sequences of operations for each decisionmaker are not clearly defined, and if the updating tempo of the database does not take these sequences into account, chaos can result. In brief, the situation could arise where different decisionmakers would be accessing different databases at different times, with different levels of accuracy and relevance of the data, in order to process the same input.

Since the Petri Net representation (Tabak and Levis, 1984) clearly illustrates the organization's protocol as defined above and since a key notion in the definition of a protocol is the amount of time involved at each step of the decisionmaking process, an acceptable protocol for a given organization will consist of its Petri Net representation supplemented with the allocation of a processing time to each transition. The processing time in fact represents the maximum allowable duration of a transition for the organization to function in an orderly fashion, following its operating protocol.

Assumptions: In devising an acceptable protocol for the kind of organizations dealt with here, the following assumptions are made:

(1) - the source emits the input X with a constant interarrival time

(2) - the various transitions have constant processing times.

(3) - communication between transitions is instantaneous.

(4) - any transition can process an incoming input as soon as it has finished processing the previous one, and no sooner.

(5) - no queueing is allowed at any stage of the process.

Assumptions (1) and (2) are a corollary of the broader assumptions that the whole system operates in steady state. Assumption (3) states in fact that all the decisionmaking occurs within the transitions, and that no processing time is allocated to places. Assumption (4) is putting the "pipe-line effect" into words; this assures that the information flow through the system is continuous. Assumption (5) is a prerequisite to the application of Petri Net theory to the study of information theoretic decisionmaking models: in effect, when queueing takes place, two or more different tokens can coexist in the same place. Since transitions do not have any means of recognizing priorities in choosing one token as an input out of the same place, the queue cannot be managed, and the organization's protocol is transgressed. (For a relaxation of this assumption, see Jin, 1985).

Proposition: Under assumptions (1) to (5), two necessary conditions for an organization's protocol to be acceptable are:

- every transition in the system must have a processing time smaller than or equal to the mean input interarrival time.

- the total amount of time spent by a token in one place cannot exceed the mean input interarrival time.

Both necessary conditions provide a symmetric analytical tool. Indeed, if the processing times of the transitions in the system are fixed, then the minimum admissible input interarrival time for the organization can be determined: it is equal to the greater of two quantities: the maximum processing time present on the Petri net diagram of that organization, and the maximum time any token spends in any place. Determining this minimum interarrival time is a very useful way of comparing the effectiveness of different organizational structures in a given context.

The second necessary condition applies in cases of organizational interactions where one decisionmaker sends some information to another and cannot proceed before receiving a message back. Thus, the proposition provides a way of determining the upper limit of the response time of this other decisionmaker, everything else being fixed. This will be made clearer in the next section.

As a last comment, one should realize that the use of the proposition is not restricted to decisionmaking organizations. In fact, its arguments are relevant to any acyclical information processing structure where Assumptions (1) to (5) are satisfied.

3.2 Construction of Protocols for the Centralized Case

In this section, the proposition will be used to develop protocols for two particular organizations using a centralized database configuration. The basic quantity for each organization is τ, the processing time of any SA or RS transition. It is assumed to be identical for all such transitions in both organizations, and it will be the unit used for all quantities computed here. Furthermore, τ is assumed to be greater than the processing time of other types of transitions, on the grounds that more decisionmaking takes place in SA and RS transitions than in the others. The database's response time is assumed to be τ as well.

Parallel Organizational Structure

In a parallel organizational structure, decisionmakers are linked by somewhat symmetrical relationships: they do not formally issue commands to each other, and they can share information at all stages according to pre-established operating procedures. The parallel structure considered in this work is a three-person organization, (Fig. 5) called "Organization P" from here on. DM^1 and DM^3 use only one SA algorithm and two RS algorithms each, and DM^2 has the choice between two SA algorithms, whose output can be processed by only one RS algorithm. The command input v^{on} is absent from the model, due to the non-hierarchical structure; the decisionmakers do however share information about their situation assessments.

Organization P uses two centralized databases, DBSA and DBRS; An acceptable protocol for this organization has been derived and is given in Figure 5. Its main characteristics are the minimum inter-arrival time (IT) it allows, τ, and the organization's

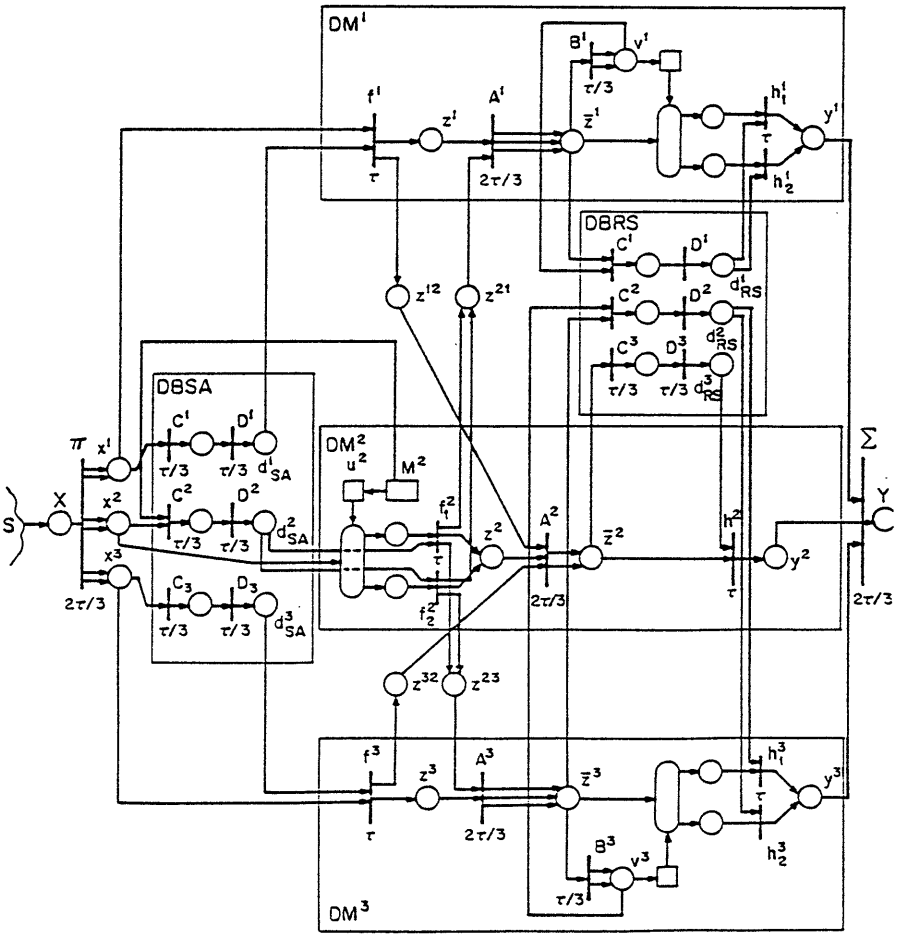


Figure 5. Protocol of Organization P Using Centralized Databases

total response time (RT), the time interval between the arrival of the input and the generation of a corresponding response, which is equal to $19\tau/3$.

Hierarchical Organization Structure

A hierarchical organizational structure allows decisionmakers to have an influence on each other's response selection. This influence can be represented by a command input, v^{on}. The hierarchical structure analyzed here is a three-person organization, known as organization H, equipped with centralized databases as shown in Figure 6.

Organization H consists of two decisionmakers who actually contribute to its output, DM^1 and DM^3, and one coordinating decisionmaker, DM^2, who analyzes DM^1's and DM^2's situation assessments in order to issue a command to them that carries his instructions about the way the organization's response should be constructed. DM^2 is not in contact with the environment, therefore he does not need an SA stage, neither do DM^1 and DM^3 need an information fusion transition, A. The three decisionmakers in organization H have each two RS algorithms.

One acceptable protocol for organization H is that represented in Figure 6. The minimum inter-arrival time, $11\tau/3$, is much greater than for organization P. This is due to the relationship between transition f_1^1 and DM^2, where the information coming out of f_1^1 has to be processed by all DM^2's transitions before transition B^1 can be fired and the last token leaves the place z^1. Application of the symmetric argument of the proposition's necessary conditions determined the mean interarrival time as $11\tau/3$. The organization's response time is calculated quite simply in this case, by adding all processing times along the path followed by the original input and is 8τ. For more complex organizations, the System Array approach is preferable for computing time delays (Tabak and Levis, 1984; Jin, 1985; Jin and Levis 1985.

3.3 Construction of Protocols for Decentralized Case

It was pointed out in section 2.2 that the only salient differences between a centralized and a decentralized structure as defined here pertain to transition D's processing time and the establishment of satisfactory updating. In this section, transition D is assumed to require a total time of $\tau/3$, which is half what was needed in the centralized configuration. Again, this number depends greatly on the nature of the organization's decision support system.

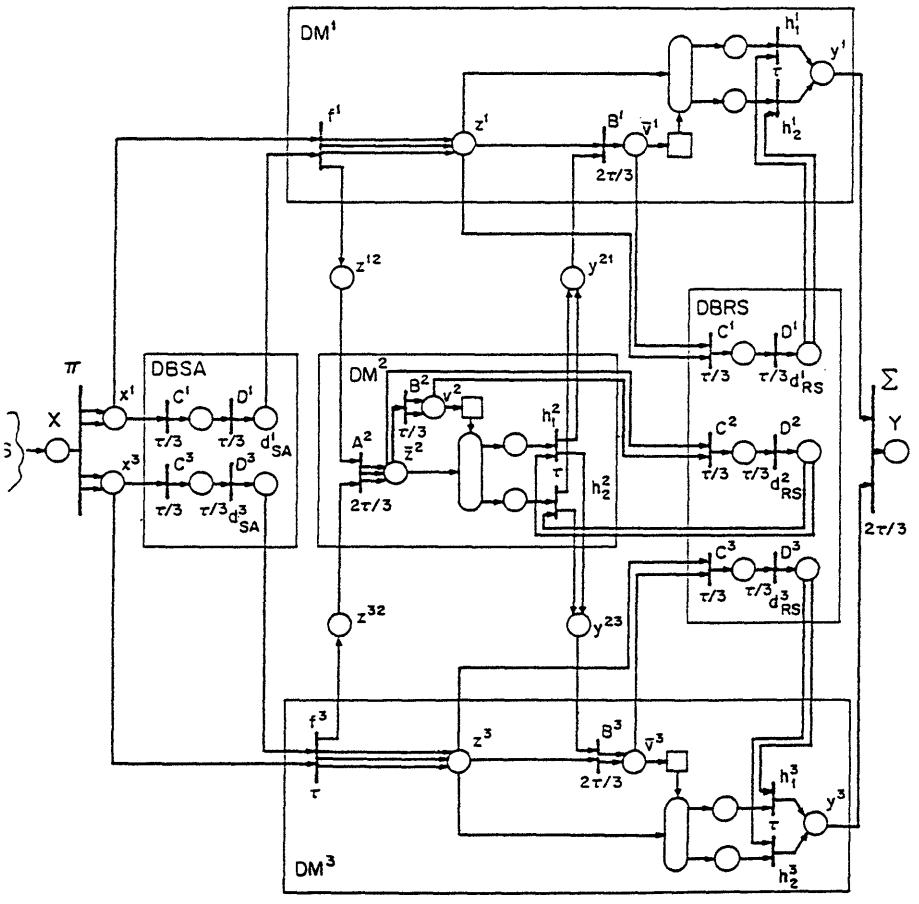


Figure 6. Protocol of Organization H Using Centralized Databases

Acceptable protocols for organizations P and H with decentralized databases are given in Figures 7 and 8, respectively. The minimun inter-arrival time IT and the response time RT for each organization are τ and $12\tau/3$ for the parallel one and $10\tau/3$ and 7τ for the hierarchical one.

The reduction in the IT and RT, when compared to the centralized cases, is due entirely to the shorter response time of the database.

3.4 Remarks

Each protocol in the previous sections has been derived under some very specific conditions, in order to make different organizations and different database structures comparable along the same criteria. These results are contingent upon using similar transition processing times for both organizational structures.

The minimum allowable input interarrival time (IT) is much greater for a hierarchical organization than for a parallel one. This follows from the more complex sequences of tasks that have to be performed in a hierarchical organization before a new input can be handled. The total response time is also greater for organization H than for organization P, and the difference is due again to the increased complexity.

The second observation is that, whatever the organization, a decentralized database structure leads to improved performance with respect to time. In organization P, the decentralized structure leads to an 11% improvement in the response time over the corresponding centralized one, while in organization H its leads to improvements in both IT and RT of 9% and 13% respectively. These results are due to the basic premise that decentralized databases takes less time to perform the data query process than centralized ones do. (The numerical results of the above two paragraphs are summarized in Table 1).

TABLE 1. TIME CHARACTERISTICS OF ORGANIZATIONS P AND H

	Centralized DB	Decentralized DB
IT(P)	τ	τ
IT(H)	$11\tau/3$	$10\tau/3$
RT(P)	$19\tau/3$	$17\tau/3$
RT(H)	8τ	7τ

IT = Minimum Allowable Interarrival Time;
RT = Response Time

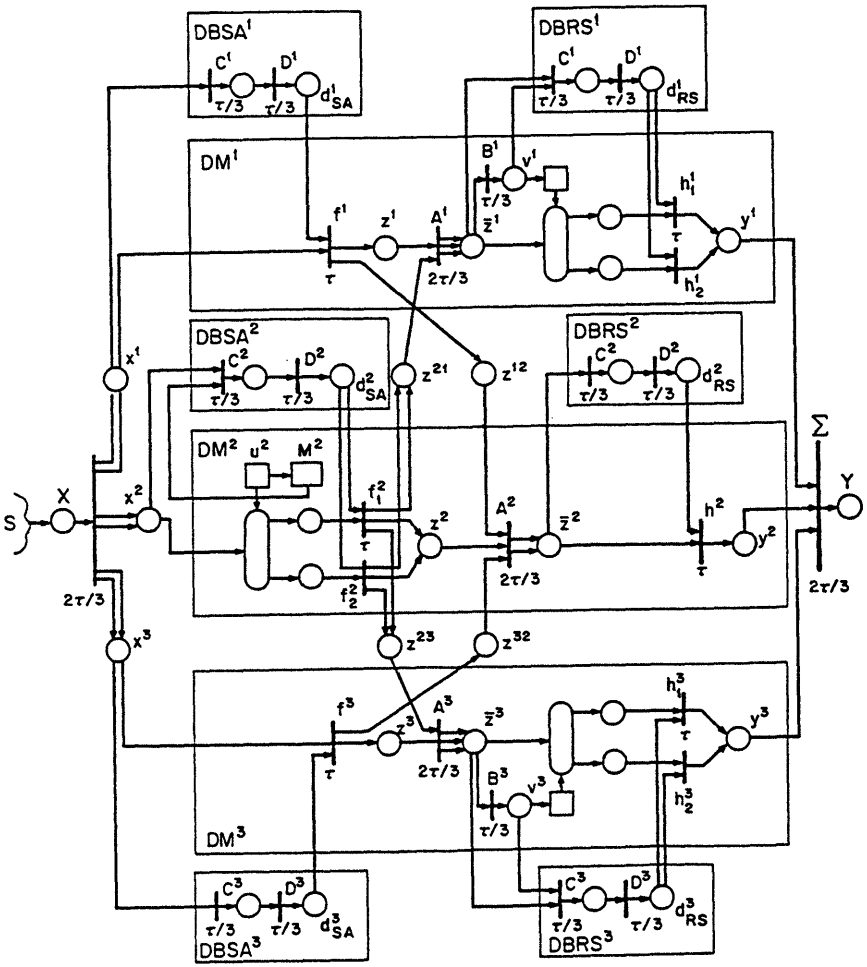


Figure 7. Protocol of Organization P Using Decentralized Databases

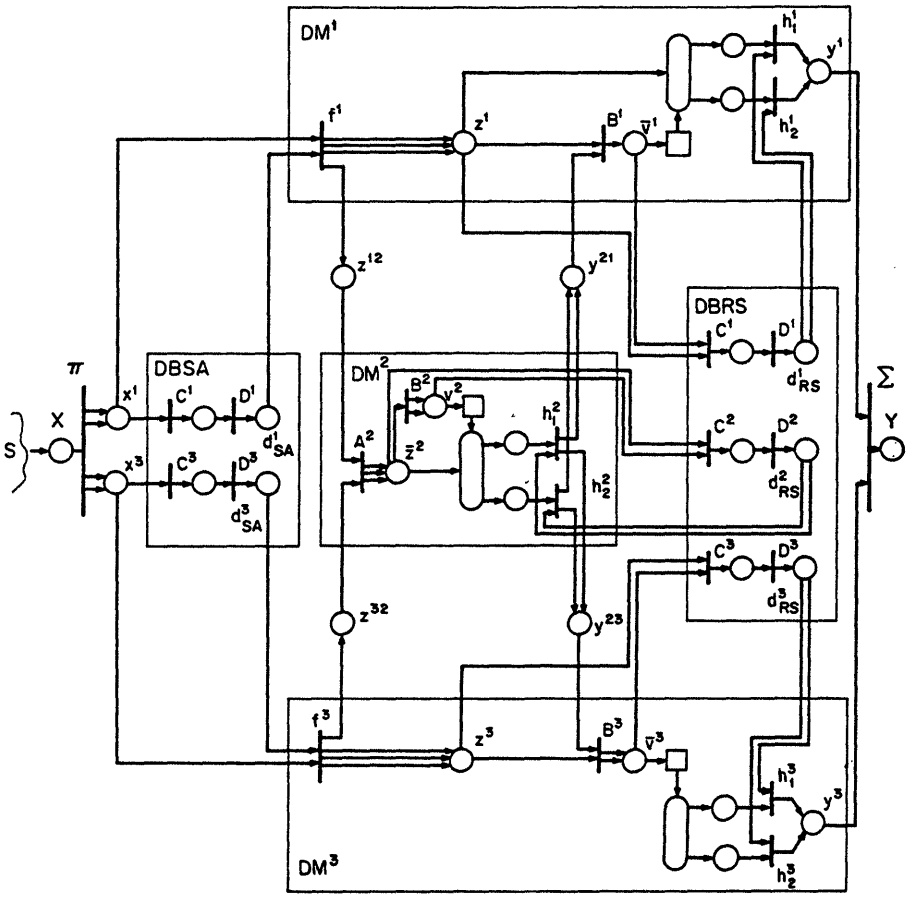


Figure 8. Protocol of Organization H Using Decentralized Databases

4. AN ILLUSTRATIVE EXAMPLE

4.1 Description of the Organizations Used

In this section, the two tactical organizations of section 3 are used to address the problems raised by a lack of coordination between several individual databases, and the trade-off between performance and timeliness. The approach is quantitative and relies on the construction and comparison of the performance-workload loci. (The example is developed in its entirety in Bejjani, 1985).

The first organization is the parallel one (Organization P) in Figure 5. It consists of three naval battle groups defending a maritime front. The first group, DM^1, holds one extremity of the front, DM^2 holds the center, and DM^3 the other end. The inputs received by the organization are signals emitted by unidentified platforms (submarines, surface ships, planes). The different decisionmakers' tasks are to attempt to identify the source of these signals (enemy or friends) in the SA stage, and to select the appropriate response (fire, request identification, or take all measures required to face an attack) in the RS stage.

The organization has two centralized databases. The SA database provides information, obtained from intelligence sources, that describes the codes the enemy could use when emitting the kind of signals received by organization P. This information will be compared to the actual input to determine the latter's identity. The RS database, DBRS, informs the decisionmakers about the level of alert present in their area at each iteration.

The second organization is the hierarchical one (Organization H) shown in Figure 6. The context is the same as for organization P, but here only DM^1 and DM^3 actually receive any external signals or select an active response. DM^2 is a coordinator who, based upon the situation assessments received from DM^1 and DM^3, gives instructions about what RS algorithm should be selected by either of them. The organization's overall mission is the same one defined for organization P. The two databases are again centralized and provide the same information as in Organization P except that d^2_{RS} in $DBRS^2$ is different to conform to the different role played by DM^2.

A primary feature of the example is its numerical simplicity: all the variables of the system are determined using binary logic based on the comparison of quantities; there are no actual computations. Detailed definition of the variables and the algorithms for the case of a single decisionmaker has already been presented (Boettcher, 1981). The performance measure J is the expected value of the cost the organization incurs when it does not produce the correct response for a given input. Instead of activity G, the activity rate G is used in constructing the performance workload locus. Activity rates are a better measure of the decisionmakers' workload than absolute activity, because of the time constraints present in real-world situations. The activity rate for this example is defined as:

$$F^i = \bar{G}^i = \frac{G^i}{\tau} \qquad i = 1,2,3 \qquad (\tau = 1) \qquad (8)$$

for either organization.

The basic step in the computation of the performance-workload pair (J,G) is determining the pure strategies present in the organization (Levis and Boettcher, 1983). In the cases at hand, each DM has two pure strategies, each obtained by the exclusive use of one algorithm (no decisionmaker here has, in any stage, more than two algorithms from which to choose). The workloads $\bar{G}^i$ determined by each pure strategy and the corresponding performance level J are plotted in the $(J, \bar{G}^1, \bar{G}^2, \bar{G}^3)$ space. Then, the performance-workload (P-W) locus for each DM is constructed where all possible mixed strategies are considered as linear combinations of the pure ones. The graphs thus obtained are projections of the overall (P-W) locus of the organization on each of three planes: $(\bar{G}^1,J)$, $(\bar{G}^2,J)$, $(\bar{G}^3,J)$. Because the input is perfectly symmentric, as well as DM^1,s and DM^3,s roles in each organization, only the projections for DM^1 and DM^2 are shown.

The use of activity rates illustrates the tradeoff between timeliness and workload (Figs. 9 and 10). The performance of organization P is better than that of H; the performance index J for P takes values between 0 and 0.9 but between 1.2 and 4.5 for H. However, the workload of the members of P is much higher than that of H, namely, it varies between 8.1 bits/sec and 11.6 for P, while it is only 1.2 and 2.7 for H. Thus, the parallel organization leads to better performance but this is achieved at a much higher workload. If the workload constraint is high, then P is the preferred organizational form. But if the bounded rationality constraint is active, then the hierarchical form may be preferred even though it leads to lower, but predictable performance.

However, another tradeoff appears here that involves the notion of timeliness. In effect, since in this example workload is a decreasing function of the mean interarrival time, Eq. (8), low workload levels are obtained by allowing a high IT, which penalizes the organization in terms of its timeliness. Thus, workload is reduced in H, but timeliness is sacrificed.

Another consideration of interest is the effect of poor updating coordination on the organization's performance when decentralized databases are used (e.g. Figures 7 and 8) the impact of the two different updating sequences on performance can be reflected on the)P-W) loci. In the first scheme, DM^2's and DM^3's RS databases are assumed to be updated, at $\tau + 0$, in coordination with the input arrival. DM^1's DBRS, however, is updated at $\tau + \tau$, with a delay of τ over the input to which the data correspond. New performance levels for each pure organization strategy were derived and a performance-workload locus was drawn; the projection on the $(J,\bar{G}^1)$ plane is shown in Fig. 11(a). The main effects are the upward movement of the original locus and a degradation in performance: the range of J is from 0.35 to 1.0 for the decentralized case as opposed to 0 to 0.9 for the perfectly coordinated (or the centralized) database case; this represents a drop of 29% in the average performance of the organization.

A second scheme exhibits a less coordinated updating sequence: $DBRS^2$ is updated at $\tau + 0$, $DBRS^1$ at $\tau + \tau$, and $DBRS^3$ at $\tau + 2\tau$. DM^1 and DM^3 both now have a greater propensity to make the wrong decision, and the resulting projection of the (P-W) locus on the $(J,\bar{G}^1)$ plane is presented in Fig. 11(b). The best performance (lowest J) is now 0.8, which is very close to what the worst performance was in the coordinated case, and the worst performance level is 1.2. The range of possible performance levels has shrunk further, and the drop in average performance with respect to the original case is 68%.

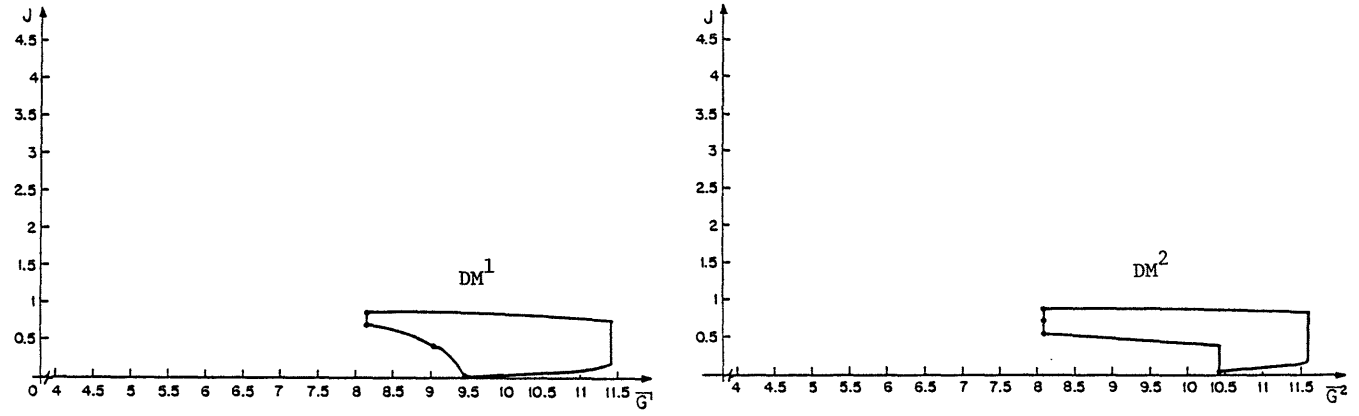


Figure 9. (P-W) Loci for P Using Activity Rates

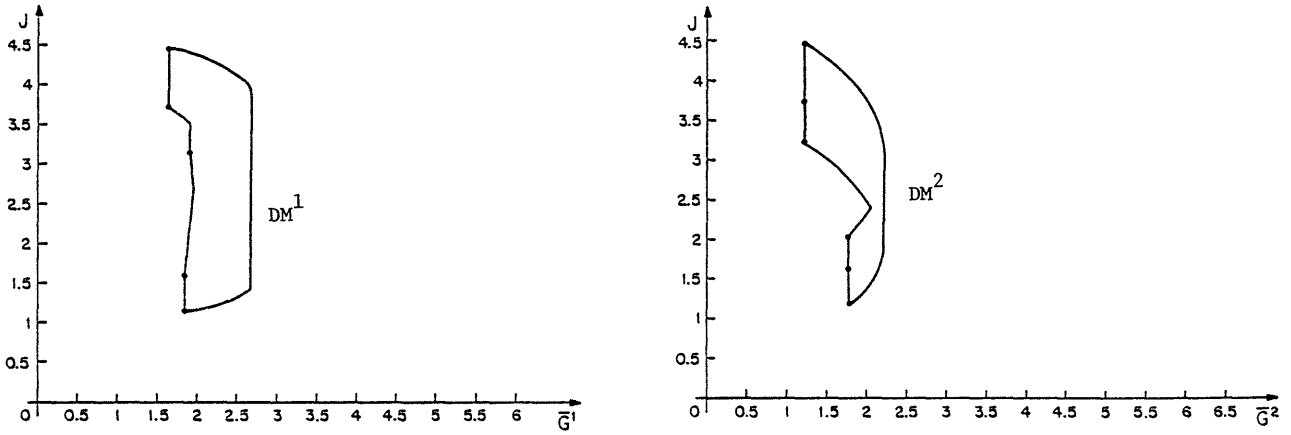
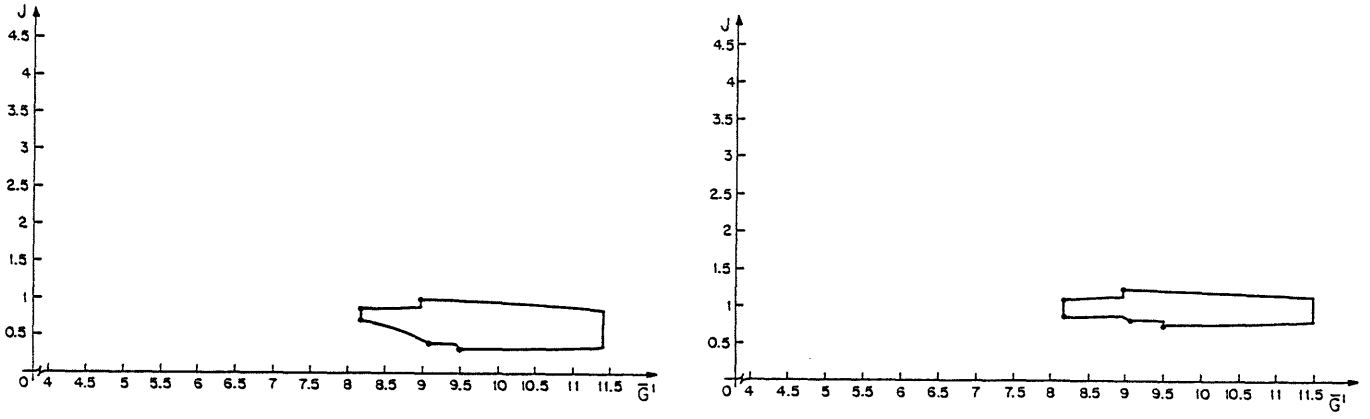


Figure 10. (P-W) Loci for H Using Activity Rates



(a) (b)

Figure 11. (P-W) Loci for DM^1 in P with Uncoordinated Databases

5. CONCLUSION

In this paper, the use of database networks was introduced into the organization, in two alternative configurations: centralized, and decentralized. Information theoretic aspects of data storage devices were analyzed. Time-related considerations were presented and used to create new criteria for the evaluation of the organization. An example illustrated the theoretical results.

6. REFERENCES

Bejjani, G. J., "Information Storage and Access in Decisionmaking Organizations," M.S. Thesis, LIDS-TH-1434, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, January 1985.

Boettcher, K. L. "An Information Theoretic Model of the Decision Maker," M.S. Thesis, LIDS-TH-1096, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, June 1981.

Boettcher, K. L. and A. H. Levis, "Modeling the Interacting Decisionmaker with Bounded Rationality," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-12, No. 3, May/June 1982.

Chyen, H. H-L, and A. H. Levis, "Analysis of Preprocessors and Decision Aids in Organizations," Proc. 2nd IFAC/IFIP/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man-Machine Systems, Varese, Italy, September 10-12, 1985.

Conant, R. C., "Laws of Information Which Govern Systems," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, No. 4, April 1976.

Cothier, P. H., and A. H. Levis "Assessment of Timeliness in Command and Control," Proc. 8th MIT/ONR Workshop on C^3 Systems, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1985.

Drenick, R. F., "Organization and Control," Directions in Large Scale Systems, Y.C. Ho and S. K. Mitter, Eds., Plenum Press, New York, 1976.

Gallager, R. G., Information Theory and Reliable Communication, John Wiley and Sons, Inc., New York, 1968.

Hall, S. A., "Information Theoretic Models of Storage and Memory," MS Thesis, LIDS-TH-1232, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, June 1982.

Hall, S. A., and A. H. Levis, "Information Theoretic Models of Memory in Human Decisionmaking Models," Proc. 9th World Congress of IFAC, Vol. VI, Budapest, Hungary, July 2-6, 1984.

Jin, V. "Delays for Distributed Decisionmaking Organizations," M.S. Thesis, LIDS-TH-1459, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, May 1985.

Jin, V., and A. H. Levis, "Computation of Delays in Acyclical Distributed Decisionmaking Organizations," Proc. 8th MIT/ONR Workshop on C^3 Systems, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1985.

Keen, P. G. W., "Value Analysis: Justifying Decision Support Systems," MIS Quarterly, March 1981.

Shannon, C. E. and W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, 1949.

Sheridan, T. B., and W. R. Ferrell, Man-Machine Systems, MIT Press, 1974.

Sprague, Jr., R. H., "A Framework for the Development of DSS," MIS Quarterly, December 1980.

Sprague, Jr., R. H. and E. D. Carlson, Building Effective Decision Support Systems, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Stabile, D. A. and A. H. Levis, "The Design of Information Structures: Basic Allocation Strategies for Organizations," Large-Scale Systems, Vol. 6, 1984, pp. 123-132.

Tabak, D. and A. H. Levis, "Petri Net Representation of Decision Models," Proc. 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Laboratory for Information and Decision Systems, MIT, Cambridge,MA,December 1984.

Tomovic, M. M., and A. H. Levis, "On the Design of Organizational Structures for Command and Control," Proc. 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1984.

RESPONSE DEPICTION FOR AUTOMATED COMBAT SYSTEMS

G. E. Frishkorn

J. R. Gersh

The Johns Hopkins University Applied Physics Laboratory
Laurel, Maryland 20707

Abstract

In previous work (1), we have discussed the general concept of automating naval combat system operation through the specification of rules of system response, called doctrine statements. This paper describes a mechanism for supporting the operation of such a system through the provision of a decision aid that models the system's response as it applies a specified set of rules to a given tactical situation. The role such a device plays in an overall decision structure made up of a human decision-maker and an automated system operating on the basis of specified response rules is described. An example of the operation of this decision aid, indicating how it serves to reduce the complex, multi-dimensional nature of a system response rule-set to an easily comprehended, two-dimensional picture is shown.

The Decision Structure of Doctrinal Control

In previous work (1), we have discussed the general concept of automating combat system operation through the specification of rules of system response, called doctrine statements. Under this concept, the system user (typically a ship's Commanding Officer or Tactical Action Officer) orders the system's response to specific tactical situations by entering and activating a set of rules that define what action the system is to take when the rules' preconditions are met.

In the format of Reference (2), Figure 1 shows the decision structure of an automated naval combat system operating under such rule-based direction of a human decision-maker. Each partner undertakes the two-step process of situation assessment and response selection, based on that portion of the overall situation information accessible to him/it (π^h for the human, π^m for the machine). Their responses, taken together, form the output of the overall system.

The partners interact with each other in two ways. First, they interact through the exchange of situation assessment **(SA)** information: the human can make direct inputs into the machine's data-base, and the situation display presented to the human can be partially controlled by the machine. Second, the human decision-maker directs the automated responses of the machine (**RS** = response selection) through the specification and activation of rules, some having to do with the situation assessment process (e.g., identification), and others having to do with the response (e.g., engagement).

The human decision-maker takes in information through an information fusion **(IF)** stage in which his overall picture is fused with the machine's assessment of the situation. This stage represents both the machine's control, through display rules, of the tactical display that the human observes, and, for example, the human observation of the machine's automated identification features.

The human then, through a particular assessment strategy ($\mathbf{AS_i}$), develops his own idea of the overall situation, and formulates a system response strategy ($\mathbf{SRS_i}$) to deal with it. This could include providing rules to the machine for situation assessment or response selection, or manual action by the human.

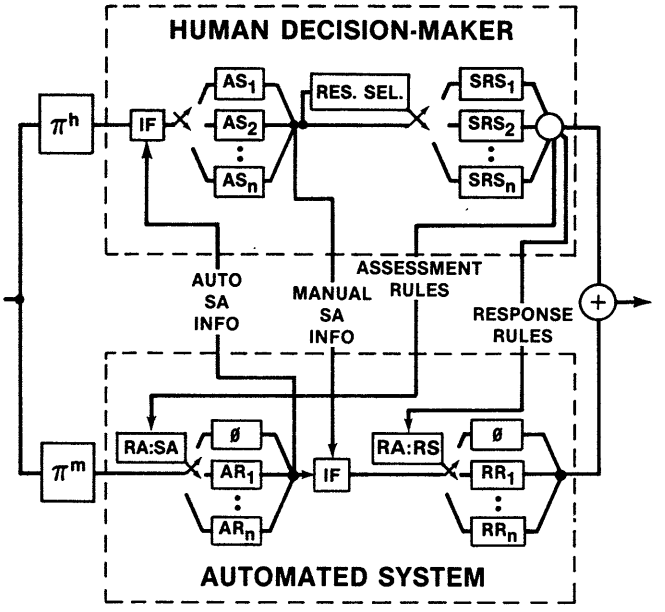


Figure 1 Overall Decision Structure

The machine's strategies are specified by the human through rule activation **(RA)**, for both situation assessment and response selection, choosing from a set of assessment rule sets ($\mathbf{AR_i}$) and response rule sets ($\mathbf{RR_i}$). Note that in each case the option exists for no rule-set to be selected; the system might be operated manually. Note also that the rule selection is entirely under the control of the human decision-maker, the information available to the machine itself has no effect on the rule-set selected.

The human can interact directly with the machine's database, for example, by making his own identification decisions. This is represented by the information fusion **(IF)** block following the machine's situation assessment.

The diagram of Figure 1 does not explicitly include the temporal sequence of doctrine statement selection, activation, and application. The rules active at any given time are those selected based on the assessment of the situation at some previous (perhaps just previous) time. Some of the apparent loops in the figure would disappear if the diagram were modified to reflect this, perhaps as a Petri net. Other loops are real, and reflect, for example, that the human decision-maker can instruct the machine what to include in its display. Current assessments can therefore be affected by past rule selections.

In order to select the appropriate overall system response, however, the human decision-maker must be able to predict the effect of the chosen rule-set in the assessed or in a postulated tactical situation. This means that he must play that situation through a model of the system's behavior as controlled by the rule-set in question. This might be a mental model, consisting only of the various rules' (*If* a track is in such a position and with such velocity...*then*...). Such an unaided decision process would be slow and cumbersome at best.

In fact, it would be much better to supply the human with a computer model of his system's automated responses so that he can expeditiously determine the efficacy of a particular rule-set in a particular situation.

The following section describes the characteristics and operation of such a model that we have built as a decision aid to support this response selection process.

The Response Depiction Model

The need for a decision aid to model rule-based system response comes from the nature of useful rule-sets. They will generally consist of many rules interacting in a complex pattern. In particular, the rule-writer can select from many different kinds of conditional parameters; a complete description of a rule-set requires a multi-dimensional space. In a naval combat system, the rules are generally based on vehicular track characteristics; the conditional parameters can include those that describe track position, kinematics, or identity.

The ideal system response decision aid would be able to model the outcome of any potential threat situation and indicate what ways, if any, the system might be modified to achieve an improved response. In practice the number of potential threat situations combined with the number of possible system configurations is too large to consider all possibilities. This is particularly true for rule-based systems where the response is not only sensitive to a multitude of threat characteristics but is also subject to a large number of user-defined parameters. Computer processing constraints and the need for timely information ultimately restrict the decision aid to a limited set of nominal threat situations.

Another consideration for decision aid design is that information be presented in a clear and concise format that is limited to the two-dimensional face of a CRT screen. While this goal may be an obvious one, its realization is not straightforward. For a complex rule-based system the problem lies in trying simultaneously to depict the effects of many multidimensional rules. Generally, some compromise might be made by using separate displays to present the effects contributed by individual elements in the rule-based system. This, however, places the burden on the human operator, who would be forced to mentally integrate the results of each element into the overall system response. A better solution would be to provide some means of compressing the effects and interactions of the many rules into a single, cogent depiction of system response. After all, the *response* of the system is what needs to be controlled; the conditional clauses of the various rules are just the means to that end.

A final consideration for decision aids which involve rule-based systems is the degree to which changes in operator specified rules are related to changes in system performance. For a system which uses many multidimensional rules that can interact with one another this can be a particularly difficult problem. There are two reasons for this: First, just one among the many operator-specified conditional parameters may actually affect the system response under any particular set of threat and environmental circumstances. Being able to determine which parameter this is can be virtually impossible if only the parameter values themselves are available and the overall system response itself is unknown. Second, the fact that the rules interact may make it difficult to isolate the rule which is actually responsible for a particular characteristic of system response. What might first appear as a response limit imposed by one rule may actually turn out to be a limitation imposed by a second rule whose action modulates the action of the first rule. Given that some means has been developed to display the system response, as described in the above paragraph, an additional capability to expand this information is needed so that response limitations can be associated with particular rules or even individual parameters in these rules. In other words, the operator needs a systematic means of determining what changes he would need to make the rule-base to achieve a given change in system response. In addition, interest in the response is as much directed toward why things don't happen as toward why they do. An internal mechanism that simply identified when specific rules are triggered, therefore, would not be sufficient.

The above features have been accomodated by a decision aid developed to aid control of the automated rule-based system described earlier. The decision aid models the action of the rule-based system against a worst-case threat situation described by specific target types taken to be radially inbound from any azimuth toward a defended point. (See Figure 2.) The action of the collection of all active rules is compressed into a depiction of the "response footprint." This shows the locus of points along potential inbound threat trajectories at which a particular response would be taken by the system. This format is the key to the compression of the rule-base's many dimensions into two.

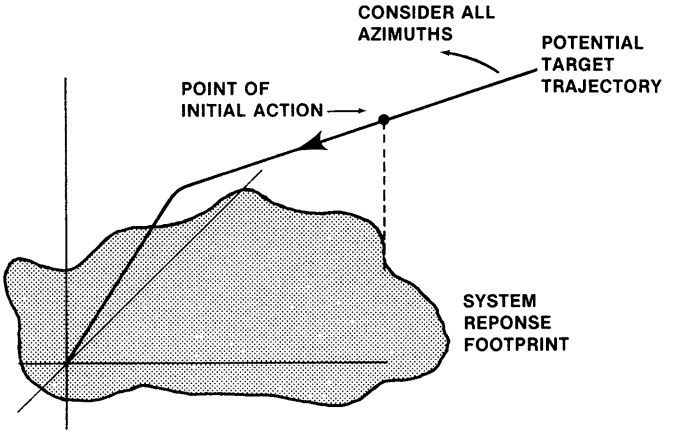


Figure 2 The Response Footprint

Figure 3 shows such a system response display as it was generated by the decision aid running in an HP-9020 desktop computer. The action taken here refers to the points for which surface-to-air missile (SAM) intercepts would be ordered by the system through doctrine statement action. It therefore includes the additional specification of firing unit location, thus setting the framework for modeling the area defense actions that would be initiated by the rule-based system against the selected target type. This decision aid would be used prior to action, for evaluating doctrine statement effectiveness against postulated threats.

This emphasizes the objective of the automated combat system, which is to provide antiair warfare area defense. At the same time, it can incorporate the effects of the many multidimensional rules into a single depiction of system response and thus effectively depicts the essence of the system response capability with a single display. For simplicity, this display was generated for a single doctrine statement. The outer line boundary represents the two-dimensional geometric parameter limits for the doctrine statement. The filled region represents the actual area of system intercept response using this doctrine statement against a specified target. Besides being specified as radially inbound toward the defended point, the target's profile, which includes altitude, velocity, and typical maneuvers, is also used to compute the doctrine response. This example illustrates the fact that plotting only the two-dimensional geometric parameters of the doctrine statement is not enough to indicate the region of its reponse. In this case another parameter, altitude, restricts intercepts ordered by the doctrine state-

ment to a region somewhat smaller than the specified range and bearing boundary.

Unfortunately, the identity of the crucial parameter cannot be ascertained from the response display alone. A solution to this problem might be to display a three-dimensional projection of the doctrine statement geometry and the response which includes the altitude parameter. Even this solution would fall short, however, when kinematic parameters like target course or speed turn out to limit the response. Thus, to obtain a complete understanding of the automated system a display of at least five dimensions would be required.

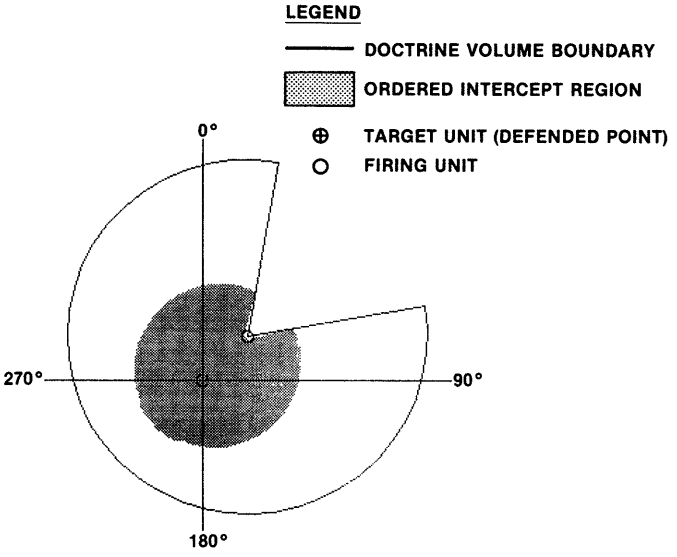


Figure 3 Doctrine Response Depiction

Before discarding this approach of depicting doctrine response, however, consider again what is represented. The region of system response by itself represents the effects of many doctrine statement parameters. This meets the original objective of compressing the effects of the multidimensional rules into a two-dimensional display of system performance. Although the example uses only one doctrine statement, the technique can clearly be extended by generating a composite response region associated with the action of multiple doctrine statements. Thus, the real shortcoming of the Figure 3 display is not in the depiction of system response, but in the depiction of the doctrine statement parameters themselves, of which only two (range, bearing) of many can be shown.

The motivation that exists for displaying all of the doctrine statement parameters has to do with the last decision aid objective, which was to provide a means to relate how changes in individual doctrine statement parameters will effect the overall doctrine response. What Figure 3 successfully shows is that the doctrine statement range parameter limit does not actually limit the intercept range in the system's response to this particular target. This is evident because the doctrine statement limit and the response limit do not coincide. By the same reasoning it shows that the doctrine statement bearing limits do restrict the response limit since these limits do coincide. This tells the decision maker that changes he might make to the bearing limits of the doctrine statement are likely to result in corresponding changes to the bearing limits of the response, whereas changes made to the range limit of the doctrine statement is unlikely to result in a corresponding change to the range limit of the response. All that is needed to complete the picture is a way to compare other doctrine statement parameters like altitude, course, and speed with the boundaries of the response region. This would permit the decision maker in this example to determine exactly which parameter will effect a change in the doctrine response range.

The proposed method for relating the remaining doctrine statement parameters to the effects they produce in the response uses vertical cross sections of the response plot. These cross sections indicate the response for one selected azimuth by displaying range on the **x** axis and the selected doctrine statement parameter on the **y** axis. The defended point is located at the origin. The doctrine zone boundaries and the target profile are shown according to the selected parameter dimensions. An example of such a plot is given in Figure 4 which is an altitude cross section of the Figure 3 response plot along the 000° bearing line.

This display clearly indicates that the limiting effect on the range of the doctrine response is the altitude parameter. Because the target cruises at a higher altitude than the maximum altitude limit of the doctrine statement, no response is possible until the target begins its turn down maneuver. The decision maker now has the necessary information to change doctrine statement parameters such that predictable changes in the way in which the system responds will result. In this case, if the object were to increase the response range, raising the altitude limit above the cruising altitude of the target would accomplish this goal. Further control of the response range would then be possible by simple adjustment of the maximum range parameter of the doctrine statement.

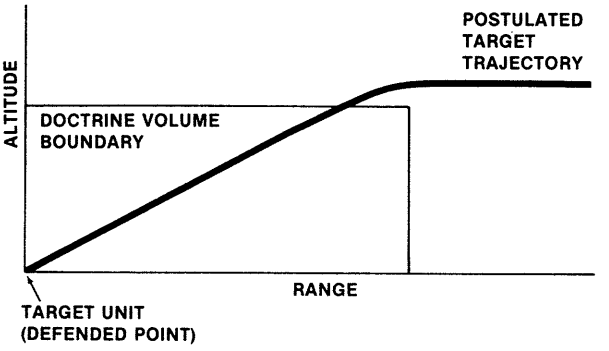


Figure 4 Altitude Cross Section; Bearing = 0°

These examples describe only SAM engagement doctrine. Clearly, there is more to combat system response than just those actions. Our system can also show an estimate of system **capability** to successfully engage targets following specific trajectories and includes the capability to model identification doctrine statements as well as engagement ones.

Future work will include the addition to the model of detection capability, given specified sensor doctrine and environmental conditions; and the effect of engagement scheduling constraints.

In conclusion, the doctrine response displays fill an important need of the decision maker in understanding the response of his automated system. Not only do they show the composite effects of many, multidimensional rules in a single depiction of system response, but even more importantly provide information necessary for the positive control of system capability against changing threat and environmental conditions.

(This work has been supported by the Naval Sea Systems Command, AEGIS Shipbuilding Program, under Contract No. N00024-85-C-5301.)

References:

(1) J. R. Gersh, "Rule-Based Automation of Command Action in Naval Combat Systems," Proceedings of the Seventh MIT/ONR Workshop on C^3 Systems, December, 1984, pp. 85-87.

(2) K. L. Boettcher and A. H. Levis, "Modeling the Interacting Decisionmaker with Bounded Rationality," IEEE Transactions on Systems, Man, and Cybernetics, SMC-12, No. 3 (May/June 1982), pp. 334-344.

A THEORY TO GUIDE THE DESIGN OF SITUATION ASSESSMENT AIDS FOR DECISION MAKING

DAVID NOBLE

ENGINEERING RESEARCH ASSOCIATES, INC.

ABSTRACT

Schemata are theoretical cognitive structures which model how people understand and organize information. Schemata have been used by psychologists interested in cognition, learning, linguistics, and social psychology. This paper discusses how schemata can be used to guide the design of situation assessments aids. These guidelines are based on the premise that schemata are the cognitive structures used for understanding situations and that data presented to correspond to the way that people structure information for situation understanding will be more easily understood, can be more easily combined with other data available to the decision maker, and can be more deliberately designed to support a sound decision process. This article briefly summarizes the principal features of schema theory applicable to situation assessment aid design, and illustrates the properties of situation aids whose structure and organization parallels the structure and organization of schemata.

SITUATION ASSESSMENT AIDS

A situation assessment is an interpretation of the meaning of a situation. This assessment includes what events are occurring at present, what people and equipment are participating in these events, why these events are occuring, what changes to these events may occur in the future, what actions may be taken to influence these future events, and how future events are likely to respond to such actions. Determining these issues and forming an accurate situation assessment is a critical requirement for tactical decision making.

A situation assessment as defined above is clearly much more than just a chart of the situation players and events. It also includes a model of the relationships between the players and their environment, beliefs about the objectives of the players, predictions of how the different players are likely to act in order to achieve these events, and a framework for predicting possible changes in this behavior that may result from when different kinds of obstacles are encountered.

Situation assessments are developed by combining observations about the physical nature of the situation with background knowledge about the events that usually occur in these situations. The observables include the appearance of the players and objects. In military settings the background knowledge includes the capabilities of hostile platforms, the way that these different platforms need to support one another, and the tactics and doctrine which an adversary is expected to use to attain his objectives.

Unfortunately, it is often very difficult for people to accurately assess situations. The ability to assess a situation accurately can be limited by poor data quality, by the use of faulty judgemental heuristics, by limited experience, and by limitations on the amount of data that people can retain and evaluate at one time. Frequently, data quality is poor because the observables are incomplete, ambiguous, and unreliable. People often interpret the data using faulty judgemental heuristics which weight too heavily redundant or unreliable data, which focus only on certain preferred aspects of the situation, and which preclude proper consideration of undesired interpretations of the data. Human information processing limitations restrict the number of alternative interpretations of the data that may be considered, and limit the ability to forecast the impact of possible actions. Inexperience prevents people from associating observed behaviors with the reasons for these behaviors and from predicting possible future events.

Because of the difficulties that people can have making accurate situation assessments, there have been attempts to use computers to help people make better situation assessments. These computer-based situation assessment aids help by performing part of the information processing required for situation assessment. They may filter and correlate data, may infer unobserved features of the situation from these data, and may relate patterns of observed and inferred situation features to the most likely causes of these patterns.

Frequently, for complex situation assessments such as those encountered in military contexts, the operator and computer aid must interact continually. The machine performs operations on limited pieces of the problem and then hands over the product of its processing to the operator. The operator in turn uses his experience and judgement to combine this product with other information, and may then task the computer to perform further processing.

If this interactive man-machine system is to be effective, then the operator must understand how the system's computations relate to his judgements, and the computer system should discourage faulty judgemental heuristics. Designing situation assessment systems for which this is true has proven difficult to do. It is possible, however, that these design goals may be more easily met in situation assessment aids whose structure parallels the schema structure proposed to underly human information processing and understanding. Because in such aids the organization of presented information parallels the proposed organization of information in the mind,

information so presented may be more easily understood, may be more easily combined with other data available to the operator, and may be more easily formatted to support a sound decision process.

SCHEMA THEORY

There are three aspects of schema theory most important to the design of situation assessment aids. These are the use of a large number of template-like schemata to encode knowledge, the way that a schema constrains the values of the entities within the schema, and the way that a particular schema is selected to represent a situation.

According to schema theory, a person's knowledge may be modeled as a large set of general time-event and part-whole models. Generally, the time-event models have been called scripts, and are considered a specialized kind of schemata. This paper will use the term schemata to include scripts. Each of these general schemata represents a class of situations with similar properties. Every situation that a person can understand must be understood in terms of these schemata.

Each of these schemata is characterized by a set of features. These features are the components of the schema -- its events, participants, and equipment. Each feature can take on a value. When a schema is used to represent a class of situations, its feature values are specified only to within a permissible range. This range may be interpreted as a constraint on the values which each feature may assume when the schema is used to represent a specific situation. In that case, those schema features that are observed in any particular situation assume the values of the feature in the situation. Those schema features that are not directly observed will take on a default value for that feature.

Schemata function within a network of related schemata. One of the most important of these relationships is the hierarchial relationship between features and schema. Under this relationship features will sometimes be schema themselves, with their own events, participants, and equipment. Thus schema may be embedded within other schema.

A situation is understood when it is recognized that one of the schemata in memory is a good model for the situation. This recognition occurs when the values of enough of the features in the observed situation meet the constraints on feature values imposed by the schema. If the values of the features in a situation are consistent with several schema, then it may be uncertain which of these schema should serve as the model for the situation.

These concepts can be explained further with the aid of a simple example previously discussed by David Rummelhart, a cognitive psychologist who has used schema theory to model the comprehension of stories. This example is repeated here because it illustrates particularly well the nature of schemata as explained by cognitive psychologists, because it references a widely shared and commonplace experience, and because it has appeared several times in review articles about schemata. It is not intended, of course, to suggest that a situation assessment aid be developed for this particular situation.

The text to be understood with schema theory is the following: "Mary heard the ice cream truck coming down the street. She remembered her birthday money and rushed into the house."(1) Most people can interpret this text very easily. Mary is probably a little girl who knows what ice cream trucks sound like, knows that ice cream is available from these trucks, knows that money is required to attain the ice cream, and realizes that her birthday money can be used to purchase ice cream. This interpretation cannot be derived solely from the information in the text. It requires in addition that the person reading the text have a structure in memory that models purchasing ice cream from ice cream trucks. Schema provides the needed structure.

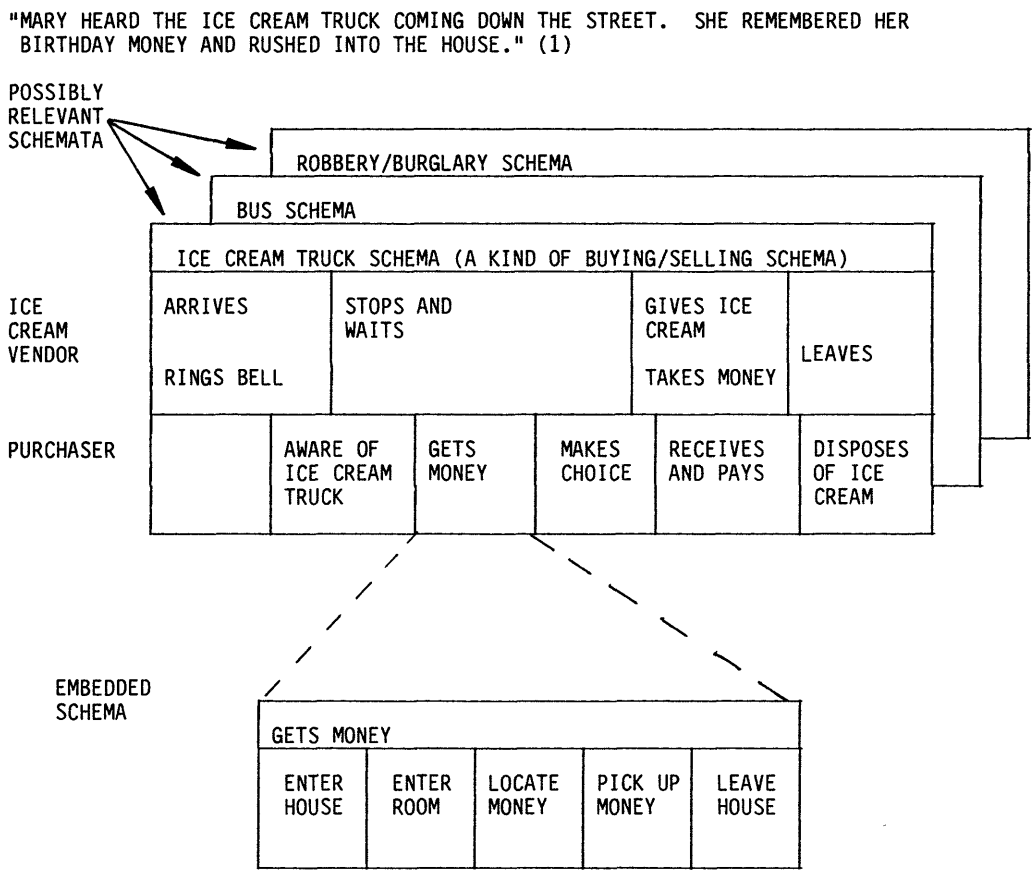
Each of the key aspects of a schema can be explained in terms of this example (Figure 1). The first aspect is that knowledge can be modeled as a large set of time-event or part-whole relationships. In this case, there are several time-event models that could be used to understand this text. Figure 1 indicates three of these. There is one for buying ice cream from ice cream trucks, there is another for taking a bus, and there is another for robbery. Although in the present case the text suggests that the first one is the relevant schema, the other two schema could be made relevant by minor changes in the story. For example, if the text were taken from a story about bandits who pose as ice cream venders that take children's money without providing any ice cream in exchange, then the bandit schema would provide the proper interpretation of the text.

The second aspect is the nature of features which characterize the schema. For the ice cream truck schema there are three different kinds of features: the people, the objects, and the events. The people are the ice cream vendor and the purchaser. The objects are the truck, the money, and the ice cream, and the events include the purchaser becoming aware of the ice cream truck and the vendor providing the ice cream.

The values that each of the features in the schema may take are constrained by the schema. The purchaser, for example, must be a person. In our example, Mary meets the constraints on the purchaser, and is assigned that role. The schema also constrains the relationships between events. The ice cream truck must arrive before the ice cream is provided. There are many features whose values are not provided by the text. The reader assigns default values to these features. The amount of money required to buy the ice cream is assigned a default value of less than a dollar, small enough to be purchased with birthday money.

The features in this schema may be schemata themselves. For example, the feature "gets money" has a schema whose events include entering the house, entering her room, locating her birthday money, picking up the money and leaving the house. This schema is considered to be embedded within the larger ice cream schema. Its events represent a greater level of detail than the event "gets money" and will be accessed only if needed for understanding the text. This embedding of details keeps the reader from being distracted by details that are not needed, yet allows him to readily access those details that are needed.

FIGURE 1 - EXAMPLE OF SCHEMA USED FOR UNDERSTANDING



The third aspect of schema theory concerns how a person recognizes that a particular schema can provide a good model of the situation. Schema theory assumes that such recognition occurs when enough of the people, objects, and events observed in a situation meet the constraints imposed on feature values by the schema. In this case the ice cream truck schema was selected to explain the text because several of the text references are consistent with this schema. Mary has the properties of a potential buyer because Mary is a person's name, and Mary can hear, can remember, and has birthday money. The ice cream truck is explicitly mentioned both in the text and also the schema. Birthday money meets the constraints on the means to attain ice cream. Mary's rushing into the house is consistent with the fact that ice cream trucks pass through quickly.

Once the ice cream truck schema is selected as representing the situation, the schema can be used to make inferences about other unobserved features and can be used to make predictions about future events. The schema also provides a context to explain why these events are occuring. These properties of schema are important components of a situation assessment.

APPLICATION OF SCHEMA THEORY TO DESIGN OF SITUATION ASSESSMENT AIDS

Because schemata include all the elements required for a situation understanding, situation assessment aids modeled after schema will also include these elements. The translation from schema theory to situation assessment aid is straightforward. Schemata themselves are structures in memory; their counterpart in situation assessment aids are time-event and part-whole structures in a computer data base. Table 1 summarizes the parallels between schemata and situation assessment aids.

The following example illustrates a situation assessment aid that is designed after the schema model. In this example, there has been a satellite observation that reports that an unknown number of large aircraft have taken off from a northern area airfield. Intelligence indicates that the large aircraft at this airfield are bombers, transport, and command and control aircraft.

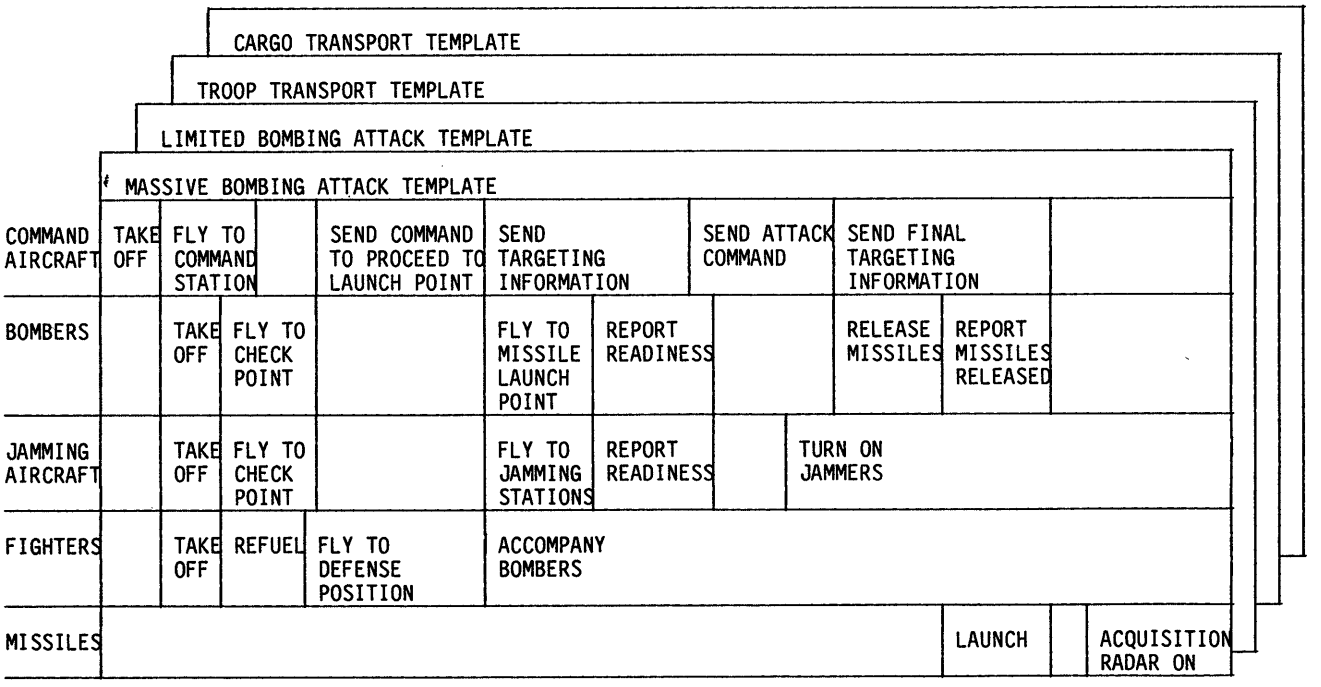
The situation assessment aid must help the operator understand what this report means in the context of other activities that are occurring, must

TABLE 1 - APPLICATION OF SCHEMA (SCRIPT) THEORY TO DESIGN OF SITUATION ASSESSMENT AIDS

SCHEMA PROPERTY	COUNTERPART IN SITUATION ASSESSMENT AID DESIGN
1. KNOWLEDGE IS REPRESENTED AS A LARGE NUMBER OF SCHEMATA.	1. POSSIBLE SITUATIONS OF INTEREST ARE PRESENTED AS A SET OF SCHEMA-LIKE TEMPLATES.
2. EACH SCHEMA IS CHARACTERIZED BY A SET OF EVENTS OR FEATURES. SCHEMATA CONSTRAIN VALUES THAT FEATURES MAY ASSUME. VALUES NOT DEFINED BY SITUATION ASSUME DEFAULT VALUES. FEATURES MAY BE SCHEMATA THEMSELVES.	2. DISPLAYS RELATE COMPONENT FEATURES TO TEMPLATES. DISPLAYS INDICATE POSSIBLE (REFERENCE) VALUES OF FEATURES CONSISTENT WITH TEMPLATE. FEATURES NOT OBSERVED ARE ASSUMED TO HAVE VALUES WITHIN A DEFAULT RANGE. THESE MAY CHANGE AS OTHER FEATURES ON TEMPLATE ARE GIVEN VALUES. REVISED DEFAULT VALUES ARE "INFERENCES". EMBEDDED TEMPLATES ARE ORGANIZED ACCORDING TO THEIR EMBEDDING TEMPLATES. EMBEDDED TEMPLATES PROVIDE DETAILS OF COMPONENT FEATURES.
3. A SITUATION IS UNDERSTOOD WHEN A SCHEMA IS SELECTED. SELECTION OCCURS WHEN FEATURES OF A SITUATION ADEQUATELY MATCH FEATURES OF SCHEMA.	3. A SELECTED TEMPLATE PROVIDES A SITUATION UNDERSTANDING. SITUATION ASSESSMENT AID SHOWS THE PATTERN OF FEATURES PRESENT IN A SITUATION, EMPHASIZING THE CLOSENESS OF MATCH BETWEEN TEMPLATE REFERENCE AND FEATURE.

FIGURE 2 - EXAMPLE OF SITUATION AID DISPLAYS

POSSIBLE SITUATIONS REPRESENTED AS ALTERNATE TIME-EVENT TEMPLATES



help him predict the different events that may unfold in the future, and must help him understand what actions should be taken in anticipation of these events. In keeping with the schema structure, the situation assessment aid is organized around a set of time-event templates, each containing the take-off of large aircraft from an airfield at a distance like that of the airfield in the report. Figure 2 suggests several templates consistent with this report. These include templates for a massive bombing attack, for a limited bombing attack, for troop transport, and for cargo transport. Given only the reported intelligence, it is not possible to determine which of these templates represents what is actually occuring. This determination requires additional information. Once this determination is made, the template becomes the basis of the situation assessment.

The template for the massive bombing attack specifies a time sequence of events associated with a set of "players". These players are command aircraft, bombers, jamming aircraft, fighters and missiles. The events as indicated on the template include take off, flying to a check point, sending targeting information, and launching missiles. One of the ways that the schema-like situation assessment aids helps with situation assessments is by making these templates available to the operator.

The situation assessment aid specifies constraints on the nature of the players and events. For example, the template may require that at least twenty bombers of particular types be involved. If later reports should indicate that fewer bombers took off, then the massive bombing attack would be eliminated as a possible interpretation of the situation. The template also constrains the nature of the events that may occur. It would specify, for example, that targeting information would be sent by certain types of platforms in certain prescribed formats at certain allowable times. These constraints serve two important purposes: they provide guidance on what observables are consistent with the template (e.g., with an all-out attack), and they also limit the range of what may occur in the future given that a particular template represents the ongoing situation.

Like the schema structure, the situation assessment aid provides more detailed information as embedded templates. For example, the event "send targeting information" can be represented as a template that has two players, the command aircraft and the bombers, and a sequence of events such as "receipt for information from HQ", "send data to bombers", and "receipt for targeting data".

Schema-like situation aids do not have to present information as time-event models. They may also present information to emphasize the spatial relationships among players and events. Figure 3 illustrates how information about the event "send and receipt targeting information" may be represented spatially. Because this display is modeled after schemata, it has two components, a set of reference values and a set of observables. The reference values indicate the locations at which a receipt is expected and permitted. They correspond to the schema contraints. The observables indicate the positions at which receipts have been observed. They correspond to the situation observables. Figure 3 indicates that the observables easily meet the schema constraints. In this case the operator would probably decide that the observed receipts substantiate that a large scale bomber attack is underway.

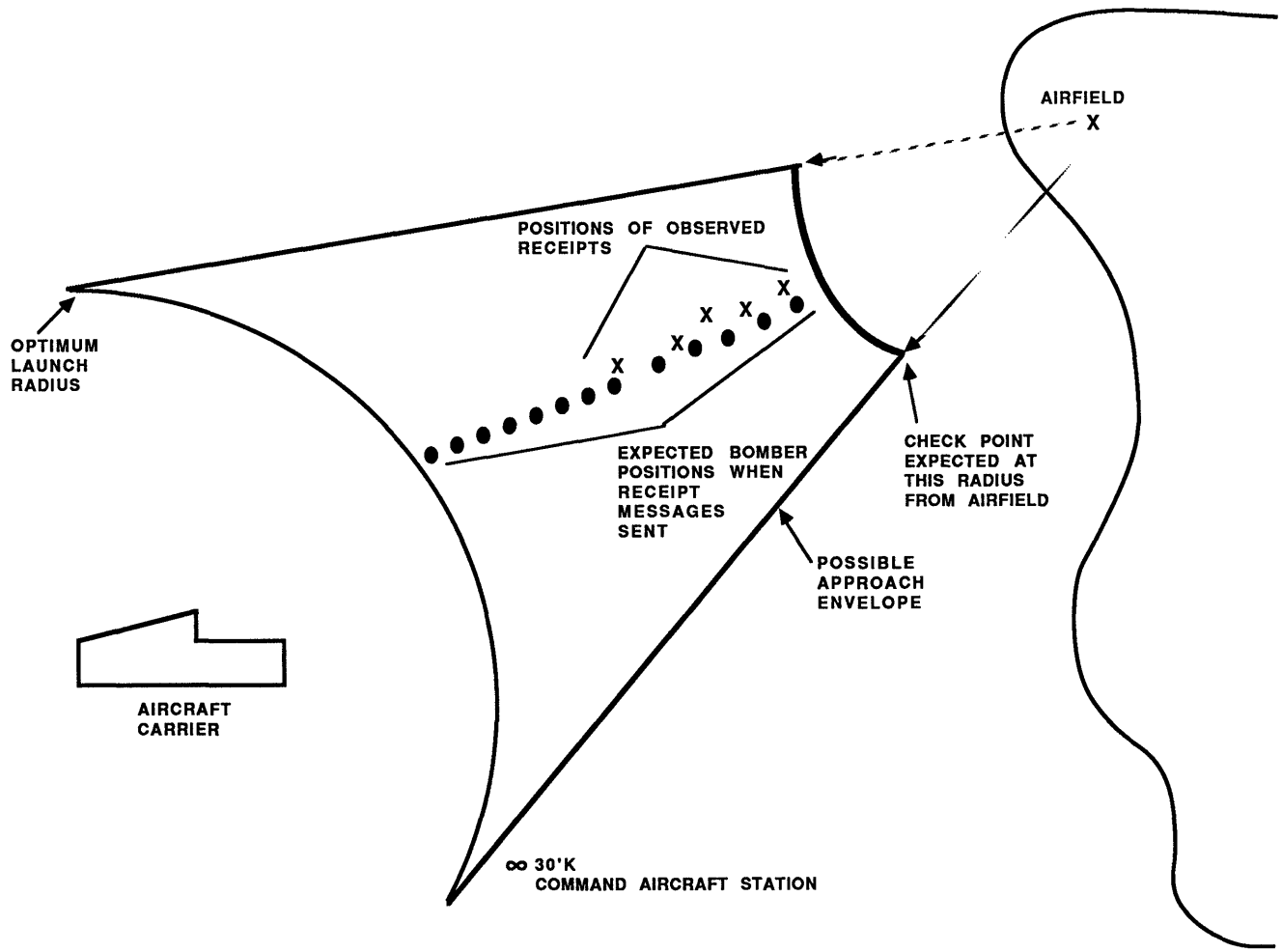
In practice, the schema-like situation aid works as follows. After the initial intelligence report, the aid would inform the operator of the general situations which are compatible with the report. The operator establishes a goal to determine which of these possible general situations is actually ongoing. The aid analyzes the different events on the templates for these situations, and informs the operator of the past or future events that can best confirm or eliminate each of the possible situations. The operator will examine the system data base or query other systems to display information for determining which of these past events have occurred. This information will be displayed so that the characteristics of actual events may be compared to the characterists of the reference events. The operator may also task information collection assets capable of collecting the data most useful for resolving the remaining situation ambiguities.

When the operator has recognized which templates relate to the actual situation, he has a good start at assessing the situation. With these templates he can interpret the meaning of the situation. He knows the events which are occurring at present, he knows what people and equipment are participating in these events, and he knows why the events are occurring.

(1) Rumelhart, David "Understanding Understanding" by David Rumerhart. In J. Flood (Ed.) Understanding, Reading, Comprehension. Newark, Delaware. International Reading

(2) Noble, David and Truelove, Joseph "Schema-based Theory of Information Presentation for Distributed Decision Making". Engineering Research Associates, Report R-028-85, 1985.

FIGURE 3 - SITUATION AND EVENT ANALYSIS FOR "SEND AND RECEIPT TARGETING INFORMATION"



C^3 Systems Modelling*

S. Rosenstark and J. Frank

Electrical Engineering Department
New Jersey Institute of Technology
Newark, N. J. 07102

Abstract

As part of an ongoing effort to develop a methodology for analyzing C^3 systems, some simple command problems were investigated. Utilizing the average cost of a mission as the criterion for the effectiveness of a command decision, some problems in positioning a weapon were analyzed. Utilizing a previously derived result for an artillery-observer system, an optimum position could be found which minimized the average mission cost. Curves are presented showing average mission cost as a function of the decision variable.

Introduction

An analytical theory of C^3 systems would be an extremely useful tool to designers and evaluators of C^3 systems. A C^3 system reference model was proposed [1] which decomposed C^3 systems into a set of basic building blocks called primitives. The set of primitives is shown in figure 1. The term "counter" is used in the broad sense to include all means available to inflict damage and minimize being damaged.

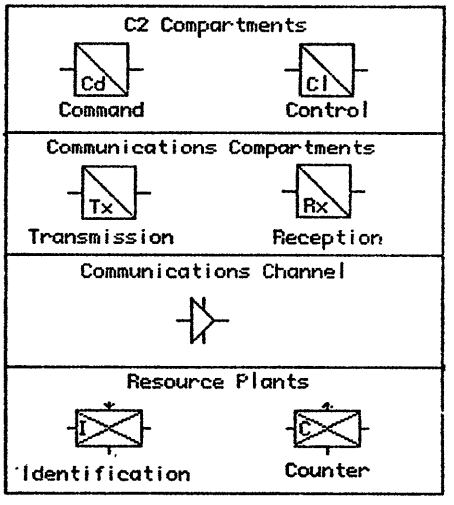


Figure 1 - C^3 system primitives.

For the purpose of evaluating the performance of a friendly C^3 system in combat with an enemy C^3 system, a function called the utility function was defined. This function was designed to provide a measure of the superiority of one system over another and to be consistent with Lanchester's theory of combat. The utility function is expressed in terms of another function called the capability function which is a measure of effectiveness of a single element of a C^3 system. In [1], the capability function of an identification plant-counter plant system consisting of an artillery piece and a forward observer was worked out in detail. In this case the capability function was the probability of a kill, and the expression obtained, which will be used in this paper is

$$P(K) = \frac{\sigma_K^2}{\sigma_I^2+\sigma_C^2+\sigma_K^2} \qquad (1)$$

In the above, σ_K is the kill radius of the shell, σ_I is the standard deviation of the the observer and σ_C is the standard deviation of the counter. In the present paper we will deal with the more difficult problem of evaluating the performance of the command block.

Command Capability

Some of the functions of the command block in a C^3 system are the positioning of units and the allocation of resources for a particular mission. A measure of the performance of the command block in carrying out these functions is the cost associated with the carrying out of a mission. As many of the parameters involved in a mission are random variables, it seems appropriate to use the average cost of a mission as a figure of merit for evaluating the performance of the command block.

To obtain some insight into the problem of allocating resources in order to minimize the cost of a mission that the command block is faced with, we will consider a series of command problems of increasing complexity.

Counter Positioning Problem

Assume we have a counter (which is in this case depicted as a gun) at an initial position r_0 from the target. The counter has a supply of n shells with an associated cost of 1 unit/shell. The counter can move to a new position a distance r from the target before starting to fire as shown in

*This work was supported by the U. S. Army, CECOM Center for Systems Engineering and Integration at Ft. Monmouth, N. J., under contract number DAAB07-83-K-K549.

figure 2.

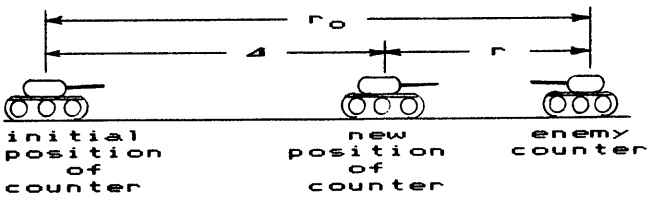


Figure 2 - Geometry of the problem.

The cost of moving the counter is b units/mile. The mission of the counter is to destroy the target, and the command problem is to choose the value of r to minimize the total cost of the mission. The cost of the mission is the sum of the cost of the shells and the cost of moving the counter.

The probability of a kill on each shot is P_k, which is a function of r. We will assume cylindrical symmetry so that (1) will apply in this case. Furthermore it will be assumed that the counter error σ_C and the intelligence error σ_I are both proportional to the distance from the target. Accordingly we will replace $\sigma_C^2+\sigma_I^2$ in (1) by a^2r^2 so that we will now use a simplified form of the kill probability

$$P_k(r) = \frac{\sigma_k^2}{\sigma_k^2+a^2r^2} \qquad (2)$$

where σ_k is the kill radius of the shell and a is the standard deviation of the counter at a range of one unit.

It will be assumed that the counter fires at the target, until the target is destroyed or the counter's ammunition is depleted. To find the average cost of completing the mission, we take the cost of completing the mission using i shells which is i units, and multiply by the probability of completing the mission with i shells. This probability is the probability of i-1 misses, times the probability of a kill. This has to be summed for all possible i from 1 to n.

$$C_s = 1P_k+2(1-P_k)P_k+3(1-P_k)^2P_k+$$
$$\ldots+(n-1)(1-P_k)^{n-2}P_k+n(1-P_k)^{n-1} \qquad (3)$$

The term on the extreme right of (3) is not multiplied by P_k, as the cost is independent of a hit or a miss when the last shell has to be fired.

Using

$$P_m = 1-P_k = \text{probability of a miss} \qquad (4)$$

in (3), we obtain

$$C_s = P_k\,[1+2P_m+3P_m^2+\ldots+(n-1)P_m^{n-2}+nP_m^{n-1}] \qquad (5)$$

and it becomes clear that we have been dealing with a finite-length modified geometric series. The general form for the sum of this series is

$$\sum_{i=1}^{n-1} ix^{i-1} = \frac{1+(n-1)x^n-nx^{n-1}}{(1-x)^2} \qquad (6)$$

Using (6) in (5), we obtain the following expression for the average cost of the shells

$$C_s = P_k\frac{1+(n-1)P_m^n-nP_m^{n-1}}{(1-P_m)^2}+nP_m^{n-1} \qquad (7)$$

As $n\to\infty$ all the terms involving P_m^n and P_m^{n-1} go to zero and

$$\lim_{n\to\infty} C_s = \frac{P_k}{(1-P_m)^2} = \frac{1}{P_k} \qquad (8)$$

In the subsequent analysis we will assume that $n\gg1$ and use

$$C_s = \frac{1}{P_k} \qquad (9)$$

to simplify the calculations. In terms of the distance r, we have upon using (2) in (9)

$$C_s = \frac{\sigma_k^2+a^2r^2}{\sigma_k^2} \qquad (10)$$

It will be assumed that the cost of moving is proportional to the distance moved

$$C_m = b|r_0-r| \qquad (11)$$

where b is the cost per unit distance moved.

The average total cost of a mission, (where a mission is defined as the moving of the counter from r_0 to r and firing until the target is destroyed) is given by

$$C_t = b|r_0-r| + \frac{\sigma_k^2+a^2r^2}{\sigma_k^2} \qquad (12)$$

To minimize C_t we have to consider two regions in (12). If $r>r_0$ then both terms on the right side of the equation increase as r increases from r_0, so there is no minimum in that region. For $r<r_0$ as r decreases from r_0 the first term on the right of (12) increases whereas the second term decreases. We can therefore expect to find a minimum of C_t in the region $r<r_0$. The equation for C_t in that region is

$$C_t = b(r_0-r) + \frac{\sigma_k^2+a^2r^2}{\sigma_k^2} \qquad (13)$$

To minimize the total average cost with respect to r, we differentiate C_t with respect to r and set the derivative to zero, which yields

$$-b + \frac{2a^2}{\sigma_k^2} r = 0$$

with the result that the optimum r for minimizing the cost of the encounter is

$$r_{opt} = \frac{\sigma_k^2 b}{2a^2} \qquad (14)$$

If the counter is moved to r_{opt} then the cost of the encounter will be $(C_t)_{min}$ which is obtained by substituting (14) into (13)

$$(C_t)_{min} = b\left[r_0 - \frac{\sigma_k^2 b}{2a^2}\right] + \left[1 + \frac{b^2}{2a^2}\right] \qquad (15)$$

As an example let $a^2=1$, $b=0.5$, $\sigma_k^2=1$ and $r_0=10$. We find that $r_{opt}=0.25$ and $(C_t)_{min}=1.938$. When these parameters are substituted into (13), the expression for C_t becomes

$$C_t = r^2 - 0.5r + 2$$

Figure 3 shows a plot of C_t as a function of Δ for this example.

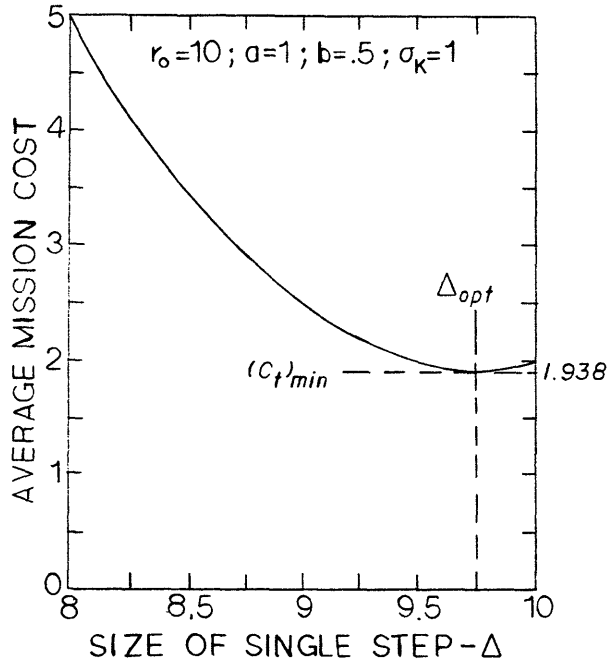


Figure 3 - Total cost as a function of step size Δ for the illustrative example.

13. Counter Moves, Shoots, Moves, Shoots,...

This is the analysis for the case where only friendly forces fire and move a distance Δ between rounds. This procedure is illustrated schematically in figure 4.

To calculate the average cost of this mission we have to consider two cases.

Case 1: $r_o/\Delta = N$ where N is an integer.

Case 2: $r_o/\Delta = N+\xi$ where N is an integer and $0<\xi<1$.

For case 1 a maximum of N+1 shells will be used, while for case 2 a maximum of N+2 shells will be used. The cost of the mission of exactly i shots depends on the value of i.

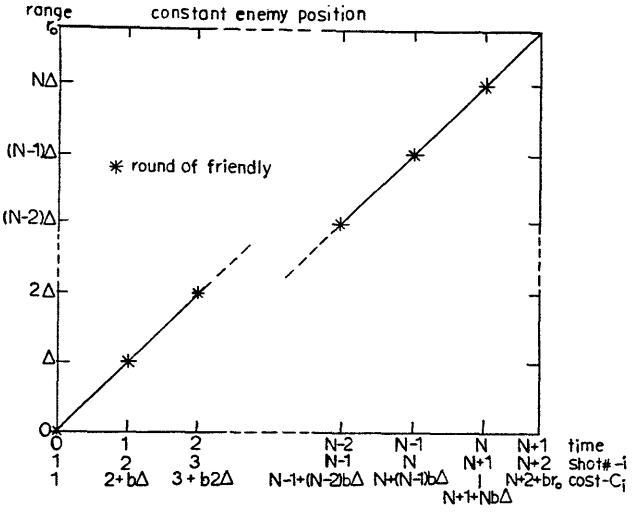


Figure 4 - Schematic representation of the move, shoot, move,...problem.

There are two cases

$$C_i = i + b(i-1)\Delta \qquad 1 \le i \le N+1 \qquad (16a)$$

$$C_i = N+2+br_o \qquad i=N+2 \qquad (16b)$$

The probability that a mission will end with the i'th shot is

$$P_i = P_M(r_o)P_M(r_o-\Delta)\ldots P_M[r_o-(i-2)\Delta] \cdot P_K[r_o-(i-1)\Delta] \qquad 1 \le i \le N+1 \qquad (17a)$$

$$P_i = P_M(r_o)P_M(r_o-\Delta)\ldots P_M[r_o-(i-2)\Delta] \qquad i=N+2 \qquad (17b)$$

The average cost of the mission is given by

$$C_{av} = \sum_{i=1}^{N+1} C_i P_i \qquad r_o/\Delta = \text{integer} \qquad (18a)$$

$$C_{av} = \sum_{i=1}^{N+2} C_i P_i \qquad r_o/\Delta \neq \text{integer} \qquad (18b)$$

A program was written to evaluate C_{av} for various cases so that the variation of average cost with the choice of step size could be studied. The results obtained from this program for a particular set of parameters are shown in figure 5. These results indicate that both a local and global minimum may exist, depending on the cost of moving. The choice of step size is not very critical as long as it is chosen reasonably.

14. Counter Positioning Problem - Enemy Returns Fire

This is the analysis of the case where the counter moves a distance Δ to range $r_o-\Delta$ and trades shots with the opponent until one is hit. In a battle of this kind it

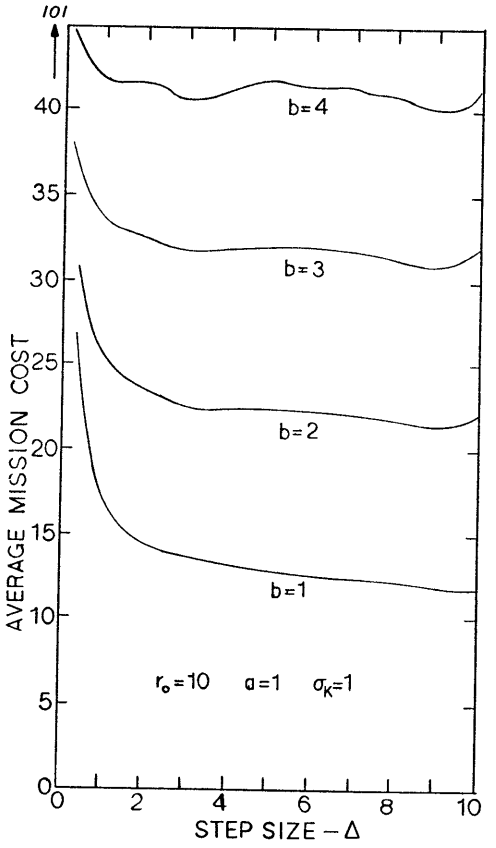


Figure 5 - Average mission cost as a function of step size for the move, shoot, move,...problem.

makes a considerable difference who fires the first shot. Therefore all costs were evaluated for both cases and the results shown for the average cost were obtained by equally weighting the two cases. In this problem there are three cost terms. The average cost of the shells is C_S, the average cost of moving is C_M and the average cost of being hit is C_H. The cost of the shells is one unit.

The expression for the average cost of the shells in the case where the enemy fires first is given by

$$C_{SE} = 1P_{M2}P_{K1}+2P_{M2}P_{M1}P_{M2}P_{K1}+ \ldots+j(P_{M2})^{j}(P_{M1})^{j-1}P_{K1}\ldots$$

$$= P_{K1}P_{M2}\sum_{j=1}^{\infty} j(P_{M2}P_{m1})^{j-1} \qquad (19)$$

which can be written in the closed form

$$C_{SE} = \frac{P_{K1}P_{M2}}{(1-P_{M1}P_{M2})^2} \quad \text{Enemy fires first} \qquad (20)$$

(Note that the cost of the shells is only calculated for the case where the friendly forces win, the reasoning being, that the cost of shells used before you are hit is of no consequence and is included in the cost of being hit.)

In a similar manner, we get the expression for the average cost if the friendly forces fire first

$$C_{SF} = \frac{P_{K1}}{(1-P_{M1}P_{M2})^2} \quad \text{Friendly forces fire first} \qquad (21)$$

We assume that the cost associated with being hit is D. The average cost of being hit for the case where the enemy fires first is derived as folows:

$$C_{HE} = D[P_{K2}+P_{M2}P_{M1}P_{K2}+P_{M2}P_{M1}P_{M2}P_{M1}P_{K2}+\ldots]$$

$$= DP_{K2}[1+P_{M1}P_{M2}+(P_{M1}P_{M2})^2+\ldots] \qquad (22)$$

which can be written in the closed form

$$C_{HE} = \frac{DP_{K2}}{1-P_{M1}P_{M2}} \quad \text{Enemy fires first} \qquad (23)$$

We note that the expression multiplying D is the probability of losing the engagement.

In a similar manner we can derive the expression for the cost of being hit for the case where the friendly forces fire first.

$$C_{HE} = \frac{DP_{K2}P_{M1}}{1-P_{M1}P_{M2}} \quad \text{Friendly forces fire first} \qquad (24)$$

The cost of moving is given by

$$C_m = b\Delta \qquad (25)$$

where b is the cost per unit distance moved. We note in passing that this cost can include the time delay involved in moving as well as actual costs, such as fuel.

It could be argued that the cost of moving should not be considered in cases where you are hit, as all fuel would be destroyed, however time lost may be important. The analysis could easily be modified to include this, if desired.

The total average cost of the mission when the enemy fires first is given by

$$C_E = C_{SE}+C_{HE}+C_m \qquad (26)$$

Similarly if the friendly forces fire first

$$C_F = C_{SF}+C_{HF}+C_m \qquad (27)$$

The average cost of the mission is given by

$$C = pC_E+(1-p)C_F \qquad (28)$$

where p is the probability that the enemy forces fire first.

A computer program was written to evaluate the average cost of a mission as a function of all the parameters involved. The results for one set of parameters are shown in figure 6 for p=0.5. These results confirm our intuitive feeling. In the case $(a_2/a_1)=5$, where the friendly counter has 5

times the accuracy of the enemy counter, the average cost of the mission is minimized by duelling at a range of 2 units. In the case where $(a_2/a_1)=0.2$, where the enemy counter has 5 times the accuracy of the friendly counter, the average cost is minimized by duelling at point blank range.

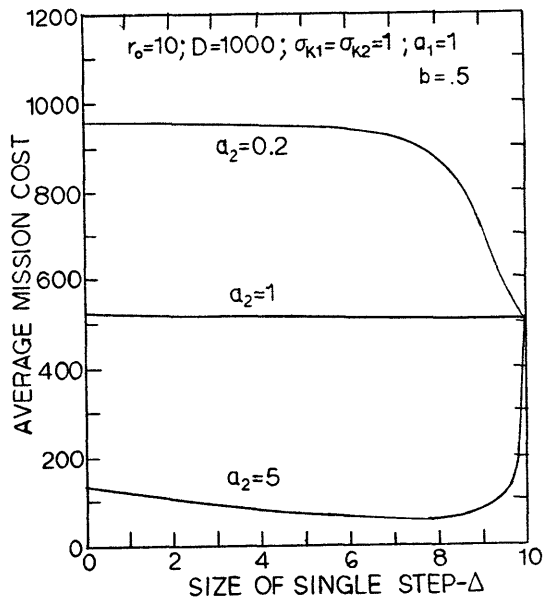


Figure 6 - Average mission cost as a function of step size for the case where the enemy returns fire.

Conclusion

Motivated by a desire to obtain a figure of merit called the capability function for the various blocks in a C^3 system, an analysis of some simple command problems was undertaken. The average mission cost was used as a criterion of command decision effectiveness, and it exhibited a minimum in most cases, when plotted as a function of the decision variable. This indicates that for these simple command decision problems and a minimum average cost decision criterion, an optimum solution exists. It also suggests that in evaluating a command decision, the ratio

$$\frac{\text{minimum average cost}}{\text{average cost associated with the decision}}$$

could be used as a figure of merit. It would be normalized to the range 0 to 1, with unity representing the optimum decision. Additional work must be done to see if these ideas can be generalized to more complex command problems.

References

1. I. Mayk, S. Rosenstark and J. Frank, "Analysis of C^3 Systems Based on a Proposed Canonical Reference Model." Proceedings of the 7th MIT/ONR Workshop on C^3 Systems, December 1984.

A MODEL FOR ONE-DIMENSIONAL IDENTIFICATION/COUNTER DYNAMICS

Andrew U. Meyer*

Israel Mayk#

*Department of Electrical Engineering
New Jersey Institute of Technology
Newark, New Jersey 07102

#US Army, Communications and Electronics Command (CECOM)
Communications / Automatic Data Processing Center (COMM/ADP)
Fort Monmouth, New Jersey 07703

ABSTRACT

With the aim of gaining an understanding of dynamic behavior in a class of C^3 systems, a primitive model is considered, which contains two important modules involving identification (ID) and counter (CO) dynamics respectively. Each module includes decision making and initiation of actions. The relatively simple model involves two opposing forces in terms of their (average) locations along one dimension. The model is discrete in the sense that observations, identification, decisions and initiation of actions are performed at fixed intervals.

Possible ID - initiated actions include re-maneuvering, remaining stationary and/or transfer to the counter (module). Possible CO -initiated actions also include remaneuvering and/or remaining stationary, as well as weapon engagement. Decisions on ID and CO initiated actions are made separately in each respective module. They are based on minimization of expected action cost and maximization of expected damage inflicted upon the opponent in the case of weapon engagement. Costs and damage, in turn, are affected by the observed values of both absolute and relative position and motion of the two parties and the probabilities of the identification (being true).

A simulation of the model has been constructed which produces responses in both tabular and graphical form. The model, in its basic configuration, is of the form of a 4 th order difference equation. Such equations are known to exhibit interesting behavior including chaotic motion under certain conditions. It is hoped that the establishment of the model and its simulation has provided the groundwork for the study of possible modes of dynamic behavior in this class of systems.

INTRODUCTION

The understanding of the dynamics of military combat situations is usually obscured by their complexity. Thus, C^3 researchers are seeking ways to describe some of the fundamental aspects of military system behavior d[Lawson 1979], [Alberts 1979], [Mayk, Rosenstark, Frank 1984].

Basically, military dynamics may be considered in terms of a hierarchy of dynamic systems, each representing some discrete level of command. The dynamics of the system below a given command level are usually expressed in terms of a model representing the average behavior of some relevant variables. For example, the attrition processes in certain battle situations may be modeled by Lanchester equations [Dolanski 1964], [Taylor 1981], [Schutzer 1982], [Wozencraft, Moose 1983], [Mayk, Rosenstark, Frank 1984], which describe the dynamics of the sizes (or strengths) of two opposing forces.

Though such overall models of given military situations constitute elements in higher-level control systems, they, in turn, may be broken down into more elementary (primitive) components [Mayk, Rosenstark, Frank 1984]. Obviously, the more is known about the structure and basic elements of a system or subsystem, the more information can be gained about its various possible modes of dynamic behavior. On the other hand, a "complete model", if such could ever be found, could easily obscure any understanding of system behavior and its underlying factors. Thus it is the philosophy governing this work to consider a "primitive" system model that contains only those of the features deemed relevant to military system dynamics and to study the basic modes of behavior that such a system may exhibit.

The most significant aspect of the command element in a C^3 system, at any hierarchical level, constitutes decision making. It is the objective of this work, to investigate the dynamics of a relatively simple system involving two opposing forces in terms of their (average) locations. In the interest of simplicity, for the initial model developed in this project, the locations are assumed to lie along one dimension. The extention to two dimensions will be undertaken in the future.

THE MODEL

A basic flow diagram of the (one-dimensional identification/counter) model is presented in Figure 1. It consists of two parts, "Friend" (F) and "Enemy" (E), both of which have the same basic structure but, in general, different parameter sets.

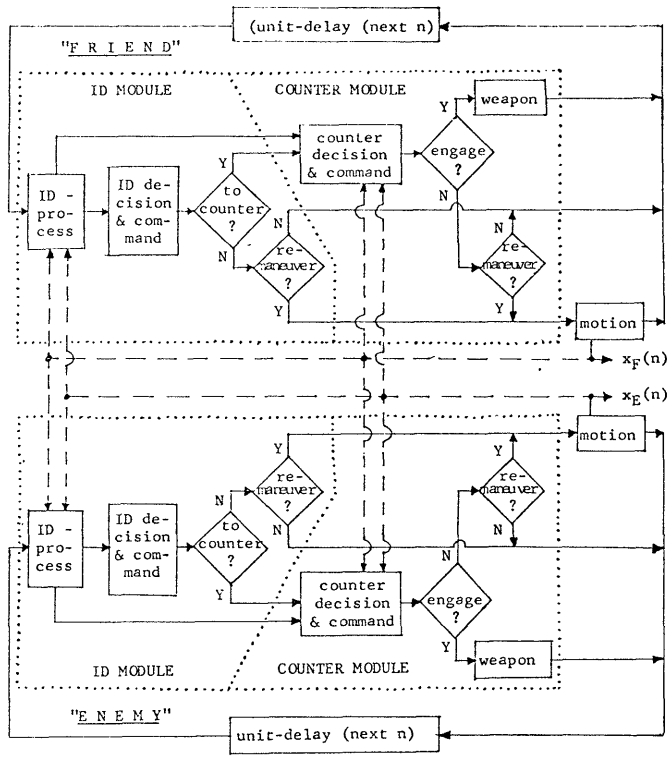


Figure 1. Basic flow diagram of the one-dimensional Identification/Counter model. Action flow is represented by solid lines. The (average) position variables, $x_F(n)$. and $x_E(n)$, for Friend and Enemy respecti vely, are represented by dashed lines.

Each party (F or E) has two basic modules, namely the identification (ID) module and the counter (CO) module. Each of these modules, in turn, involves decision and command on the basis of the information obtained during the identification process in the ID module.

The ID module and the CO module each can initiate action, the choice of which is limited to weapon engagement or remaneuvering of one's own position. The actions are chosen on the basis of observations from a strategy which, in turn, is selected on the basis of minimum net loss. Net loss consists of anticipated cost of both identification and counter actions minus the benefit received from any weapon engagement, each of which is computed on the basis of the observed positions of both parties. Following is a list of the basic assumptions made.

Basic Assumptions:

(1) The model is "discrete" in that observations (identification), computations, decisions and initiation of actions are performed at fixed time intervals (see Figure 1).

(2) The model is "one-dimensional" in space. Each party, Friend "F" or Enemy "E", is characterized along a line x by its average position, x_F or x_E respectively

(3) Change of position is limited to a fixed increment, $\pm XM_F$ or $\pm XM_E$ for friend (F) or enemy (E) respectively. [For example, with given time-interval and average velocity, XM_F = (average velocity of friend)*(time-interval)]

(4) In keeping with the philosophy of this project, namely to express C^3 systems in terms of "building blocks", attrition is assumed to be replenished, both in the ID and CO modules of the basic model. The CO module incorporates computation of damage inflicted on the opponent, from which attrition can be obtained. This, in turn, will modify the cost and damage factors in the opponent's decision and command modules. Thus, consideration of attrition-replenishment unbalance requires additional "coupling" modules between friend and enemy. The design of such modules is subject to future activity. Without additional attrition modules, the basic ID/CO model presented here (Figure 1) is "net-attrition-free" in the sense that all losses are beeing re-plenished. Of course, the losses are computed (see Fig.6).

Since the model structures (though not necessarily the parameter values) are the same for both "Friend" (F) and "Enemy" (E), the discussion to follow is presented in the more general terms "Self" (S) and "Opponent" (O). The locations at instant n then are referred to as xs(n) for "Self" and xo(n) for "Opponent".

Actions:

Actions can be initiated by either the ID or the CO module or by both. They are limited to the following:

Identification (ID) - Initiated Actions:

IA(1) = accept identification; go to counter (no motion involved)

IA(2) = remain stationary (do nothing at this time)

IA(3) = move toward opponent (in anticipation of obtaining a better identification later)

IA(4) = move away from opponent (too costly to do otherwise)

Counter (CO) - Initiated Actions:

CA(1) = engage weapon (no motion involved)

CA(2) = remain stationary (do nothing at this time)

CA(3) = move toward opponent (in order to increase the chance for inflicting greater damage on the opponent next time)

CA(4) = move away from opponent (too close for weapon engagement)

Note that the actions involving remaneuvering of position [(2), (3) and (4)] are the same for both ID and CO. The actions are determined on the basis of minimum estimated net cost which, in turn, is computed on the basis of the observations.

Observations:

The observations, incorporated in the ID module, are broken down into two basic sets of observation categories, namely,

(1) Opponent's absolute motion state observation $[\theta_1, \theta_2, \theta_3]$

(2) Relative motion state observation $[z_1, z_2, z_3, z_4]$

Each of these categories is further broken down into classes, three for the opponent's absolute motion state observation and four for the relative motion state observation. The classes are defined in terms of ranges of the (average) positions and their changes. The purpose of the classification is to facilitate the decision process. This, however, is not to say that decisions are made solely on the basis of occurance of observation classes. They are made on the basis of estimated net losses computed from the observed position and motion variables.

Opponent's Absolute Motion State Observation θ_i (i=1,2,3):

i=1: θ_1 = opponent moves toward self

i=2: θ_2 = opponent is "stationary" (within a specified range; see Fig. 2)

i=3: θ_3 = opponent moves away from self

Relative Motion State Observation z_j (j=1,2,3,4):

j=1: z_1 = no target identified

j=2: z_2 = target identified when relative distance between opponent and self decreased

j=3: z_3 = target identified when relative distance between opponent and self is stationary (within a cified range; see Fig. 4)

j=4: z_4 = target identified when relative distance between opponent and self increased

Observation Set: $y_{ij} = \theta_i z_j$

Example: $y_{31} = \theta_3 z_1$ = opponent moved away from self and no target is identified.

At each time-interval n, the probability of each of the above observation states (to be true) is computed from the present and previous positions of both parties. Gaussian type distributions are assumed, together with relevant parameters such as variances and threshold values. The following probabilities are computed:

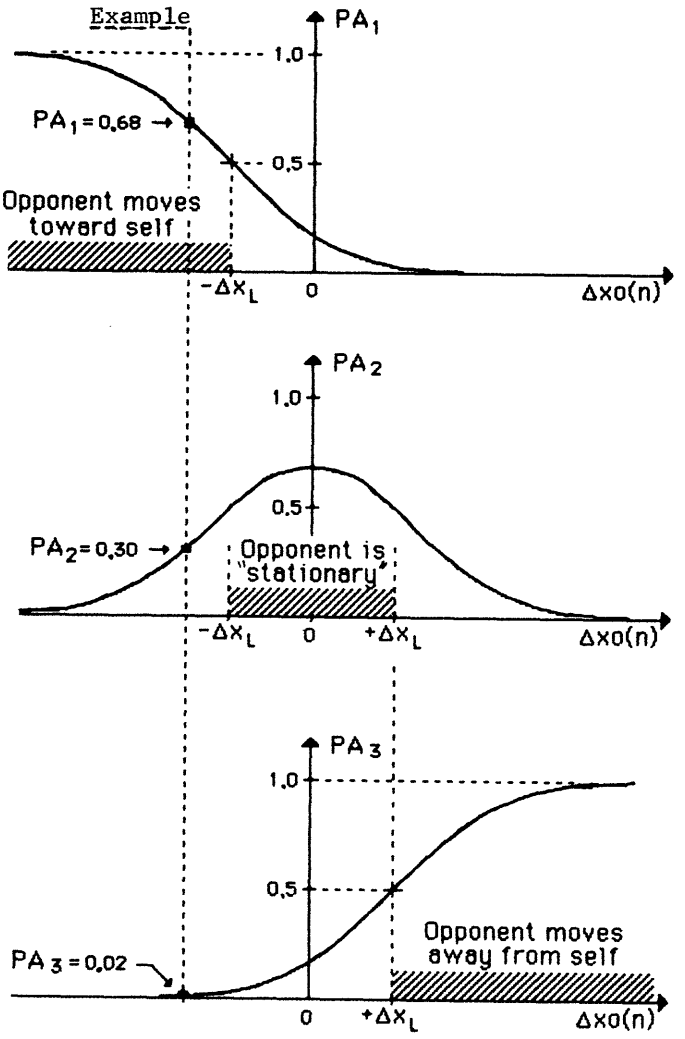


Figure 2. Probabilities of opponent's absolute motion state observation θ_i being true (i=1,2,3), as functions of the opponent's motion $\Delta xo(n) = xo(n)-xo(n-1)$. For motion within $\pm\Delta x_L$, the opponent is considered "stationary", where Δx_L is defined such that $PA_1(-\Delta x_L)=PA_3(\Delta x_L)=0.5$.

PA_i = $Prob\{\theta_i\}$
= probability of opponent's absolute motion state observation θ_i being true (i=1,2,3)

PR_{ij} = $Prob\{z_j | \theta_i\}$
=(conditional) probability of relative motion state observation z_j being true (j=1,2,3,4), given the oppopponent's absolute motion state θ_i (i=1,2,3)

These two probabilities depend on the opponent's motion and the relative distance (range) between opponent and self respectively. This is shown in Figures 2 and 3 whose captions give further descriptions. The computation of PR_{ij} requires consideration of another probability function, shown in Figure 4, namely

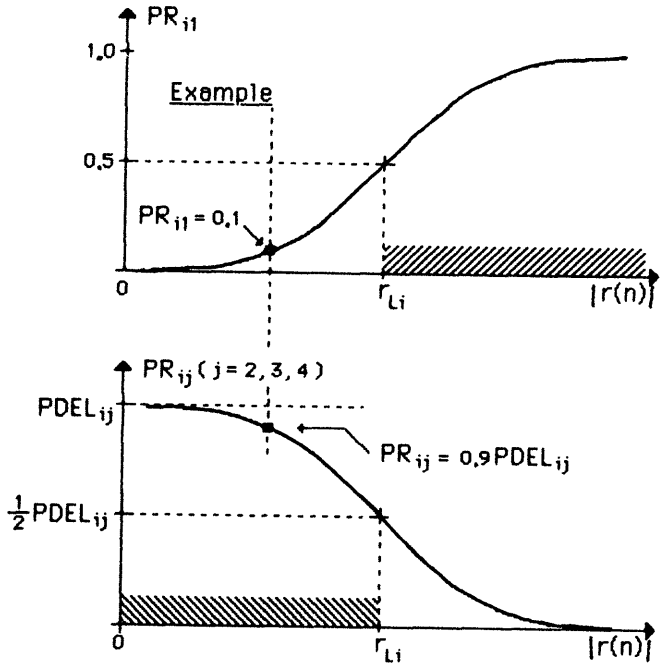


Figure 3. Probabilities of relative motion state observations z_j being true (j=1,2,-3,4), as functions of the range (relative distance) between the positions of Opponent, xo(n) and Self, xs(n), at intant n, $|r(n)|=|xo(n)-xs(n)|$. A "range limit" r_{Li}, depending on the absolute motion state θ_i, is defined such that the probability of no target being identified (j=1) is $PR_{i1}=0.5$. The probabilities of the other relative motion states being true (j=2,3,4) also depend on the (conditional) probability $PDEL_{ij}$, described in Figure 4.

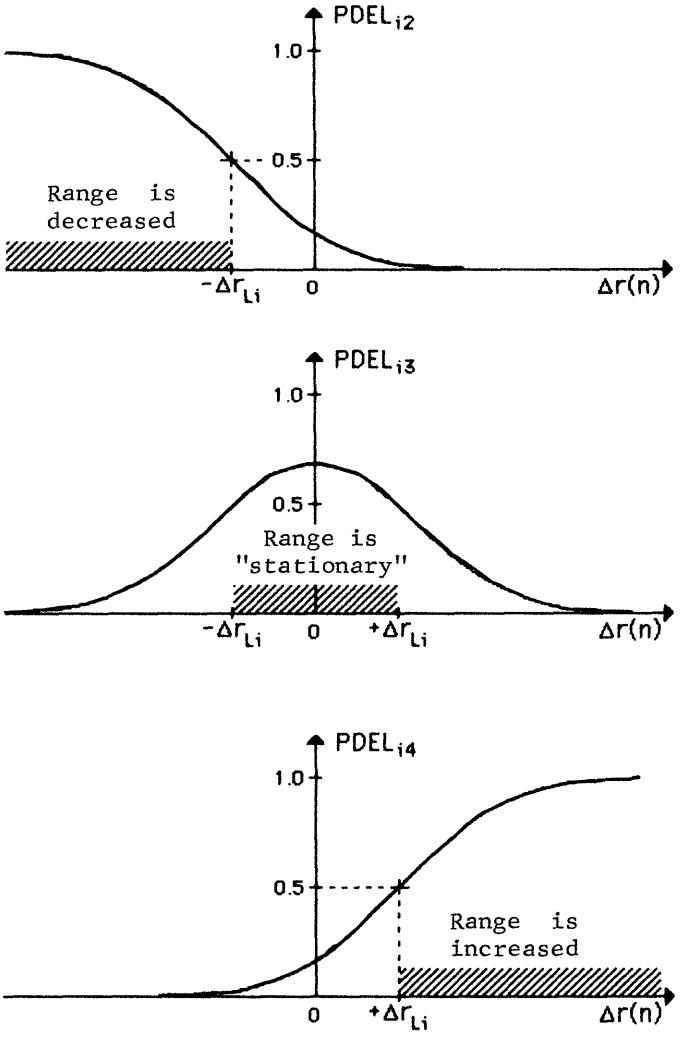


Figure 4. (Conditional) probabilities of relative motion state observations z_j being true (j=2,3,4) given that observation z_1 is not true, as functions of the range-change $\Delta r(n)=|r(n)|-|r(n-1)|$. Ranges (relative distances) within $\pm\Delta r_{Li}$ (depending on absolute motion state θ_i) are considered "stationary", where Δr_{Li} is defined such that $PDEL_{i2}(-\Delta r_{Li})=PDEL_{i4}(\Delta r_{Li})=0.5$.

$PDEL_{ij} = Prob(z_j|\theta_i, \text{not } z_1)$

(conditional) probability of relative motion state observation z_j being true (j=2,3,4) given that observation z_1 is not true (i.e., given that a target is identified and also given the opponent's absolute motion state θ_i(i=1,2,3)). Note that $PDEL_{i1}$ is not defined.

Note that the (conditional) probability of the relative motion observation state z_j being true depends on the probability of the opponent's absolute motion state θ_i which, in turn, however, does not depend on z_j In fact,

$$Prob(y_{ij}) = Prob(\theta_i z_j) = PA_i PR_{ij}$$

Cost of Actions

For both ID-initiated and CO-initiated actions it will be assumed that the action cost depends on the opponent's absolute motion state θ_i but not on the relative motion state z_j.

For example, ID action IA(2) [remain stationary, while continuing to detect] appears to require more effort (and therefore more cost) when the opponent is moving toward self (i=1) than if the opponent is stationary (i=2) or moving away from self (i=3). Though a similar, but perhaps not as strong, argument can be made with respect to relative motion state z_j, its effect (on cost) will be neglected in the interest of keeping the problem within reasonable bounds. Thus, cost is defined in terms of the following two matrices:

Identification Cost: **IC** = [IC(i,m)] (i=1,2,3; m=1,2,3,4)

Counter Cost: **CC** = [CC(i,m)] (i=1,2,3; m=1,2,3,4)

where IC(i,m) and CC(i,m) represent the cost of identification-initiated action IA(m) (m=1,2,3,4) and counter-initiated action CA(m) (m=1,2,3,4) respectively, given an opponent's absolute motion state θ_i (i=1,2,3).

Strategies (Assignment of Actions):

Both identification and counter actions are assigned upon occurance of certain observations. Recall that the definition of observation set y_{ij} implies the combination of two observations, namely of (a) the opponent's absolute motiion observation state θ_i (i=1,2,3) and (b) the relative motion state z_j (j=1,2,3,4). However, in the interest of less model complexity, actions will be assigned on the basis of relative motion state z_j only (recall, however, that action costs are assumed to depend only on absolute motion state θ_i).

For each ID- or CO- action, a number of strategies will be stored in the system, from which one will be chosen on the basis of minimum net average loss. Strategies are given in terms of matrices

ID - Strategy: $\mathbf{IS} = [IS(k,j)]$ (k=1,2,...,IK; j=1,2,3,4)

CO - Strategy: $\mathbf{CS} = [CS(k,j)]$ (k=1,2,...,CK; j=1,2,3,4)

where elements IS(k,j) or CS(k,j) represent the action assignment on the basis of relative motion state observation z_j under strategy k of IK or CK possible strategies stored in the system for ID- or CO- actions respectively.

Compensation (Win) Score and Counter Net Cost:

If counter-action CA(1) is initiated, which calls for weapon engagement, then the probability of damage inflicted on the opponent depends mainly on the relative distance between opponent and self. Again, a Gaussian type distribution is assumed, with appropriate reference values and standard deviations which, in turn, depend on the opponent's absolute motion state θ_i. This probability is multiplied by a "damage factor" D which depends on the weapon effectiveness. Thus, the Compensation (Win) Score (see Figure 5) is defined by

UC(i) = probable damage done to opponent in terms of loss

The compensation (win) score is deducted from the cost of counter action m=1 (weapon engagement). The net counter cost then is defined as

$$CCN(i,m) = \begin{cases} CC(i,1)-UC(i), & m=1 \\ CC(i,m), & m=2,3,4 \end{cases} \quad (i=1,2,3)$$

The compensation (win score will be used in the computation of optimum strategies and, upon issuance of appropriate action, be added up to form the cumulative (win) score.

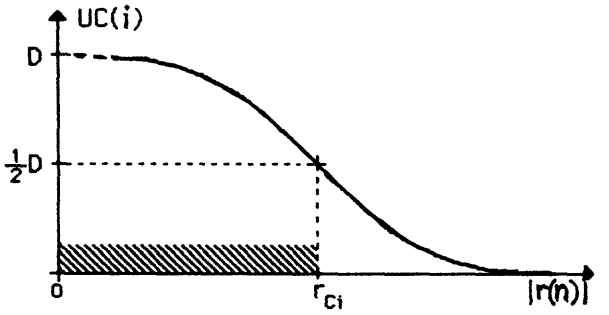


Figure 5. Probable damage done to Opponent upon counter action m=1, as a function of range (relative distance) between Opponent and Self, $|r(n)|=|xo(n)-xs(n)|$. Damage is measured in terms of cost (loss) to Opponent. A "damage factor" D and a "half-damage range" r_{ci} (depending on θ_i) are defined such that UC(i)=0.5D at $|r(n)| = r_{ci}$.

Average Losses and Optimum Strategies:

For each possible ID- or CO- initiated strategy k, the average loss IL(i,k) or CL(i,k) respectively, represents the estimated average cost considering all possible relative motion state observations (j=1,2,3,4), with their probabilities of occurance PR(i,j), as well as their consequent actions under strategies IS(k,j) or CS(k,j) respectively. Moreover, for counter strategies, the average counter net loss CNL(i,k) represents the estimated average net counter cost, i.e., counter cost minus counter benefit.

For each opponent's absolute motion state θ_i (i=1,2,3), one wants to select a strategy for which the average net loss is minimum. However, the selection of strategy is to be based on the observation of the opponent's absolute motion state θ_i. Thus, the average losses must further be weighted by the probabilities PA(i) of the observations of the absolute motion states θ_i being true. This, then results in the total average ID-loss TIL(k) and the total average net CO-loss TCNL(k) for the k th ID or CO strategy respectively.

The strategy k chosen will be that for which TIL(k) or TCNL(k) is minimum for ID or CO respectively. This is done during each time interval n.

The procedure described above follows the principles of decision theory [Chernoff and Moses, 1959]. In that sense, the opponent's absolute motion state θ_i (i=1,2,3) may be considered the "state of nature" and the relative motion state z_j (j=1,2,3,4) the "observation", on which decision on action is based.

N	POSITION X(N)	OPPON. ABSOL. ACTION STATE II	REL. MOT. OBSV STAT JJ	IDENT STRAT KMIN	IDENT ACTION I_ACTION	COUNTER STRAT CKMIN	COUNTER ACTION C_ACTION	AVERAGE IDENT LOSS LI	AVERAGE COUNTER LOSS LC	AVERAGE SYSTEM LOSS L(N)	CUMLTV SYSTEM LOSS LL(N)	AVERAGE SYSTEM COMP U	CUMLTV SYSTEM COMP UU(N)	NET SCORE UUN(N)
1	0.0	1	2	2	3	0	0	4.988	0.0	5.0	5.0	0.0	0.0	-19.99
1	1100.0	2	3	4	1	1	1	5.000	5.0	10.0	10.0	15.0	15.0	-5.00
2	150.0	2	1	2	3	0	0	4.767	0.0	4.3	9.3	0.0	0.0	-24.25
2	1100.0	1	2	8	4	0	0	5.000	0.0	5.0	15.0	0.0	15.0	0.00
3	300.0	3	1	2	3	0	0	6.963	0.0	7.0	16.2	0.0	0.0	-45.44
3	1175.0	1	3	1	1	1	1	10.000	10.0	20.0	35.0	14.2	29.2	-5.78
4	450.0	2	2	2	1	1	1	0.200	1.1	1.3	17.5	10.0	10.0	-36.73
4	1175.0	1	2	8	4	0	0	5.000	0.0	5.0	40.0	0.0	29.2	-20.76
5	450.0	3	1	2	3	0	0	6.963	0.0	7.0	24.4	0.0	10.0	-58.69
5	1250.0	2	3	1	1	1	1	5.000	5.0	10.0	50.0	15.0	44.2	-15.76
. . . . .														
21	1650.0	3	4	2	3	0	0	6.963	0.0	7.0	90.3	0.0	90.0	-164.55
21	1650.0	2	3	1	1	1	1	5.000	5.0	10.0	120.0	15.0	164.2	-85.74
22	1800.0	2	2	2	1	1	1	0.200	1.1	1.3	91.6	10.0	100.0	-155.82
22	1850.0	1	2	7	4	0	0	5.000	0.0	5.0	125.0	0.0	164.2	-110.24
23	1800.0	3	4	2	3	0	0	6.963	0.0	7.0	98.5	0.0	100.0	-177.78
23	1825.0	2	3	1	1	1	1	5.000	5.0	10.0	135.0	15.0	179.2	-105.24
24	1950.0	2	2	2	1	1	1	0.200	1.1	1.3	99.8	10.0	110.0	-169.05
24	1925.0	3	3	5	2	0	0	0.000	0.0	0.0	135.0	0.0	179.2	-115.24
25	1950.0	2	3	2	1	1	1	0.200	1.1	1.3	101.1	10.0	120.0	-175.32
25	1925.0	2	3	1	1	1	1	5.000	5.0	10.0	145.0	15.0	194.2	-120.24
26	1950.0	2	3	2	1	1	1	0.200	1.1	1.3	102.3	10.0	130.0	-181.98
26	1925.0	2	3	1	1	1	1	5.000	5.0	10.0	205.0	15.0	209.2	-125.24
27	1950.0	2	3	2	1	1	1	0.200	1.1	1.3	103.6	10.0	140.0	-187.85
27	1925.0	2	3	1	1	1	1	5.000	5.0	10.0	215.0	15.0	224.2	-130.24

Figure 6. Example of a response printout. For each time interval N, the first row re-fers to the friend and the second row to the enemy. Graphs of the positions and cumulative net scores for this case are shown in Figure 3.

The Cummulative Net-Score:

The cummulative net score for the Friend at instant n, UUNF(n), represents the addition of all compensation (win) scores achieved so far, UUF(n), minus all total losses incurred so far, LLF(n), minus the cumulative compensation (win) score of the enemy, UUE(n) [the latter representing loss to the friend], i.e.,

UUNF(n) = UUF(n)-LLF(n)-UUE(n) for Friend

and

UUNE(n) = UUE(n)-LLE(n)-UUF(n) for Enemy

SIMULATION

A simulation of the entire system has been designed and implemented on a VAX-780 computer. It provides both tabular and graphical output. For given past and initial positions, it computes the system dynamics in terms of position, losses, compensation (win) score, net score (compensation minus losses and other relevant variables. Figure 6 shows an example of a tabular printout which allows monitoring of the various variables involved in the system, each of which can also be displayed graphically. For the same case, Figure 7 presents two graphical displays, namely friend and enemy positions and cummulative net-score, each versus (time) intervals n.

In the example illustrated, friend and enemy are getting closer to each other. Weapon engagement commences at n=1. At n=24, the

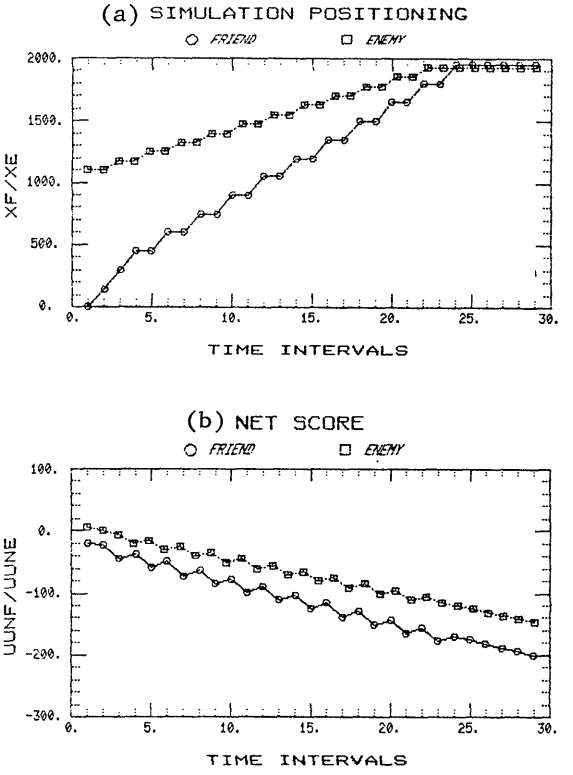


Figure 7. Graphical display of (a) positions and (b) cumulative net score, each for the same case as shown in the printout of Figure 2.

two parties are "on top of each other" which, of course, calls for interpretation of the situation and most likely for parameter modification.

It should be be emphasized that this model was designed to represent only some of the essential aspects of military C^3 systems in order to study their contributions to the system dynamics. The simulation will aid not only in these tasks but also in the design of modules to be added to the basic model to represent a variety of situations.

THE BASIC SYSTEM STRUCTURE

It can be shown that the system, in its basic form, has the structure of a forth-order nonlinear (vector) difference equation of the form

$$\mathbf{x}(n+1) = \mathbf{F}[\mathbf{x}(n)]$$

which may be represented by state variables $x_F(n)$, $x_E(n)$, $x_F(n+1)$and $x_E(n+1)$. Systems of this type can exhibit rather interesting behavior including bifurications of operational modes and "chaotic motion" [Meyer 1978], [Blackmore 1985]. Recognizing that this model represents certain elements involved in military combat situations, it becomes evident that basic research is required to obtain a better understanding of the dynamics of such systems. It is hoped that such research will lead toward the development of criteria on parameters for desired system performance and that the model presented in this paper, and its future extentions, will aid in these tasks.

ACKNOWLEDGEMENTS

This work was supported by the US Army - CECOM/ARO/BATTELLE under Contract No. DAAG 29-81-D-0100. Thanks are due to Messrs. L.D. Diedrichsen and J.M. Dressner of CECOM/CENSEI for their encouragement and the staff of the Battelle Research Triangle Office for their administrative support. The authors wish to acknowledge the fruitful discussions with their colleagues at NJIT, namely Drs. D.L. Blackmore, J. Frank, S. Rosenstark and J. Tavantzis. Particular thanks is due to Messrs. A. Verducci and J. Fyfe, both of whom wrote the initial simulation program and to Messrs. W. Au and E. Goldman for their continuing help and advise on problems involving the computer simulation.

REFERENCES

Alberts, D.S. (1979). C I Assessment: A proposed Methodology. Proceedings of Conference/Workshop on Qualitative Assessment of Command and Control Systems, October 1979 at the National Defense University, Washington, DC. Published by The MITRE Corporation, McLean, VA, January 1980, pp. 67-91.

Blackmore, D.L. (1985). The Mathematical Theory of Chaos. J. Computers and Math. Applications, to appear.

Chernoff, H. and L.E. Moses (1959). Elementary Decision Theory. John Wiley and Sons, New York, 1959, pp. 1-16.

Dolansky, L. (1964). Present State of the Lanchester Theory of Combat. Operations Research, vol. 12, pp. 344-358.

Lawson, J.S., Sr. (1979). The State Variables of a Command Control System. Proceedings of Conference/Workshop on Qualitative Assessment of Command and Control Systems, October 1979 at the National De-fense University, Washington, DC. Published by The MITRE Corporation, McLean, VA, January 1980, pp. 93-99.

Mayk, I., S. Rosenstark and J. Frank (1984). Toward a Canonical Reference Model for C^3 Systems. Proceedings of 7 th ONR/MIT Workshop on C^3 Systems, San Diego, CA, June 11-15, 1984. Published by Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1984.

Meyer, A.U. (1978). Some Fundamental Principles of Nonlinear Systems. Chapter 1 of Nonlinear System Analysis and Synthesis, vol. 1: Fundamental Principles, edited by J.K. Hedrick and H.M. Paynter. ASME, 1978.

Schutzer, D. (1982). Concepts and Thoughts Concerning Control Strategy for Conducting Information Warfare. Proceeding of 5 th MIT/ONR Workshop on C^3 Systems, Monterey, CA, August 23-27, 1982. Published by Laboratory for Information and Decision Systems, MIT, Cambridge, MA, December 1982, pp. 14-23.

Taylor, J.G. (1981). Battle-Outcome-Prediction Conditions for Variable Coefficient Lanchester-Type Equations for Area Fire. Journal of The Franklin Institute, vol. 311, no. 3, March 1981, pp. 151-170.

Wozencraft J.M. and P.H. Moose (1983). Lanchester's Equations and Game Theory. Proceedings of the 6 th MIT/ONR Workshop on C^3 Systems, December 1985, pp. 24-29.

A Stochastic Model of Lanchester's Equations

J.Tavantzis, S.Rosenstark and J.Frank

New Jersey Institute of Technology
Newark, N. J. 07102

Abstract

Lanchester's deterministic equations describing attrition between opposing forces are generalized to a stochastic model. The model is developed and it is shown that averages derived from it represent the original Lanchester equations except for a factor which is due to boundary conditions. A simple example is solved to illustrate the application of the model. The analysis is then carried out using the Laplace transform, which is done in order to simplify the algebra. The results of this latter analysis are also illustrated.

Introduction

Ever since the original work of Lanchester [1], many contributions have been made [2-4], extending the model describing attrition between two opposing forces to the case of a non-deterministic(stochastic one). These contributions are interesting but not in the mainstream of current work [5] where a measure of performance a utility function related to attrition is defined. With this in mind we develop a stochastic model generalizing Lanchester's deterministic law of conflict between two opposing forces.

Letting x(t) and y(t) be the size of the opposing forces with initial size x(0),y(0). Lanchester's equations covering self attrition and aimed fire are

$$\frac{dx}{dt} = -ax-by \qquad (1a)$$

$$\frac{dy}{dt} = -cx-dy \qquad (1b)$$

To get an understandig of the meaning of the equation we examine (1a). The negative rate of change of the x force due to self attrition, (such as diseases and accidents) is dependent on the size of the x force, and the constant a reflects the state of safety and health maintenance. The negative rate of change of the x force due to aimed enemy fire is dependent on the enemy force y and on their capability, and is given by the constant b.

Model

We construct a Markov process which models (1). Let X(t) and Y(t) be the random variables representing the sizes of the opposing forces at time t, we define the joint probability function, the probability that the sizes of the opposing forces are x and y respectively at time t, as

$$P_{x,y}(t) = P[X(t)=x,Y(t)=y] \qquad (2)$$

Making the usual assumptions for a death process [6], we define for a small interval of time Δt:

$\lambda(x,y)\Delta t$: the probability of a death occuring in force x when the sizes are x and y respectively, and

$\mu(x,y)\Delta t$: the probability of a death occuring in force y when the sizes are similarly x and y.

It follows that $1-(\lambda(x,y)+\mu(x,y))\Delta t$ is the probability that there are no deaths. In the model λ and μ correspond to the right hand side of (1) for x and y positive, i.e.

$$\lambda(x,y) = ax+by \qquad (3a)$$

$$\mu(x,y) = cx+dy \qquad (3b)$$

Now, to a small order of Δt, the probability that the sizes are x and y at time $t+\Delta t$ is equal to the probability that there were no deaths given that the sizes were the same at time t plus the probability there was a death in force x or y in that time interval. Expressing this algebraically we have

$$P_{x,y}(t+\Delta t) = \lambda(x+1,y)\Delta t P_{x+1,y}(t) + \mu(x,y+1)\Delta t P_{x,y+1}(t) + [1-\lambda(x,y)\Delta t-\mu(x,y)\Delta t]P_{x,y}(t)+o(\Delta t^2) \qquad (4)$$

The possible state transitions which are involved are summarized in figure 1.

*This work was supported by the U. S. Army, CECOM Center for Systems Engineering and Integration at Ft. Monmouth, N. J., under contract number DAAB07-83-K-K549.

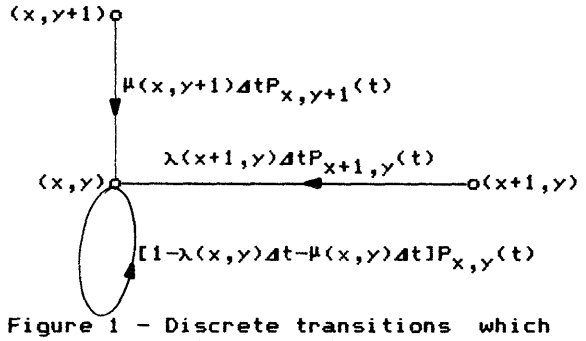


Figure 1 - Discrete transitions which lead to (4).

After subtracting $P_{x,y}(t)$ from both sides, dividing by Δt and taking the limit as Δt approaches zero, we obtain

$$\frac{d}{dt}P_{x,y}(t) = \lambda(x+1,y)P_{x+1,y}(t) + \mu(x,y+1)P_{x,y+1}(t) - [\lambda(x,y)+\mu(x,y)]P_{x,y}(t) \quad (5)$$

We assume we have an initial force distribution $P_{x,y}(0)$ with the usual property that

$$\sum_x \sum_y P_{x,y}(0) = 1 \quad (6)$$

Noting the natural boundary $x=0$ and $y=0$, this process cannot leave the first quadrant $x \geq 0$, $y \geq 0$. We impose the condition

$$\lambda(0,y) = 0 \quad (7a)$$

$$\mu(x,0) = 0 \quad (7b)$$

We now prove that (6) also holds at any time t

$$\sum_x \sum_y P_{x,y}(t) = 1 \quad (8)$$

This is done by showing that the derivative of the left side of (8) is zero and since it is initially one, it is one for all time. After differentiating the left side of (8) and substituting (5) we have

$$\sum_x \sum_y \frac{d}{dt}P_{x,y}(t) = \sum_x \sum_y \{\lambda(x+1,y)P_{x+1,y}(t) + \mu(x,y+1)P_{x,y+1}(t) - [\lambda(x,y)+\mu(x,y)]P_{x,y}(t)\} \quad (9)$$

By observing that

$$\sum_x \lambda(x+1,y)P_{x+1,y}(t) = \sum_x \lambda(x,y)P_{x,y}(t) \quad (10a)$$

$$\sum_y \mu(x,y+1)P_{x+1,y}(t) = \sum_y \mu(x,y)P_{x,y}(t) \quad (10b)$$

and by making use of condition (7), it follows from (9) that

$$\sum_x \sum_y \frac{d}{dt}P_{x,y}(t) = 0 \quad (11)$$

and (8) then follows directly from (11).

We next show that (5) is the probabilistic model of (1). Defining the expected value of the random variable X and Y at time t as

$$E(X) = \sum_x \sum_y xP_{x,y}(t) \quad (12a)$$

$$E(Y) = \sum_x \sum_y yP_{x,y}(t) \quad (12b)$$

respectively, we show that

$$\frac{dE(X)}{dt} = -E[\lambda(X,Y)] \quad (13a)$$

$$\frac{dE(Y)}{dt} = -E[\mu(X,Y)] \quad (13b)$$

We shall only derive (13a). Equation (13b) can be obtained in a similar manner. We differentiate E(X) in (12a) with respect to t

$$\frac{dE(X)}{dt} = \sum_x \sum_y x\frac{dP_{x,y}(t)}{dt} \quad (14)$$

and substitute (5)

$$\frac{dE(X)}{dt} = \sum_x \sum_y x\{\lambda(x+1,y)P_{x+1,y}(t) + \mu(x,y+1)P_{x,y+1}(t) - [\lambda(x,y)+\mu(x,y)]P_{x,y}(t)\} \quad (15)$$

In order to simplify (15) we make use of condition (7) to obtain

$$\sum_x x\lambda(x+1,y)P_{x+1,y}(t) = \sum_x x\lambda(x,y)P_{x,y}(t) - \sum_x \lambda(x,y)P_{x,y}(t) \quad (16)$$

and

$$\sum_y \mu(x,y+1)P_{x,y+1}(t) = \sum_y \mu(x,y)P_{x,y}(t) \quad (17)$$

Using (16) and (17), (15) reduces to

$$\frac{dE(X)}{dt} = -\sum_x \sum_y \lambda(x,y)P_{x,y}(t) = -E[\lambda(X,Y)] \quad (18)$$

We therefore have (13a). Since $\lambda(x,y)$ and $\mu(x,y)$ are given by (3) and (7), we have upon substitution in (13) (recalling that the expected value E is a linear functional)

$$P_{0,2} = \frac{2}{3} \qquad P_{1,2} = 0 \tag{31a}$$

$$P_{0,1} = \frac{1}{6} \qquad P_{1,1} = 0 \tag{31b}$$

$$P_{0,0} = 0 \qquad P_{1,0} = \frac{1}{6} \tag{31c}$$

From the above results it follows that the expected sizes of the forces y and x are

$$E(Y) = \frac{3}{2} \qquad E(X) = \frac{1}{6} \tag{32}$$

Based on this simple example, we make the following observations which will shortly be justified:

1. At the interior states (all points not located on the x or y axis), the probabilities tend to zero as t goes to infinity.

2. At the boundary (points located on the x or y axes), the probabilities go to nonzero values unless there is self attrition i.e. constant a or d is not equal to zero. In this case all states other than the (0,0) state are transient, that is

$$P_{x,y}(\infty) = \begin{cases} 1 & \text{if } x=0 \text{ and } y=0 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

which means that eventually complete annihilation occurs.

3. In the non self-attrition case when constants a and d are zero, the steady state solutions of (19), (when all time derivatives vanish,) reduces to

$$E(Y) = \sum_{y} yP_{0,y}(\infty) \tag{34a}$$

$$E(X) = \sum_{x} xP_{x,0}(\infty) \tag{34b}$$

which is exactly equation (23) when t tends to infinity.

Analysis

If the initial forces are large, then it is most difficult to find the various probabilities as functions of time. However, it turns out that the marginal steady states probabilities $P_{0,y}(\infty)$ or $P_{x,0}(\infty)$ can be easily calculated numerically. We can then find for the various forces the probabilities of eventually winning. The method is based on the use of Laplace transform and we define the Laplace transform which is defined by

$$F_{x,y}(s) = \int_0^{\infty} P_{x,y}e^{-st}dt \tag{35}$$

Applying the transform to (5) we have

$$\begin{aligned} sF_{x,y}(s)-P_{x,y}(0) = &-[\lambda(x,y)+\mu(x,y)]F_{x,y}(s) \\ &+\lambda(x+1,y)F_{x+1,y}(s) \\ &+\mu(x,y+1)F_{x,y+1}(s) \end{aligned} \tag{36}$$

and solving for the transform

$$F_{x,y}(s) = \frac{\lambda(x+1,y)F_{x+1,y}(s)+\mu(x,y+1)F_{x,y+1}(s)+P_{x,y}(0)}{s+\lambda(x,y)+\mu(x,y)} \tag{37}$$

We can then obtain $P_{x,y}(t)$ by taking the inverse Laplace transform using residue theory recursively. A numerical program could perhaps be used to obtain these. However the steady states probabilities can be found directly with the use of the final value theorem [7] which expresses the steady state value of a function in terms of its transform evaluated at zero, i.e.

$$P_{x,y}(\infty) = \lim_{s\to 0} sF_{x,y}(s) \tag{38}$$

We can see this relationship holds in our case since $F_{x,y}(s)$, given recursively by (37), has poles on the negative x-axis including possibly one at the origin. When $F_{x,y}(s)$ is transformed back to the time domain, all the poles with the exception of the one at the origin, will produce exponential functions in time with negative exponents. As t goes to infinity these terms will go to zero. Therefore, the only nonzero term remaining will be the one obtained from the pole at the origin. We thus have expression (38).

We next show that interior points (x and y greater than zero) are transient states, i.e.

$$P_{x,y}(\infty) = 0 \tag{39}$$

This is seen by first substituting (37) into (38) giving

$$\begin{aligned} P_{x,y}(\infty) = &\frac{\lambda(x+1,y)}{\lambda(x,y)+\mu(x,y)} \lim_{s\to 0} sF_{x+1,y}(s) \\ &+\frac{\mu(x,y+1)}{\lambda(x,y)+\mu(x,y)} \lim_{s\to 0} sF_{x,y+1}(s) \end{aligned} \tag{40}$$

Equation (40) is well defined since at interior points the sum $\lambda(x,y)+\mu(x,y)$ cannot be zero. Following one of the strings in the recursive formula, we reach an initial state (x_0,y_0) where (37) reduces to

$$F_{x_0,y_0}(s) = \frac{1}{s+\lambda(x_0,y_0)+\mu(x_0,y_0)}P_{x_0,y_0}(0) \tag{41}$$

And in particular,we have

$$P_{x_0,y_0}(\infty) = \lim_{s\to 0} sF_{x_0,y_0}(s) \tag{42}$$

$$\frac{dE(X)}{dt} = -aE(X)-bE(Y)+b\sum_{y} yP_{0,y}(t) \quad (19a)$$

$$\frac{dE(Y)}{dt} = -cE(Y)-dE(X)+c\sum_{x} xP_{x,0}(t) \quad (19b)$$

This is the equivalent of Lanchester's linear deterministic model (1). The extra terms on the right of (19) are due to the boundary conditions and more will be said about them later. The initial conditions for this system are given by

$$E(X)_0 = \sum_{x}\sum_{y} xP_{x,y}(0) \quad (20a)$$

$$E(Y)_0 = \sum_{x}\sum_{y} yP_{x,y}(0) \quad (20b)$$

We can now ask the following questions:

1. What is the probability that force x or y wins the conflict at time t ? In other words what is the value of

$$P(X \text{ wins at time } t) = \sum_{x} P_{x,0}(t) \quad (21a)$$

$$P(Y \text{ wins at time } t) = \sum_{y} P_{0,y}(t) \quad (21b)$$

2. What is the probability that force x or y eventually wins? That is

$$P(X \text{ eventually wins}) = \lim_{t\to\infty}\sum_{x} P_{x,0}(t) \quad (22a)$$

$$P(Y \text{ eventually wins}) = \lim_{t\to\infty}\sum_{y} P_{0,y}(t) \quad (22b)$$

3. What are the expected sizes of the remaining forces ?

$$E(X/X \text{ wins}) = \sum_{x} xP_{x,0}(t) \quad (23a)$$

$$E(Y/Y \text{ wins}) = \sum_{y} yP_{0,y}(t) \quad (23b)$$

We illustrate the above questions by looking at a simple example. Suppose the initial sizes of the opposing forces are x(0)=1 and y(0)=2. We simply have

$$P_{x,y}(0) = \begin{cases} 1 & \text{if } x=1 \text{ and } y=2 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

We also assume the aimed fire constants b and c are unity and the self attrition constants a and d are zero. Equations (5) become

$$\frac{d}{dt}P_{1,2}(t) = -3P_{1,2}(t) \quad (25a)$$

$$\frac{d}{dt}P_{0,2}(t) = 2P_{1,2}(t) \quad (25b)$$

$$\frac{d}{dt}P_{1,1}(t) = -2P_{1,1}(t)+P_{1,2}(t) \quad (25c)$$

$$\frac{d}{dt}P_{0,1}(t) = P_{1,1}(t) \quad (25d)$$

$$\frac{d}{dt}P_{1,0}(t) = P_{1,1}(t) \quad (25e)$$

$$\frac{d}{dt}P_{0,0}(t) = 0 \quad (25f)$$

Upon integration using the initial condition we have

$$P_{1,2}(t) = e^{-3t} \quad (26a)$$

$$P_{0,2}(t) = \frac{2}{3}(1-e^{-3t}) \quad (26b)$$

$$P_{1,1}(t) = e^{-2t}(1-e^{-t}) \quad (26c)$$

$$P_{0,1}(t) = \frac{1}{2}(1-e^{-2t})-\frac{1}{3}(1-e^{-3t}) \quad (26d)$$

$$P_{1,0}(t) = \frac{1}{2}(1-e^{-2t})-\frac{1}{3}(1-e^{-3t}) \quad (26e)$$

$$P_{0,0}(t) = 0 \quad (26f)$$

We note in passing that (8) is satisfied by the above expressions. For force y to win at time t, force x must not have any survivors. Hence

$$P(y \text{ wins at time } t) = \sum_{y=0}^{2} P_{0,y}(t)$$

$$= \frac{1}{2}(1-e^{-2t})+\frac{1}{3}(1-e^{-3t}) \quad (27)$$

Similarly

$$P(x \text{ wins at time } t) = \sum_{x=0}^{1} P_{x,0}(t)$$

$$= \frac{1}{2}(1-e^{-2t})-\frac{1}{3}(1-e^{-3t}) \quad (28)$$

To evaluate the event that force y or x eventually win, we let t go to infinity in the above expression. Therefore

$$P(y \text{ eventually wins}) = \frac{1}{2} + \frac{1}{3} = \frac{5}{6} \quad (29)$$

$$P(x \text{ eventually wins}) = \frac{1}{2} - \frac{1}{3} = \frac{1}{6} \quad (30)$$

The probability distribution as t goes to infinity is

Combining (40), (41) and (42), we obtain (39) that is the interior states are transient.

In the case of self attrition the same argument as above also shows that the boundary states are transient. The only exceptions being the origin, where

$$P_{0,0}(\infty) = 1 \qquad (43)$$

In the case of non self-attrition, $\lambda(x,y)+\mu(x,y)=0$ holds at the boundary. From (38) it follows

$$P_{x,y}(\infty) = \lambda(x+1,y)F_{x+1,y}(0) + \mu(x,y+1)F_{x,y+1}(0)+P_{x,y}(0) \qquad (44)$$

$F_{x+1,y}(0)$ and $F_{x,y+1}(0)$ are given recursively by (37) and corresponding to (41) we have

$$F_{x_0,y_0}(0) = \frac{1}{\lambda(x_0,y_0)+\mu(x_0,y_0)}P_{x_0,y_0}(0) \neq 0 \qquad (45)$$

The above thus shows that the boundaries are absorbing, i.e.

$$P_{x,y}(\infty) \neq 0 \qquad (46)$$

We return to the previous example with the following changes:

$$\lambda(x,y) = by \qquad \mu(x,y) = cx \qquad (47)$$

Using (37) and the same initial condition (24) as before, we find immediately that

$$F_{1,2}(0) = \frac{1}{2b+c} \qquad F_{1,1} = \frac{c}{(b+c)(2b+c)} \qquad (48)$$

and from (39)

$$P_{0,2}(\infty) = \frac{2b}{2b+c} \qquad (49a)$$

$$P_{0,1}(\infty) = \frac{bc}{(b+c)(2b+c)} \qquad (49b)$$

$$P_{0,0}(\infty) = 0 \qquad P_{1,0}(\infty) = \frac{c^2}{(b+c)(2b+c)} \qquad (49c)$$

Using (22) we now find that

$$P(Y \text{ eventually wins}) = \frac{2b^2+3bc}{(b+c)(2b+c)} \qquad (50a)$$

$$P(X \text{ eventually wins}) = \frac{c^2}{(b+c)(2b+c)} \qquad (50b)$$

If b and c are equal to one, then we have the prevous result

$$P(Y \text{ eventually wins}) = 5/6 \qquad (51a)$$

$$P(X \text{ eventually wins}) = 1/6 \qquad (51b)$$

Since initially force y is 2 and x is 1, we ask: How much more aimed fire capability should x have over y in order for x to win? That is, we want

$$P(X \text{ eventually wins}) > 1/2 \qquad (52)$$

or substituting (50b), we require that the Lanchester coefficients for aimed fire satisfy

$$\frac{c^2}{(b+c)(2b+c)} > 1/2 \qquad (53)$$

Upon solving the inequality we get

$$c > \frac{4}{\sqrt{17}-3}b \approx 3.56b \qquad (54)$$

We see that x should have roughly at least four times the fire power of y.

Conclusion

The theory derived in this paper is not only applicable to the small forces used in the examples but can be implemented numerically to analyze conflicts between large opposing numbers. The next step would be to calculate the probabilities of winning for large initial forces using various values for the aimed fire and self attrition constants. These results would then be tabulated and inferences drawn. This task is now being carried out.

References

1. Lanchester, F. W., "The principle of Military Concentration. The n-square Law," in Oliveira-Pinto, F., Connolly and B. W. "Applicable Mathematics of Non-Physical Phenomena." Ellis Horwood Ltd., Chichester; Halstead Press (John Wiley and Sons Inc), 1982, pp. 253-264.

2. Shultz, G. W. and Tsokos, C. P., "Stochastic Modelling-Lanchester's First Law," in Tsokos, C. P. and Thrall, R. M., "Decision Information," Academic Press, New York, 1979, pp. 185-211.

3. Taylor, J. G. "Recent Developments in the Lanchester Theory of Combat," in K. B. Haley, Operational Research 1978. North Holland Publishing Co., Amsterdam, 1979, pp. 773-806.

4. Dolansky, L. "Present State of the Lanchester Theory of Combat," Operational Research, No. 12, 1964, pp. 355-358.

5. Mayk, I., Rosenstark, S., Frank, J., "Analysis of C^3 Systems Based on a Proposed Canonical Reference Model," Proceedings 7th MIT/ONR Workshop on C^3 Systems, December 1984, pp. 45-53.

6. Bharucha-Reid, A. T., "Elements of the Theory of Markov Processes and their Applications," McGraw-Hill, New York, 1960.

7. Churchill, R. V., "Operational Mathematics," McGraw-Hill, New York, 1958.

AN ASSOCIATIVE MEMORY PARADIGM TO SOLVE C^3I PROBLEMS

Harold Szu, Sheldon Gardner, Leonard Sweet

Code 5709, Tactical EW Division,
Naval Research Laboratory, Washington, D.C. 20375

Abstract

One-dimensional conventional data arrangements (attributes, objects, values, ...) used in AI semantic networks can be usefully extended to higher dimensional matrix/tensor storage by means of outer products. Since outer products generate a global communications distributed matrix memory, an inner product of the matrix memory with a partial input data vector can yield a fault-tolerant associative recall. Such a write-by-outer-product and read-by-inner-product memory is the mathematical basis of fault-tolerant associative memory. Geometrically speaking, more degrees of freedom exist in volume storage where many more nearest-neighbor states are involved for alike grouping (intraclass) and unlike separation (interclass) than in one dimensional sequential storage. Such a distributed storage can be essential to solution of C^3 problems provided a proper knowledge representation in higher dimensions, such as associative memory, is implemented.

1. INTRODUCTION

Code 5709 at the Naval Research Laboratory conducts research and development in Counter Command, Control, Communications and Intelligence (Counter C^3I). In other words, how might our Naval forces best induce errors and delays in hostile tactical decisions. In order to accomplish this goal effectively, our forces need accurate and timely tactical intelligence so that our Counter C^3 actions against hostile C^3I can be optimally selected and orchestrated. Quite often, the processing and application of this tactical information requires rapid search and correlation of large and diverse sources of data. The associative memory search technique discussed in this paper could significantly enhance our ability to perform these functions.

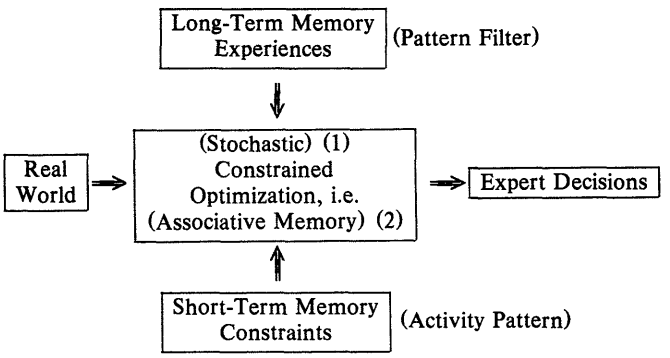
A simple (1-D) storage of ordered sets (attribute, object, value, etc.) is used to illustrate search techniques based on Boolean algebra (i.e. union, intersection, etc.). For example: "the color of an apple is red," we have (color, apple, red) and for example "The harvest of apples in October" we have (harvest, apple, October). Then we ask the question: What is the color of apples when harvested in October? The answer by the intersection of apple and the union of the rest of the entries gives, "The color of apples when harvested in October is red."

Such an elementary technique is quite useful and has been developed into AI deduction (closed set) and induction (open set). Recently, neural-network models of associative memory have been advanced with applications related to natural intelligence (NI) in contrast to AI-based software approaches.[4] As an illustration, we compare a simple 1-D storage and search technique with a search based on a 2-D storage and associative memory matrix recall technique. A pathological example in such an AI search is constructed to bring out the advantages of a higher dimensional storage and search method. The tradeoff between storage space and resolution of ambiguity is addressed.

Matrix models of associative memory may have applications in the field of system identification, multi-sensor data fusion and target tracking. Several of these applications are discussed in the framework of machine implementation of NI.

The authors would like to acknowledge the sponsorship of SPAWAR Codes 624 and 615.

Athans et. al.[3,6] at MIT proposed the following mathematical model of the real world decision making of which we add two explicit attributes Eq. (1) "stochastic" for the ability of hill-climbing out of the local minimum and Eq. (2) "associative" for the fault-tolerance in recalling experiences



Besides (1) (2), NI interpretations of the model are that constraints and short term memory act like *activity pattern* of which experiences and long term memory becomes a *pattern filter.* This 2D-picture language will become clear in Sect. 3 in the light of Sect. 2.

An associative memory[4] is nonlocally distributed among redundant interconnections (vector outer products to form rank 1 matrices). It is thus fault tolerant. Furthermore, associative memory recall is based on neighborhood arrangements in higher dimensions and a nonlinear (thresholding) operation upon the similarity measure (inner product as a distance measure in the higher dimension). Although the logic (Boolean algebra) is identically based on set theory and thus is independent of the set arrangement (i.e. 2-D can be represented as scanned 1-D), the neighborhood information will be totally different in N-D as opposed to 1-D. In fact, the number of nearest neighbor (n-n) states is proportional to the correlation length Δl to the power of N-dimensionality.

$$n.\text{-}n. \text{ correlated states} \cong (\Delta l)^N \qquad (3)$$

In Sect. 2, natural intelligence and associative memory are introduced and compared with conventional computer memory. In Sect. 3, applications to data fusion in search problems that are useful for decision aids are addressed. An ad hoc example of alias removal is used to indicate the advantage of 2D knowledge representation. In Sect. 4, dynamic resource allocation is discussed in terms of adaptive associative memory. The limitations of associative memory are discussed, and an attentive associative memory paradigm is introduced for C^3 systems.

2. NI ASSOCIATIVE MEMORY

A simple operational definition of natural intelligence (NI) based on neural network computing will allow us to represent (biological) associative memory. Then, it follows the crucial knowledge representation of NI is in a higher dimension, as opposed to a 1-D tuple storage in present day computer architecture for AI.

Qualitatively, the operational difference between AI and NI is mainly in so-called "experience." The former uses 1-D heuristic rules in

an expert system while the latter can be based on the equivalence model of 2-D mental pictures. For example, a spatiotemporal succession of events which may be imprecise or deliberately vague can lead to a vivid, sharp mental picture as an equivalence model in the context C of the union of those A,B events (by combination/contrast syntheses) (Fig. 1).

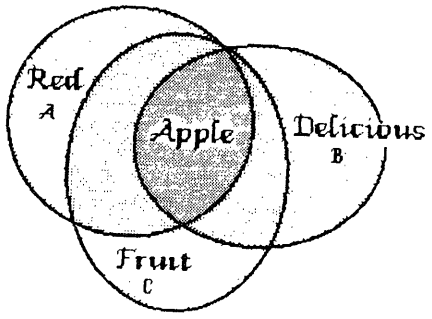


Figure 1. Equivalence model for an apple

If A=Green, B=Granny Smith, C=Fruit, then (green) apples follow as well. Consequently, both green and red, seemingly ambiguous, are bona fide attributes of apples. The natural ability of abstraction and of generalization beyond a red-delicious are essential to NI and the ability of learning. The main point is that N-D mental pictures rather than 1-D rules are often used in our thinking in terms of "neighborhood." A relative arrangement in N-D indicates a relative relationship, which, however, could become remotely disjointed in a 1-D lexicographical rearrangement.

A neural network model can be used to model NI which can be therefore thought of as network intelligence (NI). Neurological processing in the human brain has long been emulated by many researchers, notably the neuron model of McCulloch and Pitts, von Neumann in electronic digital computers, Widrow, Rosenblatt in the perceptron era, as well as Simon, Feingenbaum, Winston in the era of AI. It is our belief[8] that NI is ready to be simulated now and optically implemented in the coming decade, because the mathematical models of neural networks are ready.

Historically, Aristotle (370 B.C.), Lashley (fifties), Steinbuch and Caianiello (sixties); Morishita, Kohonen, Grossberg and Anderson (seventies); and Hopfield et. al. (eighties) have contributed to distributed associative memory networks. Basically, a neural network model implementing NI must possess three important attributes

$$\text{Nonlocal} \Longrightarrow \text{Outer product matrices} \tag{4a}$$

$$\text{Nonlinear} \Longrightarrow \text{Thresholding operations} \tag{4b}$$

$$\text{Nonstationary} \Longrightarrow \text{Iterative feedback} \tag{4c}$$

The interwoven complexity of these three organizational principles is furthermore reducible by means of simplifying mathematical tools indicated by three implication arrows and the respective approximations which will be discussed separately.

(1) Nonlocal Communication

The distributedness occurs through the vector outerproduct

$$\sum_l \vec{f}_l \, \vec{e}_l^{\,T} = \sum_l \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \overbrace{\cdots} \equiv [\mathrm{AM}] \tag{5}$$

where an event vector $\vec{e}$ is represented by a column matrix and a small size feature vector $\vec{f}$. After transposition $\vec{e}^{\,T}$ becomes a row vector such that each element of $\vec{e}$ broadcasts to and multiplies every element of $\vec{f}$ to produce the rectangular associative memory matrix [AM]. Such a global communication models the interconnects of neural networks which can produce fault tolerance. A simple example is the following auto-associative memory, which reveals the tradeoff between the memory capacity and the fault tolerance. Let us consider two training vectors $\vec{x}^{\,T} = (1,0,0)$ and $\vec{y}^{\,T} = (0,1,0)$. We construct 2-D knowledge representation by means of the [AM] formula (5).

$$[\mathrm{AM}] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} 100 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} 010 = \begin{bmatrix} 100 \\ 000 \\ 000 \end{bmatrix} + \begin{bmatrix} 000 \\ 010 \\ 000 \end{bmatrix} \tag{6}$$

Assume an input which may have a one bit mistake say, $\vec{\mathrm{in}}^{\,T} = (0,1,1)$. One can easily verify the associative recall given, by the inner product, produces the correct result $\vec{y}$.

$$\vec{\mathrm{out}} = [\mathrm{AM}]\, \vec{\mathrm{in}} = \vec{y}, \tag{7a}$$

since

$$\begin{bmatrix} 100 \\ 010 \\ 000 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{7b}$$

The geometrical meaning of the simple example is depicted in Fig. 2. By inspection, the input vector is closer to the memory vector $\vec{y}$ than to the other vector $\vec{x}$. Also, the simple example shows the near saturation of the memory capacity in 3 degrees of freedom. If the vector $\vec{z}$ had also been a training vector in the [AM], then the identical input can no longer produce the fault tolerant $\vec{y}$, but rather an ambiguous output result $\vec{y} + \vec{z}$. Researchers have extended uni-polar entry to bipolar, to N bits and to real numbers to investigate the tradeoffs between memory capacity and fault tolerance in such a distributed system. This brings us to the second point of nonlinear operation by means of thresholding.

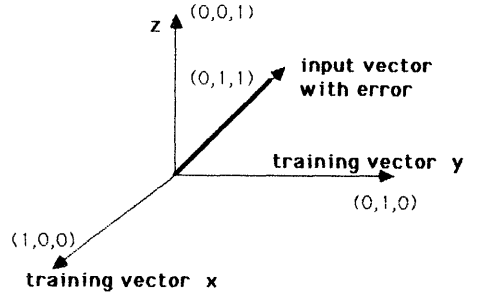


Figure 2. Fault tolerance and memory capacity in 3 degrees of freedom

(2) Nonlinear Operation

Any final decision requires a sharp nonlinear operation, yes or no. A thresholding operation is the simplest of such an operation and an adaptive level of the threshold is the next simplest. Apart from hysteresis oscillation loops (in the growing of magnetic domain walls, or optical bistable devices), a nonlinear operation in time can be approximated by a set of thresholding step operations at different levels and different times in the following sense.

$$f(t) \cong \sum_l f_l \text{ step } (t - t_l) \tag{8a}$$

$$\text{step } (t) = 0 \text{ or } 1, \text{ when } t < 0 \text{ or } t > 0 \tag{8b}$$

This fact allows us to successively implement the interwoven complex operation of (4a) and (4b) in terms of the third aspect (4c).

(3) Nonstationary Process

When the correlation function depends on two time points,

$$< \vec{x}\,(t + t_0)\, \vec{x}^{\,T}\,(t_0) > \; = [\mathrm{AM}(t|t_0)] \tag{8c}$$

we speak of a nonstationary process. Since a nonstationary process may be approximated by the Kolmogorov stationary increments of N-th order, one can approximate the N-th order finite difference scheme by a stationary approximation [AM(t)]. Then successive iterative feedback may be used piecewise to handle the necessary adaptation in a neural network.

In summary, the three "non"-principles of the natural (neural network) intelligence (NI) yield, among others, a model of the (biological) associative memory (AM). One shall differentiate AM from the conventional Location-Addressed Memory (LAM) and the Content-Addressable Memory (CAM) in that only AM is a distributed pattern of activities which describes a matrix of feedback connections. Both LAM and CAM are localized operations based on either the address-decoder key-to-lock concept or the directory circuit key-word comparison technique (shown in Fig. 3a). Introducing a hetero-associative formalism, one can generalize all three in the corresponding mathematical formula using the self-explanatory notations (Fig. 3b). The thresholding operation upon all shows that the third scheme auto-associative memory i.e.

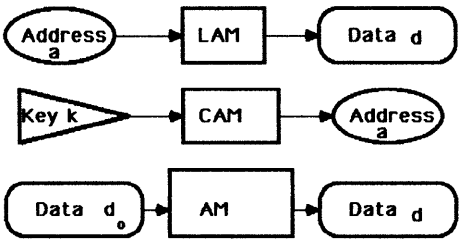


Figure 3a. Local and non-local memory

$$\text{LAM} \quad \{\sum[d\ a^T]\}a = \|a\|^2 d \qquad (9)$$

$$\text{CAM} \quad \{\sum[a\ d^T]\}k = (d,k)\,a \qquad (10)$$

$$\text{AM} \quad \{\sum[d\ d^T]\}d_0 = \|d_0\|^2 d \qquad (11)$$

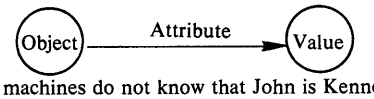
Figure 3b. Matrix memory representations

data-to-data association may have the strongest discriminating norm $||d_0||^2$. Currently, researchers are investigating the combined effect of dynamic range thresholding and spatial domain distributed outer product with regard to the net effect of fault-tolerance.

3. APPLICATION TO DATA FUSION IN SEARCH PROBLEMS

A fundamental problem in AI is knowledge representation. Conventional storage is a linear ordered set, e.g. the tuple (attribute, object, values) used in a semantic network. We wish to show that conventional AI knowledge representation cannot handle the aliasing problem (e.g. both John and Kennedy are identical) while 2-D [AM] storage can discover the alias by thresholding additions of those subliminal querry responses in terms of 2-D AM matrices. This occurs because the 2-D neighborhood relationship and fault tolerance endowed within the rank-one outer product formalism of AM help the data fusion.

In Fig. 4 we illustrate a semantic network where the node defines either the object or its value and the arrow indicates the attribute of the object.



(12)

Both NI and AI machines do not know that John is Kennedy.

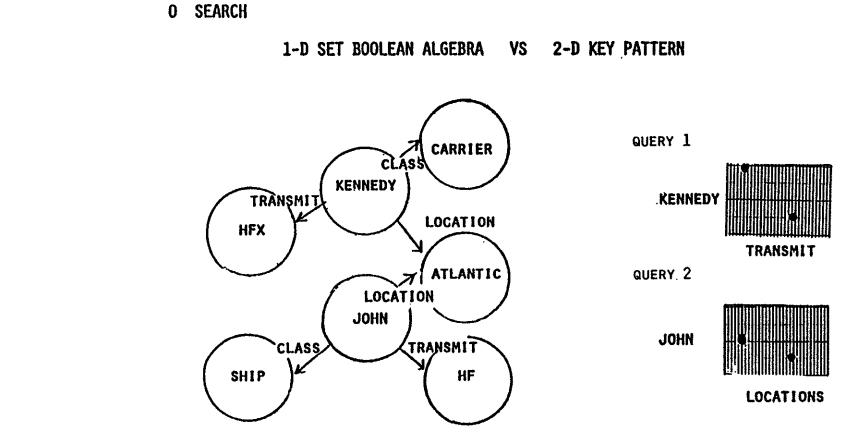


Figure 4. AI semantic network vs associative memory searching

The [AM] matrix is composed of three parallel submatrices which have respectively been coded by the attribute, the object and the value in some distributed fashion, and two queries have been asked-namely "Kennedy transmits?" and "John locates?" Answers based on the AI semantic network are straightforward but inadequate because the computer does not store the 2-D semantic network picture but rather the sequential 1-D tuple listed in Fig. 5. However, 2-D patterns of activity are coded in Fig. 5 for each tuple in the NI [AM] approach. Then, the [AM] recall becomes like a game of charade. That is, all patterns that are consistent with two key patterns in Fig. 3 will be added up in parallel in two subliminal matrix sums, which upon addition and thresholding yield the unique geolocation "Atlantic" and then alias "John" follows, as shown in Fig. 6.

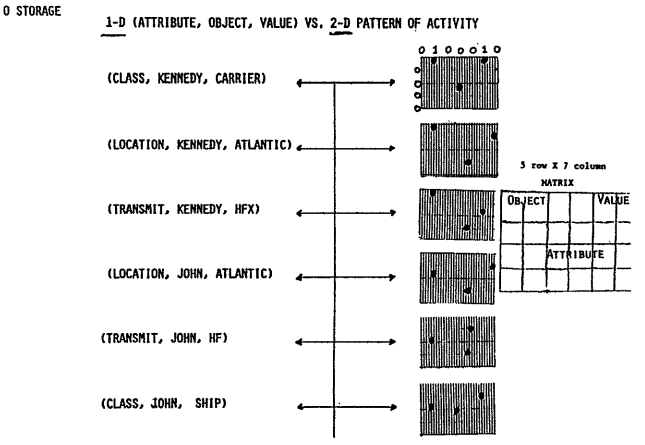


Figure 5. AI linear order set storage vs associative memory matrix storage

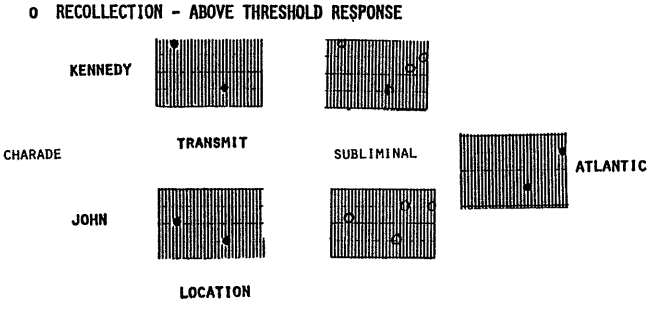


Figure 6. Associative memory searching by subliminal query response addition and thresholding

We note incidently that the danger of natural language when represented in 1-D tuple is well known. G. Pullum of UC showed that the missile off position (position, missile, silos) is dangerously mixed with the missile firing position (silos, missile, position). Obviously, pictures of off and on position are distinct. An ad hoc remedy based on human nature: "jumping to conclusion" and "sweeping generalization" is the *adduction logic*, namely "A $\Rightarrow$ B and B is true, therefore A is true," that has been incorporated into AI expert systems to overcome the alias problem.

4. DYNAMIC RESOURCE ALLOCATION FOR DISTRIBUTED SYSTEMS

When the total information is distributed in a partially shared and partially partitioned manner for each decision maker, we encounter the problem of dynamic resource allocation. Such a mixed (distributed/centralized) system has been considered by Kleinman[2] et al. at the University of Connecticut. They propose a local expertise matrix $[L]$ which maps a task attribute vector $\vec{a}(t)$, whose digital components represent damage reduction, strength, evasiveness, etc., into a resource requirement vector $\vec{r}(t)$ at time t.

$$\vec{r}(t) = [L]\,\vec{a}(t) \qquad (13)$$

The norm of the resource vector $\vec{r}(t)$ is constant. Since they also considered an integrated model of such a distributed dynamic decision (D^3) for arbitrarily distributed information (dyad, heirarchical, parallel), we

would like to extend their approach using NI associative memory. If both sets of input and output vectors of a global expert system are concatenated as rectangular matrices

$$[R(t)] = [\vec{r}_1(t), \vec{r}_2(t), \ldots \vec{r}_n(t)] \tag{14}$$

$$[A(t)] = [\vec{a}_1(t), \vec{a}_2(t), \ldots \vec{a}_n(t)] \tag{15}$$

then the global expert matrix $[\mathrm{AM}(t)]$ can be defined in terms of the generalized inverse of the attribute matrix $[A(t)]^{-1}$.

$$[\mathrm{AM}(t)] = [R(t)]\,[A(t)]^{-1} \cong \sum_{j=1}^{n} [\vec{r}_j(t)\vec{a}_j^T(t)] \tag{16a}$$

which can be simplified to a sum of outer-product cross-correlation matrices for a nearly orthonormal set of input attributes. Such a global definition takes both the advantage of the neighborhood relationship in $n-$D and the fault tolerance of associative recall similar to example Eq. (7) which in this case becomes fault-tolerant resource-allocation!

$$\vec{r}(t) = [AM(t)]\vec{a}(t) \tag{16b}$$

Moreover, the dynamic resource allocation problem is readily formulated, similar to a Kalman filtering orthogonal update, in terms of the vector version of Widrow's Adeline[4]

$$\frac{d}{dt}[\mathrm{AM}(t)] = \text{gain} \sum_j (\vec{r} - \vec{r}_j(t))\vec{a}_j^T(t) \tag{17a}$$

$$\vec{r}_j(t) = [\mathrm{AM}(t)]\,\vec{a}_j(t) \tag{17b}$$

where the r, a_j pairs are the desired output input pairs that we wish to train the global expert system matrix $[\mathrm{AM}(t)]$, $\vec{r}_j$ given by (17b) is the actual output and the difference $(\vec{r} - \vec{r}_j)$ drives the change in (17a). A parallel matrix version[7] of the Adeline can be also adopted

$$\frac{d}{dt}[\mathrm{AM}(t)] = \text{gain}\,([R(t)] - [\mathrm{AM}(t)][A(t)])[A(t)]^{-1} \tag{18}$$

which yields a matrix Langevin equation for stochastic input output

$$\frac{d}{dt}[\mathrm{AM}(t)] + \text{gain}\,[\mathrm{AM}(t)] = \text{gain}\,[R(t)][A(t)]^{-1} \tag{19}$$

Details remain to be developed for specific applications. In summary, a decision must be a consequence of a nonlinear operation, and a simple nonlinear operation is thresholding. Thus, for D^3, we furthermore impose the thresholding operation in the gain appearing in the previous Eqs. (17-19).

The associative memory in the simplest form Eq. (5) is not general enough for all C^3 applications. For example, hierarchical layered structures of a set of associative memories are conceivably needed to accomplish a complicated set of C^3 functions. Such a pyramid of neural nets exist in the human visual system along the neural pathway which could serve as a model of the complexity reduction problem vital to the C^3 community. We shall not go into details of this issue. Rather, we would like to point out the need of attentive associative memory (A^2M) of which an optical implementation has been introduced recently.[9] The attention amounts to introduce a nonuniformly weighted λ_l outer product in the "write mode."

$$\sum_l \lambda_l \vec{f}_l \vec{e}_l^T = [\mathrm{A^2M}] \tag{20}$$

and a nonuniformly weighted μ_{in} inner product in the "read mode"

$$\text{Recall} = \mu_{in}\,[\mathrm{A^2M}]\,\vec{e}_{in} \tag{21}$$

which takes into account a priori knowledge in the data domain. There remains to translate command and control into those nonuniform weights λ_l, μ_{in} which serve as the measure of priority.

REFERENCES

1. Proceedings of the 7th MIT/ONR Workshop on C^3 Systems, editor M. Athans and A. H. Levis, Dec. 1984 (LIDS-R-A37 Report) MIT.

2. The company volume 8th Workshop at MIT, June 24-28, 1985.

3. M. Athans, "System Theoretic Challenges in Military C^3 Systems," NR Rev. (1983) pp. 18-28 (Two/1983).

4. T. Kohonen, "Self-Organization and Associative Memory," Springer-Verlag, Berlin, New York, 1984.

5. J.N. Tsitsiklis, M. Athans, "On the Complexity of Decentralized Decision Making and Detection Problems," IEEE Trans, Aut. Cont. **AC-30**, No. 5, pp. 440-446, May 1985.

6. D.A. Castanon and D. Teneketzis, "Distributed Estimation Algorithms for Nonlinear Systems," IEEE Trans. Aut. Cont. **AC-30**, No. 5, pp. 418-425, May 1985.

7. H.H. Szu and R.A. Messner, "Adaptive Invariant Novelty Filters," appear in Proc. IEEE, 1985.

8. H.H. Szu, "Neural Network Models for Computing," submitted to Applied Optics, March 1986, special issue.

9. R.A. Athale, and H.H. Szu, "Optical Attentative Associative Memory," J. Opt. Soc. Am. Vol. 2 **A**, No. 11, 1985.

AN ALGORITHM FOR THE SOLUTION OF A CLASS OF DECENTRALIZED STOCHASTIC CONTROL PROBLEMS

Marcia P. Kastner and David A. Castanon

ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, Massachusetts 01803

1. Introduction

This technical paper describes some of the research results of the ONR-Navelex special focus program on distributed decisionmaking. This basic research program has two components: mathematics and engineering psychology. The former involves the development of mathematical models and optimization algorithms, while the latter involves the development and analysis of an experimental paradigm. This memo focuses on the mathematics component.

The objectives of the mathematics component of the research program described here is to develop optimal decision algorithms for a certain class of problems, namely, those involving finite-state, discrete-time, partially-observable Markov decision processes. The term "partially-observable" refers to uncertainty in the state due to noisy measurements. Furthermore, the processes are restricted to those whose sets of possible outputs at each observation and possible decisions at each time step are finite. The basic algorithm for the solution of dynamic decision problems is stochastic dynamic programming, which involves the temporal decomposition of a dynamic problem into a sequence of static problems and the characterization of the optimal performance and optimal strategies in terms of an optimal value function (i.e., "cost-to-go"). The most successful attempts to solve these problems have focused on the separation of estimation and control; that is, the design of the optimal control law can be separated from the design of the optimal estimate of the state. In particular, some problems satisfy the separation principle in that optimal decisions depend on past information only through the conditional probability distribution of the current state, given the past information [1]. Consequently, the optimal strategies, and, hence, the objective functions in the stochastic programming algorithm, have probability distributions as arguments. This poses no problem conceptually, but, since it leads to many possible strategies, may pose a major problem computationally.

The solution techniques for the optimal control of partially-observable Markov processes are surveyed in [2]. The state-of-the-art techniques for the class of Markov processes described above for the finite-horizon case is the Smallwood-Sondik Algorithm (SSA).* In [3], Smallwood and Sondik show that their formulation of the control problem satisfies the separation principle, so that optimal strategies are functions of the conditional probability of the state given past information. Hsu and Marcus [6],[7] have extended the results of [3] and [4] to a particular class of finite-state, decentralized stochastic control problems. In decentralized problems, there are multiple control stations, or decisionmakers (DMs), each with different information. Consequently, the information structure, that is, "who knows what," must be specified. The information structure in the Hsu-Marcus formulation (HMF) is the one-step-delay-sharing (OSDS) information pattern. In this case, all DMs share past data but retain as private their curremt measurements. Since the OSDS information pattern satisfies the separation principle [8], the optimal strategies are functions of conditional probability distributions, but with "past information" replaced by "shared information." In addition, the number of possible strategies grows exponentially with the number of DMs, greatly increasing the computational burden.

This paper describes two results in the mathematics component of the research program. The first result is an algorithm designed for large-scale problems, which approximates the SSA by reducing the dimensionality of the problem. A description of the SSA and the details of its approximation are found in Section 2. The second result is an algorithm extending the HMF for decentralized stochastic control problems with an OSDS information pattern to problems with a more general information structure, called sequential partitions, as defined by Yoshikawa and Kobayashi [9]. Section 3 summarizes the HMF for the OSDS information pattern and states the separation result in [9] for a class of problems with sequential partitions. Section 4 contains the details of the new algorithm extending the HMF. The conclusions are in Section 5.

2. Algorithms for the Solution of a Class of Markov Decision Problems

2.1 Problem Statement

The specific problem under consideration in this section is the single-person control of a system that can be modeled as a Markov process with the following features:

1. state $x(t)$ at each time t belongs to a finite set X of possible states, $X = \{\eta_1, \ldots, \eta_N\}$
2. decision problem has a finite horizon T
3. decision time intervals are discrete; $t = 1,\ldots,T$
4. measurement $y(t)$ at each time t belongs to a finite set Y of possible observations, $Y = \{\rho_1,\ldots,\rho_M\}$
5. observations are noisy measurements of the state and are described by a conditional probability distribution:

*Algorithms for the infinite-horizon case can be found in [4]-[5]. This memo will discuss only the finite-horizon case.

$$q_{i\ell}(t) = \Pr\{y(t) = \rho_\ell | x(t) = \eta_i\} \qquad (2\text{-}1)$$

6. decision (control, alternative) u(t) at each time t belongs to a finite set U of possible values, $U = \{\mu_1, \ldots, \mu_K\}$

7. changes in the state are characterized by a state transition conditional probability:

$$p_{ik}(u(t)) = \Pr\{x(t+1) = \eta_k | x(t) = \eta_i, u(t)\} \ . \qquad (2\text{-}2)$$

Figure 2-1 illustrates the sequence of events.

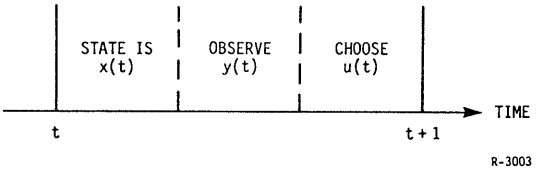


Figure 2-1. The Sequence of Events

It is shown in [3] that the current information about the state can be summarized by the information vector ξ, which is the conditional probability distribution of the state given the past measurements and decisions. More precisely,

$$\xi(t) = [\xi_1(t) \ \ldots \ \xi_N(t)]' \qquad (2\text{-}3)$$

where

$$\xi_i(t) = \Pr\{x(t+1) = \eta_i | \varepsilon(t)\} \qquad (2\text{-}4)$$

and $\varepsilon(t)$ = information available at time t. In this formulation, $\varepsilon(t) = \{y(t), u(t), \varepsilon(t-1)\}$. Since the calculation of $\xi(t)$ depends on past data only through $\xi(t-1)$, $\xi(t-1)$ is a sufficient statistic for the past history of $\varepsilon(t-1)$ [3]. Consequently, the decision problem satisfies a separation principle, so that the optimal control strategies belong to the set of strategies of the following form [3],[6]:

$$u(t) = \gamma(t, y(t), \xi(t-1)) \ . \qquad (2\text{-}5)$$

The problem is to find $\pi = [\gamma(1) \ \ldots \ \gamma(T)]'$ in the set Γ of admissible strategies so as to minimize expected total cost [6]:

$$\min_{\pi \in \Gamma} J(\pi) = E\left[\sum_{t=1}^{T} g_t(x(t), u(t))\right] \qquad (2\text{-}6)$$

where $g_t(x(t), u(t))$ is the cost at time t if the state is x(t) and the decision taken is u(t).

2.2 Smallwood-Sondik Algorithm

The problem just formulated is essentially the same as the one solved by the SSA in [3]. The crucial part of Smallwood and Sondik's result was their proof that, at each time step t, the dynamic programming expression for the optimal cost-to-go can be written as a piecewise-linear, concave function. More precisely, let $J_t^*(\xi(t-1))$ be the optimal cost-to-go at time t, parameterized by $\xi(t-1)$, the conditional probability distribution of the state. The sole parameter is $\xi(t-1)$, because (1) the separation principle is satisfied fied, and (2) the optimal cost J_t^* is averaged over all possible values of y(t), requiring brute-force enumeration of strategies with $\xi(t-1)$ fixed. It is shown in [3] that

$$J_t^*(\xi(t-1)) = \min_k \beta_k(t) \ \xi(t-1) \qquad (2\text{-}7)$$

where $\{\beta_1(t), \beta_2(t), \ldots\}$ are some N-dimensional vectors, computed by the SSA. The SSA also finds the optimal decision strategy associated with each minimizing β-vector. Figure 2-2 illustrates Eq. 2-7 using the two-dimensional simplex for ξ. In this case,

$$J_t^* = \begin{cases} \beta_1 \xi \ , & \xi \varepsilon R_1 \\ \beta_2 \xi \ , & \xi \varepsilon R_2 \\ \beta_3 \xi \ , & \xi \varepsilon R_3 \end{cases} \qquad (2\text{-}8)$$

where R_i is the ξ-region where β_i is optimal. These regions are outlined for the three-dimensional simplex in Fig. 2-3.

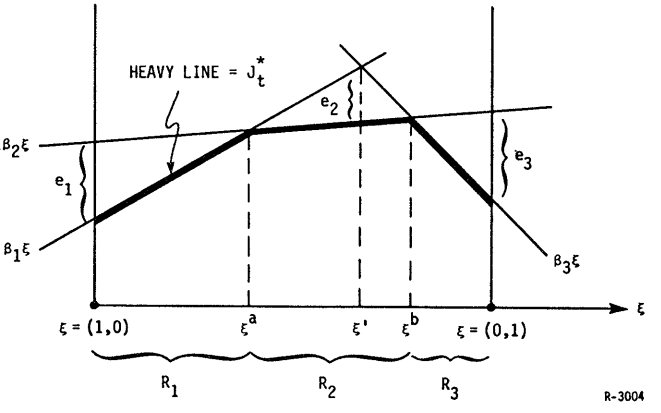


Figure 2-2. Two-Dimensional Example of Optimal-β Regions

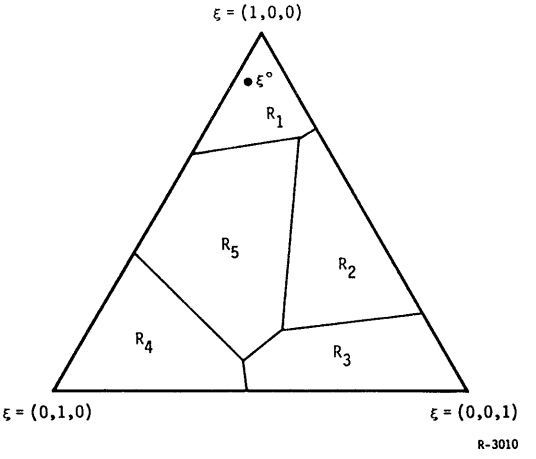


Figure 2-3. Three-Dimensional Example of Optimal-β Regions

The implementation of the SSA proceeds as follows (refer to Fig. 2-3). At each time t, an initial point ξ^o in the simplex is chosen, and its corresponding optimal strategy γ^* and β-vector, say β_1 in Fig. 2-3, are computed via the dynamic programming expression for J_t^* and knowledge of the β-vectors at time t-1. The SSA then uses a linear-programming (LP) procedure to calculate the boundaries of the region in the simplex for which β_1 is optimal. The values of the neighboring optimal β's can be computed so that the LP procedure can be used for each of them until the entire simplex is divided into optimal-β regions. Along the way, the SSA computes and stores in memory an associated γ^* for each β-region. Consequently, the optimal decision u^* at time t for some $\xi(t-1)$ and $y(t)=\rho_m$ can be found by solving Eq. 2-7 for the optimal β and letting $u^* = \gamma^*(\rho_m, \xi(t-1))$ for the γ^* associated with the optimal β.

2.3 Approximation to the SSA

As the state space, observation space, decision space, and time horizon increase in size, the number of linear pieces and the computational burden increase rapidly, especially since several LP problems are solved at each stage of the dynamic programming algorithm. To reduce the problem complexity to a fixed size, an approximation is presented in which the number of β-vectors at each time t does not exceed a specified maximum. If there are n β's, but a maximum of k is allowed, then (n-k) β's will be eliminated according to the criterion of minimizing the maximum "error," where "error" will be described below. The β's will be considered one-at-a-time in order to avoid the combinatorial problem of choosing k out of n.

Figure 2-2 illustrates the "error" e_i for the two-dimensional example, namely, the maximum difference between J_t^* with β_i eliminated and J_t^* with β_i included, in the region where β_i is optimal. That is, the error e_i is defined as

$$e_i = \max_{\xi \epsilon R_i} \left[\min_{k \neq i} (\beta_k \xi : \beta_k \text{ adjacent to } \beta_i) - \beta_i \xi \right] . \quad (2\text{-}9)$$

Then choose to eliminate the β with the minimum error*; that is, eliminate β_j if

$$j = \arg \min_i e_i \quad . \quad (2\text{-}10)$$

Figure 2-4 illustrates a possible new solution if β_5 were eliminated for the three-dimensional example. The eliminated region would be broken up into subregions, as shown by the dotted lines, where the adjacent β's would become optimal. Since J_t^* is the maximum of linear expressions, the errors e_i will occur at the intersections of these expressions, that is, at the corners of the new subregions. Consequently, the determination of e_i in Eq. 2-9 can be converted into an equivalent LP problem, as follows:

$$E_{ik} = \max_{\xi} (\beta_k \xi - \beta_i \xi) \quad (2\text{-}11)$$

$$\text{s.t.} \begin{cases} \beta_k \xi \leq \beta_\ell \xi, \text{ for all } \ell \neq k \text{ or } i, \\ \qquad \beta_\ell \text{ adjacent to } \beta_i \\ \text{and} \\ \text{the linear constraints defining} \\ \text{the } R_i \text{ region} \end{cases} \quad (2\text{-}12)$$

*The arguments t and t-1 from β and ξ, respectively, will be omitted here for simplicity.

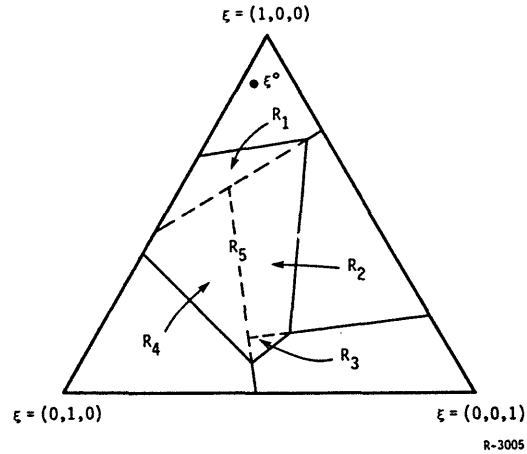


Figure 2-4. New Optimal-β Regions When β_5 is Eliminated

The complete algorithm to find j in Eq. 2-10 proceeds as follows:

Step 1: Solve the LP problem above for all k such that β_k is adjacent to β_i. Once a corner point ξ' is determined for some k, it is not necessary to solve the LP problem for those adjacent ℓ such that $\beta_k \xi' = \beta_\ell \xi'$.

Step 2: Let

$$e_i = \max_k E_{ik} \quad (2\text{-}13)$$

be the error associated with β_i.

Step 3: Repeat Steps 1 and 2 for all i. Then solve for j via Eq. 2-10.

Step 4: Go back to Step 1 and continue until n-k β's have been eliminated.

The resulting value of the optimal cost-to-go after the appropriate β's have been eliminated will be denoted $\hat{J}_t^*$, which is a lower bound on J_t^* For example, e_2 is the minimum error in Fig. 2-2, so that

$$\hat{J}_t^* = \begin{cases} \beta_1 \xi \quad , \quad \xi \epsilon R_1 \text{ and } \xi^a \leq \xi \leq \xi' \\ \beta_3 \xi \quad , \quad \xi \epsilon R_3 \text{ and } \xi' \leq \xi \leq \xi^b \quad . \end{cases} \quad (2\text{-}14)$$

Then the approximation error $J_t^* - \hat{J}_t^*$ can be easily calculated.

3. Decentralized Stochastic Control Problems Satisfying A Separation Principle

3.1 One-Step-Delay-Sharing Information Pattern

As mentioned in Section 1, Hsu and Marcus extended the SSA to decentralized stochastic control problems with the OSDS information pattern [6]. Assuming there are J DMs, the problem formulation differs from that in subsection 2.1 as follows:

For all j = 1, ... , J,

1. measurement $y_j(t)$ of DM_j at time t belongs to the finite set $Y_j = \{\rho_{j1}, \ldots, \rho_{jM_j}\}$

2. the conditional probability distribution for the measurements becomes:

$$q^{j}_{i\ell}(t) = \Pr\{y_j(t) = \rho_{j\ell} | x(t) = \eta_i\} \quad (3\text{-}1)$$

3. decision $u_j(t)$ of DM_j at time t belongs to the finite set $U_j = \{u_{j1}, \ldots, u_{jK_j}\}$

4. the state transition probability becomes:

$$p_{ik}(u(t)) = \Pr\{x(t+1) = \eta_k | x(t) = \eta_i, u(t)\}, \quad (3\text{-}2)$$

where

$$u(t) = [u_1(t) \ldots u_J(t)]' \quad . \quad (3\text{-}3)$$

Let

$I(t-1)$ = all data up to time t-1

$$= \{y_j(\tau), u_j(\tau) | 1 \leq \tau \leq t-1, 1 \leq j \leq J\} \quad (3\text{-}4)$$

$I_j(t)$ = information available to DM_j at time t, so that

$$u_j(t) = \gamma_j(t, I_j(t)) \quad . \quad (3\text{-}5)$$

Then the OSDS information pattern is defined as:

$$I_j(t) = \{y_j(t), I(t-1)\} \quad . \quad (3\text{-}6)$$

That is, at time t, I(t-1) is the shared information among all DMs, and $y_j(t)$ is the private information for DM_j. Varaiya and Walrand [8] and Yoshikawa [10] showed that this information structure satisfies a separation principle in that it is sufficient to search for optimal strategies among the strategies of the form

$$u_j^*(t) = \gamma_j^*(t, y_j(t), \xi(t-1)) \quad , \quad (3\text{-}7)$$

where

$$\xi_i(t-1) = \Pr\{x(t) = \eta_i | I(t-1)\} \quad . \quad (3\text{-}8)$$

They also showed that for n-step-delay-sharing patterns, defined as

$$I_j(t) = \{I(t-n), (y_j(\tau), u_j(\tau) | t-n+1 \leq \tau \leq t-1), y_j(t)\} \quad , \quad (3\text{-}9)$$

the separation principle does not necessarily hold for n>1.

In the HMF, the optimal cost-to-go at time t, illustrated in Fig. 3-1, is

$$J_t^*(\xi(t-1)) = E[g_t(x(t), \gamma^*(t))] + E[J_{t+1}^*(\xi(t))] \quad , \quad (3\text{-}10)$$

where

$$y(t) = [y_1(t) \ldots y_J(t)]' \quad (3\text{-}11)$$

$$\gamma(t) = [\gamma_1(t, y_1(t), \xi(t-1)) \ldots \gamma_J(t, y_J(t), \xi(t-1))]' \quad (3\text{-}12)$$

and the expectations are taken with respect to the observations y(t). The term $\gamma^*(t)$ is the vector of optimal strategies, determined by brute-force enumeration of all possible strategies with $\xi(t-1)$ fixed. It was proved in [7] that, as in the single-DM case, J_t^* is piecewise-linear and concave and so can be found via the SSA.

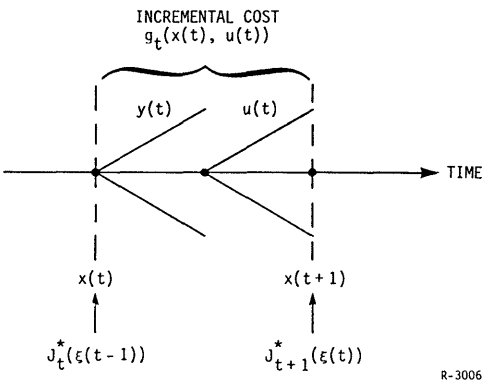


Figure 3-1. Temporal Representation of Dynamic Programming Algorithm

3.2 Sequential Partitions

Since the extension of the SSA to decentralized stochastic control problems requires a separation principle, one must look to information structures other than the n-step delay sharing for other applications of the SSA. One such information structure, described in [9] and [10], involves the decomposition of time into sequential periods, corresponding to a decomposition of the objective function. The problem defined in [9] is as follows:

$$x(t+1) = f(t, x(t), u(t), v_s(t)), \quad t=1,\ldots,T \quad (3\text{-}13)$$

$$y_j(t) = g_j(t, x(t), v_o(t)), \quad j=1,\ldots,J \quad . \quad (3\text{-}14)$$

The vectors x(1), $v_s(t)$, and $v_o(t)$ are finite-valued and random with a given a priori probability distribution. The objective function is

$$w = w(x(t), x(T+1), u_j(t) ; t=1,\ldots,T ; j=1,\ldots,J) \quad . \quad (3\text{-}15)$$

It is assumed that

$$\{y_j(t)\} \quad I_j(t) \quad \{y_j(t), I(t-1)\} \quad . \quad (3\text{-}16)$$

In other words, the j^{th} DM knows his current measurement but not the current measurements of the other DMs. The problem is to find $\gamma^* = \{\gamma_j(t); t=1,\ldots,T; j=1,\ldots,J\}$ such that

$$W(\gamma^*) \leq W(\gamma) \quad (3\text{-}17)$$

for all admissible γ, where $W(\gamma) = E(w|\gamma)$.

Definition 1: I(k) for some time $k \leq t$ is nested in $I_j(t)$ if I(k) can be recovered from $I_j(t)$ for any given γ. That is, knowing $I_j(t)$ implies knowing I(k).

Definition 2: The problem is said to have a sequential partition (s-partition) $\{t_1, t_2, \ldots, t_M\}$ such that

i) $1 \leq t_1 < t_2 < \ldots t_M < T$

ii) $I(t_m)$ is nested in $I_j(t)$, where

$$t_m = \max_i \{t_i \text{ is in the partition} | t_i < t\}$$

for all t=1,...,T and j=1,...,J.

In other words, an s-partition totally orders the time periods so that the information of each DM at each time in a period is available to all DMs at all times in subsequent periods. The example* in Fig. 3-2 illustrates the following information structure, where a dotted line from point (t,j) to point (t',j') means $I_j(t)$ is nested in $I_{j'}(t')$.

Example:

$I_j(1) = \{y_j(1)\}$, j = 1, 2

$I_1(2) = \{I(1), y_1(2)\}$

$I_2(2) = \{y_2(2)\}$

$I_j(3) = \{I_1(2), I_2(2), u_j(2), y_j(3)\}$, j = 1, 2

$I_j(4) = \{I_j(3), u_j(3), y_j(4)\}$, j = 1, 2

$I_j(5) = \{I_1(4), I_2(4), u_j(4), y_j(5)\}$, j = 1, 2

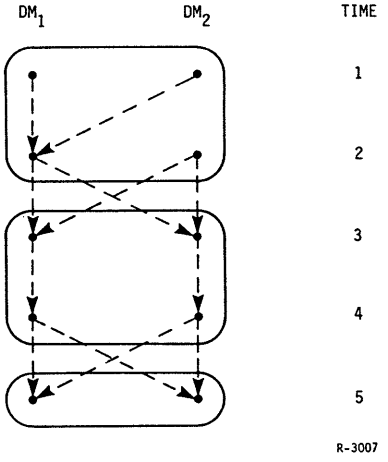


Figure 3-2. Example of an S-Partition

The s-partition is {2,4}. Figure 3-3 shows that the s-partition for the OSDS pattern is just the set of all time steps {1,...,T-1}, since information is nested at each stage.

*This example is a reordered version of one in [9].

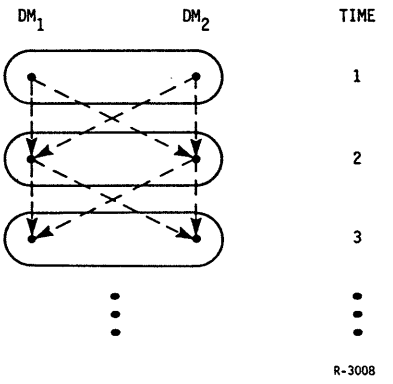


Figure 3-3. S-Partition for One-Step-Delay-Sharing Information Pattern

The separation result requires the following two conditions:

Condition A: The groups of random vectors $\{x(1), v(t);\ t=1,\ldots,t_1\}, \{v(t);\ t=t_1+1,\ldots,t_2\}, \ldots, \{v(t);\ t=t_M+1,\ldots,T\}$ are mutually independent.

Condition B: The cost function has the form

$$w = \sum_{m=0}^{M} w_m\big(x(t_m+1),\ldots,x(t_{m+1}+1), u(t_m+1),\ldots,u(t_{m+1})\big) \qquad (3\text{-}18)$$

where $t_0 = 0$ and $t_{M+1} = T$.

In other words, the s-partition is characterized by a total decomposition of the problem; not only the information sets, but also the random variables and cost function are partitioned. Define the control strategies as

$$u_j(t) = \gamma_j\big(t, \Delta I_j(t), \xi(t_m)\big) \qquad (3\text{-}19)$$

where

$$\Delta I_j(t) = I_j(t) - I(t_m) \qquad (3\text{-}20)$$

= information of DM_j from time $t_m + 1$ to t, where t_m is as defined in Definition 2.

Note that $\Delta I_j(t) = \{y_j(t)\}$ when $t = t_m+1$. The separation result can now be stated:

Theorem: If a problem has an s-partition satisfying conditions A and B, then it is sufficient to search for optimal strategies among strategies of the form in Eq. 3-19.

Proof: See [9].

The s-partition appears to be a generalization of the OSDS information pattern. For OSDS, each time period is a single time step. Since a problem satisfying conditions A and B also satisfies a separation principle, its dynamic programming formulation should be similar to the HMF. The next section gives a detailed description of an algorithm extending the HMF and the approximate SSA to decentralized stochastic control problems that admit sequential partitions satisfying Conditions A and B.

4. Algorithm for a Class of Decentralized Stochastic Control Problems with Sequential Partitions

4.1 Problem Statement

Presently, the only algorithm for computing the optimal control of Markov processes in a decentralized stochastic control problem is the SSA applied to the HMF for the OSDS information pattern. This section extends the algorithm to a class of problems with s-partitions. The precise problem under consideration is the one in [9], defined by Eqs. 3-13 through 3-17 with s-partitions satisfying Conditions A and B, thereby satisfying the separation theorem. In order to extend the HMF to this problem, the state and observation dynamics in Eqs. 3-13 and 3-14 must be rewritten in terms of Eqs. 3-2 and 3-1, respectively. Thus, it is assumed that f and g_j are "invertible" functions in that:

$$p_{ik}(u(t)) = \Pr\{v_s(t) = \text{the appropriate value to make } x(t+1) = \eta_k \text{ when } x(t) = \eta_i \text{ and } u(t) \text{ is given}\}$$

$$q_{i\ell}{}^{j}(t) = \Pr\{v_o(t) = \text{the appropriate value to make } y_j(t) = \rho_{j\ell} \text{ when } x(t) = \eta_i\}.$$

As an illustration of the algorithm, Fig. 4-1 shows a temporal representation of the example in subsection 3.2. As mentioned earlier, the s-partition is {2,4}. In order to use the decomposition technique associated with the separation result, the optimal cost-to-go $J_t{}^*$ is calculated at the start of each time period, as opposed to each time step, as defined by the s-partition.

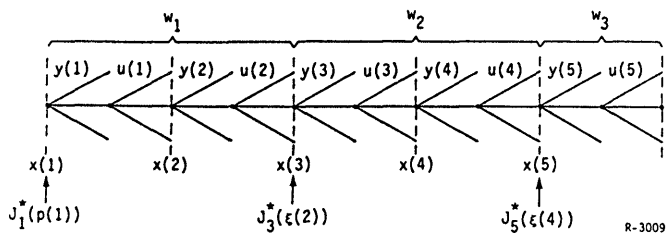


Figure 4-1. Temporal Representation of Example in Fig. 3-2.

In the example, this means that $J_1{}^*\{p(1)\}$, $J_3{}^*\{\xi(2)\}$, and $J_5{}^*\{\xi(4)\}$ are calculated, where p(1) is the a priori probability distribution of x(1). Thus, the strategies are written as:

$$u_j(1) = \gamma(1, I_j(1), p(1)) = \gamma_j(1, y_j(1), p(1)) \quad (4\text{-}1a)$$

$$u_j(2) = \gamma_j(2, I_j(2), p(1)) \quad (4\text{-}1b)$$

$$u_j(3) = \gamma_j(3, \Delta I_j(3), \xi(2)) = \gamma_j(3, y_j(3), \xi(2)) \quad (4\text{-}1c)$$

$$u_j(4) = \gamma_j(4, \Delta I_j(4), \xi(2)) \quad (4\text{-}1d)$$

$$u_j(5) = \gamma_j(5, \Delta I_j(5), \xi(4)) = \gamma_j(5, y_j(5), \xi(4)) \quad (4\text{-}1e)$$

4.2 Solution Technique

In the previous example, the expectation of $J_5{}^*(\xi(4))$ is taken over all possible values of $y_1(5)$ and $y_2(5)$. For $J_3{}^*(\xi(2))$, the expectation is over all possible values of $\Delta I_j(3)$ and $\Delta I_j(4)$, j=1, 2. However,

$$\Delta I_j(3) = \{y_j(3)\} \quad (4\text{-}2)$$

$$\Delta I_j(4) = \{y_j(3), y_j(4), \delta I_j(4)\} \quad (4\text{-}3)$$

where

$$\delta I_j(t) \triangleq \Delta I_j(t) - \Delta I_j(t-1) - \{y_j(t)\} = I_j(t) - I_j(t-1) - \{y_j(t)\} \; . \quad (4\text{-}4)$$

In other words, $\delta I_j(t)$ is DM_j's information gathered at time t, excluding the measurement at time t. That is, $\delta I_j(t)$ contains data only from time t_m+1 to time t-1. In the example, $\delta I_j(4)$ contains data only for t=3. In the brute-force enumeration solution technique for finding $J_3{}^*$, the strategies $\gamma_1(3)$ and $\gamma_2(3)$ are specified inside the expectation, so that $\delta I_j(4)$ is specified. Thus, the expectation over $\Delta I_j(3)$ and $\Delta I_j(4)$ is just the expectation over $y_j(3)$ and $y_j(4)$, j = 1,2. In general, for a time period $\{t_m+1,\ldots,t_{m+1}\}$, the expectation over $\Delta I_j(t_m+1),\ldots,\Delta I_j(t_{m+1})$ reduces to the expectation over $y_j(t_m+1),\ldots,y_j(t_{m+1})$ for fixed strategies $\gamma_j(t_m+1),\ldots,\gamma_j(t_{m+1})$, j = 1,...,J.

The optimal cost-to-go at time t_m+1 can be written as

$$J_{t_m}{}^*{}_{+1}(\xi(t_m)) = E[\text{incremental cost } w_m] + E[J_{t_{m+1}}{}^*{}_{+1}(\xi(t_{m+1}))] \; , \quad (4\text{-}5)$$

except for the final time T, in which case the last term is omitted. Using straightforward dynamic programming techniques the expected incremental cost for the time period $\{t_m+1,\ldots,t_{m+1}\}$ can be written as a recursive expression (with J=2 for simplicity and M = number of time steps in the period):

$$E[w_m] = \sum_{i_1=1}^{N} \left\{ \sum_{k_1=1}^{M_1} \sum_{\ell_1=1}^{M_2} [\tilde{g}_{t_m+1} + G_2]\, q_{i_1 k_1}{}^{1}(t_m+1)\, q_{i_1 \ell_1}{}^{2}(t_m+1) \right\} \xi_{i_1}(t_m) \quad (4\text{-}6)$$

where

$$G_n = \sum_{i_n=1} p_{i_{n-1} i_n}(u(t_m+n-1)^*) \sum_{k_n=1}^{M_1} \sum_{\ell_{n-1}}^{M_2} [\tilde{g}_{t_m+n} + G_{n+1}]\, q_{i_n k_n}{}^{1}(t_m+n)\, q_{i_n \ell_n}{}^{2}(t_m+n) \; .$$

$$n = 2,\ldots,M \quad (4\text{-}7)$$

$$G_{m+1} = 0 \quad (4\text{-}8)$$

$$t_{m+M} = t_{m+1} \quad (4\text{-}9)$$

and

$$\tilde{g}_{t_m+n} = g_{t_m+n}{}^*[\eta_{i_n}, \gamma_1{}^*(t_m+n, y_1(t_m+1) = \rho_{1k_1},\ldots, y_1(t_m+n) = \rho_{1k_n}, \delta I_1(t_m+n) = D_1, \xi(2)), \gamma_2{}^*(t_m+n, y_2(t_m+1) = \rho_{2\ell_1},\ldots, y_2(t_m+n) = \rho_{2\ell_n}, \delta I_2(t_m+n) = D_2, \xi(2))] \; . \quad (4\text{-}10)$$

The letters D_1 and D_2 represent the values of the variables in $\delta I_1(t_m+n)$ and $\delta I_2(t_m+n)$, respectively. These values have been specified at an earlier time. For example, $\delta I_1(4)$ in the expression for g_4^* in the example above might contain $u_1(3)$, which must match the value of $\gamma_1^*(3, y_1(3) = \rho_{1k_1}, \xi(2))$ in the expression for g_3^*.

In the second term of $J_{t_m+1}^*$ in Eq. 4-5, namely the expected cost-to-go at the next member of the s-partition, the expectation is taken as follows:

$$E\left[J_{t_{m+1}}^* + 1(\xi(t_{m+1}))\right] = \sum_{k_1=1}^{M_1} \cdots \sum_{k_M=1}^{M_1} \sum_{\ell_1=1}^{M_2} \Pr\{y_1(t_m+1) = \rho_{1k_1}, \cdots, y_1(t_{m+1}) = \rho_{1k_M}, y_2(t_m+1) = \rho_{2\ell_1}, \cdots, y_2(t_{m+1}) = \rho_{2\ell_M} | I(t_m)\} \cdot J_{t_{m+1}}^* + 1(\xi(t_{m+1})) \quad . \tag{4-11}$$

Using Bayes' rule and the fact that strategies are fixed, the conditional probability can be shown to reduce to another recursive expression, as follows:

$$\Pr\{y_1(t_m+1) = \rho_{1k_1}, \cdots, y_2(t_{m+1}) = \rho_{2\ell_M} | I(t_m)\} = \sum_{i_1=1}^{N} \{q_{i_1 k_1}^1(t_m+1) q_{i_1 \ell_1}^2(t_m+1) \prod_{n=2}^{M} H_n\} \xi_i(t_m) \tag{4-12}$$

where

$$H_n = \sum_{i_n=1}^{N} \{q_{i_n k_n}^1(t_m+n) \cdot q_{i_n \ell_n}(t_m+n) \cdot R_n\} \tag{4-13}$$

and R_n is derived recursively as

$$R_2 = p_{i_1 i_2}(u^*(t_m+1)) \tag{4-14}$$

$$R_n = \sum_{i_{n-1}=1}^{N} p_{i_{n-1} i_n}(u^*(t_m+m-1)) R_{n-1} \quad , \quad n = 3, \cdots, M \quad . \tag{4-15}$$

The remaining expression to compute is $\xi(t_{m+1})$, the argument of $J_{t_{m+1}}^* + 1$, given $\xi(t_m)$. The HMF provides the following expression for $\xi(t)$, given $\xi(t-1)$:

Let

$$y_1(t) = \rho_{1k} \quad , \quad y_2(t) = \rho_{2\ell} \quad ; \tag{4-16}$$

then

$$\xi_{i_2}(t) = \frac{\sum_{i_1=1}^{N} q_{i_1 k}^1(t) \, q_{i_1 \ell}^2(t) \, p_{i_1 i_2}(u(t)) \xi_{i_1}(t-1)}{\sum_{i_1=1}^{N} q_{i_1 k}^1(t) \, q_{i_1 \ell}^2(t) \, \xi_{i_1}(t-1)} \quad . \tag{4-17}$$

with $\xi(o) = p(1)$. This formula is independent of information structure. Consequently, $\xi(t_{m+1})$ is computed from $\xi(t_m)$ by replacing $\xi(t-1)$ with $\xi(t_m)$ in Eq. 4-17 and applying Eq. 4-17 M times to get the sequence $\{\xi(t_m+1), \ldots, \xi(t_{m+1})\}$.

For large state spaces, observation spaces, decision spaces, and time horizons, the brute-force enumeration of strategies can become an enormous computational burden. Large problems could be reduced to a more manageable size by employing the approximate SSA described in subsection 2.3.

5. Conclusion

The class of decentralized stochastic control problems for which an algorithm exists has been enlarged. It has been shown that the Smallwood-Sondik algorithm can be applied to problems decomposed into sequential partitions satisfying a separation result. Since these problems tend to be "large" computationally, an approximation of the SSA, designed to simplify such problems, has been presented.

References

1. Wonham, W.M., "On the Separation Theorem of Stochastic Control," SIAM J. Control, Vol. 6, No. 2, 1968, pp. 312-326.

2. Monahan, G.E., "A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms," Man. Science, Vol. 28, No. 1, January 1982, pp. 1-16.

3. Smallwood, R.D., and E.J. Sondik, "The Optimal Control of Partially Observable Markov Processes over a Finite Horizon," Oper. Research, Vol. 21, No. 5, 1973, pp. 1071-1088.

4. Sondik, E.J., "The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs," Oper. Research, Vol. 26, March-April 1978.

5. Platzman, L., "Optimal Infinite-Horizon Undiscounted Control of Finite Probabilistic Systems," SIAM J. Control and Optimization, Vol. 18, 1980, pp. 362-380.

6. Hsu, K., and S.I. Marcus, "Decentralized Control of Finite State Markov Processes," IEEE Trans. on Automatic Control, Vol. AC-27, No. 2, April 1982, pp. 426-431.

7. Hsu, K., "On Discrete Time Centralized and Decentralized Estimation and Stochastic Control," Ph.D. Dissertation, Dept. Elec. Engin., Univ. Texas, Austin, December 1979.

8. Varaiya, P., and J. Walrand, "On Delayed Sharing Patterns," IEEE Transactions on Automatic Control, Vol. AC-23, No. 3, June 1978, pp. 443-445.

9. Yoshikawa, T., and H. Kobayashi, "Separation of Estimation and Control for Decentralized Stochastic Control Systems," Proc. IFAC Conference, Helsinki, Finland, 1978.

10. Yoshikawa, T., "Decomposition of Dynamic Team Decision Problems," IEEE Trans. on Autom. Control, Vol. AC-23, No. 4, August 1978, pp. 627-632.

DECISION PROCESSES FOR LARGE SCALE RESOURCE ALLOCATION PROBLEMS

Marc D. Diamond
FMC Corp., Northern Ordnance Div.
East Coast Engineering Office
King George, VA 22485

Olivia M. Carducci
Department of Mathematics
Carnegie-Mellon University
Pittsburgh, PA 15213

ABSTRACT

Problems of resource allocation and scheduling are emerging as important components of C^3I systems. However, some of the application areas of interest do not conform to the standard queuing and scheduling models, but require new approaches to system definition and new solution techniques. Such a situation occurs primarily in systems represented initially by hybrid state vectors. This paper presents an approach to reducing these problems to Markov decision processes. The resulting decision problem is solved by generating a decision tree. Computational complexity then emerges as a major issue, since the size of the decision tree will grow exponentially with the size of the system. An approach to pruning the search space is presented which is based on generating upper and lower bounds on the expected costs associated with available decisions at each node in the tree. These bounds are generated by solving closely related problems, or "relaxations", for which relatively efficient algorithms exist. The techniques proposed in this paper can be applied to a wide range of problems in areas such as sensor or surveillance resource scheduling, support for ordnance delivery systems, and shipboard operations, such as damage control. The example chosen to illustrate basic concepts is the allocation of illuminators for terminal homing of surface to air missiles.

1. INTRODUCTION

In a stochastic resource allocation problem, a decision maker must allocate scarce resources among tasks in order to carry out a global strategy in an optimal manner. Such problems are cast in a dynamic probabilistic environment, so that the evolution of the system cannot be fully controlled by the actions of the controller. Problems of this nature are an important component of many C^3I systems. For Naval C^3I this includes areas such as optimal end-game strategies for surface to surface warfare, tasking of combat air patrol (CAP) in outer air battle, launch scheduling for area defense, allocation of damage control parties in shipboard operations, and the allocation of sensors for surveillance. However, in some instances, the application areas have characteristics which do not allow them to be readily treated within the context of standard queuing and scheduling models. For example, the time required to process tasks may not be easily modeled as exponentially distributed random variables, and the system may not be easily modeled as a continuous time discrete state (jump state) process. This is illustrated by several examples given as follows:

In fighting shipboard fires, the objective is to minimize the effect of battle damage on the battle readiness of the ship. The critical resource in this case is the limited number of damage control parties available to be allocated to various areas of the ship. The state of the system can be represented by a vector with components indicating which rooms or areas currently contain damage. These may be binary variables. State changes occur every time a fire is extinguished or whenever a room or area first contains a fire (as in the spread of a fire from an adjacent room). However, the time required for the damage control parties to move from their current position to the position of the fire, as well as the time required for the damage control party to contain the fire, must also be represented. A good scheduling algorithm would have to account for such factors as the tactical value of the rooms or areas affected, as well as the predicted rate of spread of fires from one area to another.

In scheduling the interception of threats by surface to air missiles, the critical resources include the illuminators, use of the launch rails, and the number of surface to air missiles (SAM) remaining on board. The problem is to prioritize or schedule the engagements so as to, for example, minimize the expected number of threats entering the close-in zone. The representation of the state of the system must in some manner contain an "activity vector", with binary components for each threat currently active (i.e. not yet defeated). Whether or not a threat is in the engagement envelope may also be represented as a discrete state component. This is a "partial" jump state process, which changes state whenever a threat enters or exits the engagement envelope or when it is defeated. However, in establishing a schedule to minimize conflict in the use of resources, such as illuminators, other characteristics of the threat, such as range and speed, as well as the fly out times of intercepting SAMs, must also be represented.

In committing combat air patrol (CAP) to missions in an outer air battle, the critical resource is the number of CAP units available. The state contains components to represent each available mission (complete/not complete). However, the distance of each CAP from its current location, to the point when it can first carry out a given mission, must also be accounted for. Critical times (due dates) of the tasks are defined according to the characteristics of each mission. For example, it may be a goal to defeat a long range bomber before it has the opportunity to release air to surface missiles.

Although some of these problems do not fit well into the models which support most of the current theory in stochastic scheduling and resource allocation, they do have several important characteristics. In the first place, they consist of a certain number of distinct tasks or subproblems. Secondly, each task has a specific release date and due date, which may be random. Third, we suppose that we are given a

set of scarce resources, or "processors", which must be allocated among the tasks, and are required to solve each individual task. Each attempt to solve a problem requires a resource for a period of time specified either deterministically or probabilistically. Finally, the state of each task can be represented, or approximated, by a hybrid state vector (i.e. the components of the stated variable are a mixture of continuous and discrete variables).

This paper addresses the problem of modeling and solving the types of problems characterized above. We present a procedure for reducing the hybrid state process, describing the types of resource scheduling problems being addressed, into a discrete time, discrete state Markov decision process. This mapping is achieved by enumerating the "critical" times (due dates) of the individual tasks involved, explicitly generating intervals of time that a given resource will be required to solve a given problem before the system changes state, and "back-projecting" from the critical times to determine when a given set of resources must be committed to a problem if it is in fact to be solved before the system changes state. Examples of critical times are the time when a threat exits the engagement envelope and enters the close-in zone, the time when a fire spreads into a high value area, or the time when a bomber gets close enough to release its air to surface missiles. Thus, the entire mapping process is driven by a "latest time to action" analysis.

The latest time to action analysis yields a set of timing diagrams. The relationships and interactions among the intervals of time during which specific resources are required (referred to as "resource intervals" below) are represented in the form of a graph (the resource constraint graph), which is constructed from these timing diagrams. The resource constraint graph implicitly contains the entire set of legal sequences involving the commitment of resources to problems over time. The resulting decision problem is approached by searching a decision tree. The use of a decision tree allows for the use of pruning techniques. This is important because the size of the decision tree, and in fact the size of the state space, grows rapidly with the size of the problem: that is, the number of resources, the time span for the scenario, and the number of subtasks involved. The procedures for generating the decision tree and applying the pruning techniques make express use of the resource constraint graph. Pruning is based on "relaxations" of the original problem, which make explicit use of the fact that the systems involved are naturally represented as interacting subsystems corresponding to individual missions or tasks.

In the following, the launch scheduling problem provides a context in which the basic approach to these types of scheduling problems can be discussed. We note that in the description given below we assume that we know deterministically when the jump times of the system are, as well as what the various problems will be and when they will arise. Extension to the more general case (i.e. jump times modeled as random variables, and the nature of all problems not known in advance) can be achieved provided a sufficient model exists. However, this extension introduces complexities which will not be discussed here.

2. ILLUMINATOR SCHEDULING

The launch scheduling problem is one of specifying a schedule for the interception of a number of threats approaching a ship. We may consider these to be cruise missiles in the manner we are modeling them here. The critical resource being considered here is a single illuminator which must be allocated to each threat just prior to its interception by a SAM. It is assumed that a shoot-look-shoot doctrine is in effect. That is, the outcome of the engagement of a given threat must be known before a second round is launched against that threat.

2.1. The Single Threat Case

The approach to finding an optimal scheduling policy will be based on a means for transforming a system involving a ship defending against airborne threats to a Markov decision process. To develop this approach, consider first a scenario involving a single threat targeted at a ship, which is defending itself with surface to air missiles. The threat is launched at a distance and follows a trajectory which takes it through the engagement envelope at specific points in time (figure 1). It is assumed that the trajectory, including the launch times, entering times,and exiting times are known deterministically.

Assume the engagement envelope is such that the SAM can defeat the threat only when it is in the region between points A and C (figure 2). Assume furthermore that in order to intercept the threat before it traverses point A (that is, before it exits the engagement envelope), the engaging SAM must be launched before the threat gets to point B (this is to account for its fly out time); in order to intercept the threat before it gets to point B, the engaging SAM must be launched before the threat gets to point D; and that in order to intercept the threat after it gets to point C, the engaging SAM must be launched after the threat gets to point E. These "decision points" divide the distance between the ship and the maximum detection range into six intervals, as shown.

There is a state in a resulting state space associated with each of these intervals. The system is in a given state if the threat is in the corresponding interval. This mapping can be achieved, since, from the point of view of the number of remaining available intercepts against that threat, it does not matter where in the particular interval the threat is at any point in time. If no action is taken against the threat when it is in a given interval, the next state will always be the one representing the next interval through which the threat will pass (figure 3). The corresponding

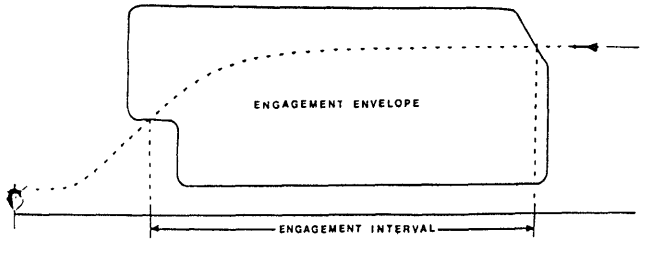


Figure 1. A Single (Cruise Missile) Threat Approaching A Ship On A Known Inbound Trajectory.

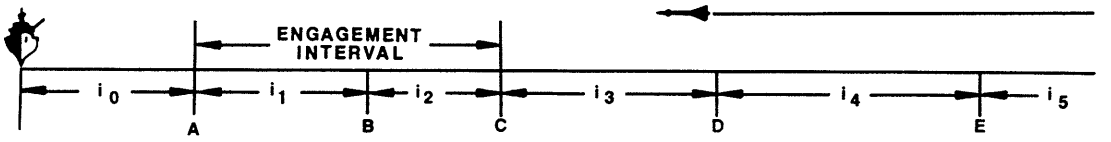


Figure 2. Control Intervals Generated By Backprojecting Boundary Points Of The Engagement Interval.

change in the system is to move along the arc marked ϕ to the next state. If a SAM is launched, the next interval to which the threat arrives may be the one adjacent to the current interval; however, the next state in which a control action pertaining to that threat is possible is the state identified with that interval in which the interception occurs. This is a consequence of a crucial assumption that the outcome of a given engagement of a SAM with a threat must be known before another round may be launched against that threat. Thus the system moves along the arc marked l/h (launch/hit) to state i_x (threat defeated) if a SAM is launched with a subsequent successful intercept, or moves along the arc l/m (launched/miss) if a SAM is launched and the threat is not subsequently defeated. The state diagram given in figure 3 describes all possible ways in which the system can evolve, in consideration of the set of possible control actions.

A control problem is defined by assigning a cost and transition probability (which in this case corresponds to the probability of defeating the threat) to each arc in the transition diagram. The problem can then be solved using any of a set of well known techniques. It is important to note that a solution to this problem, which in this case determines those intervals in which SAMs should be launched against a given threat, will not specify precisely at which point in the interval the threat should be when the SAM is launched, only that it be somewhere in that interval. A solution to the latter problem requires setting up and solving a system of linear equations.*

The basic means for constructing the Markov model described above can be readily generalized to account for differing threat types (including differing speeds, approach patterns, and altitudes). Differing threat speeds will affect the length of the control intervals created by this modeling technique. Arbitrary threat types and approach patterns can also be treated directly within the context of this scheme, the idea being to make note of the engagement envelopes, projecting crossing times onto the time line, and backprojecting those points to account for boundaries on the times/threat ranges at which control actions (engage/do not engage) must be made.

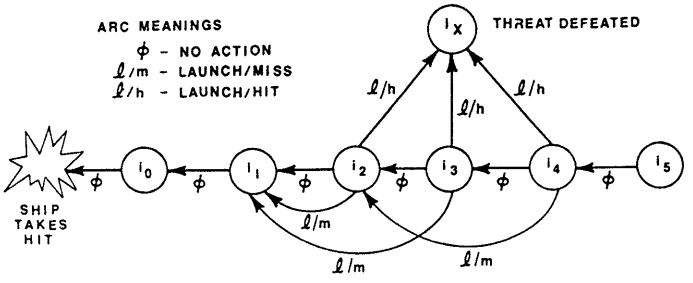


Figure 3. Markov Chain Representation Of The Resulting Control Problem; Single Threat Case.

2.2. Multiple Threat Scenarios and Resource Conflicts

Thus far, the possibility of resource conflict and multithreat scenarios has not been considered. As above, we will consider a single illuminator as being the primary scarce resource. We outline here a method for generating a set of resource intervals which are used to generate a decision tree, much in the manner of the single threat case given in the previous section. These resource intervals will be used to generate a "resource constraint graph" which can be used to describe and search all possible legal sequences of intercepts which guarantee that there will be no illuminator conflicts. In the example pursued below a two threat scenario will be discussed. The extension to larger scenarios should be obvious.

Figure 4 (a, b, c, and d) shows time lines for two threats targeted at a ship with a single SAM missile system and a single illuminator on board. The threats are assumed to travel at the same speed. The time lines are shifted with respect to each other so that the threats are always represented in the same vertical position (thus, the time lines represent time relative to the anticipated time of arrival of the threat at the ship). Using the same reasoning as before, we can construct intervals by backprojecting points from boundaries (created by the last time action can be taken) within the context of the current launch sequence. Figure 4 (a, b, c, and d) illustrates all possible engagement sequences. In each case, the intercept is assumed to occur at the end (left hand side) of the illuminator interval.

Consider figure 4a. Note that in order to engage the later arriving threat (threat 1; upper time line) before it arrives at point A, the illuminator must be free for use before the earlier arriving threat (threat 2; lower time line) gets to point B, assuming the length of the illuminator interval is as shown. Terminal illumination of the second threat for an interception at point B means that the illuminator will be in use at and before point B. Another interception of threat 1, which would occur at point C (actually, a previous interception, since we are working backwards in time), cannot be achieved because the required illumination period would extend outside the engagement interval. Similarly, a previous interception of threat 2 would, because of the delay imposed by the shoot-look-shoot doctrine, require an illuminator period extending beyond the engagement interval.

Figure 4(b) illustrates another possible sequence which is generated by scheduling the last interception of threat 2 at point A, and backprojecting resource constraints in the same way as was given in figure 4(a). This generates another set of resource intervals and another set of interception times.

* The details of this process are not difficult, but go beyond the purpose of the current discussion. A description can be found in [1].

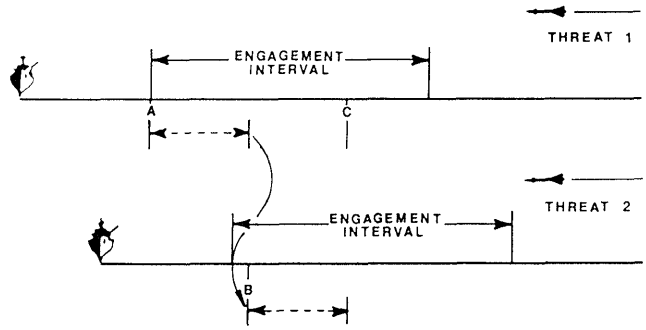


Figure 4a.

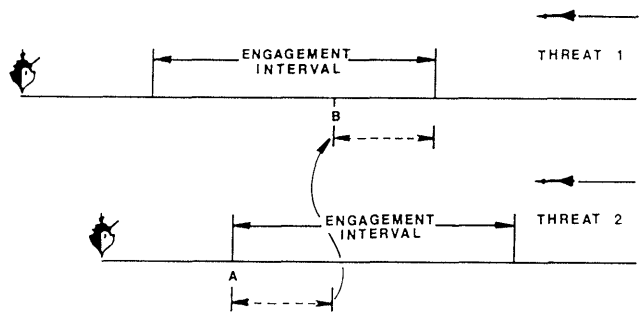


Figure 4b.

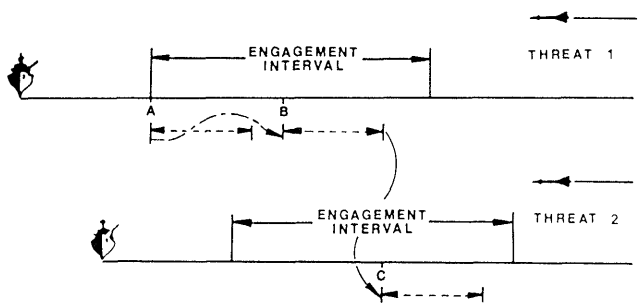


Figure 4c.

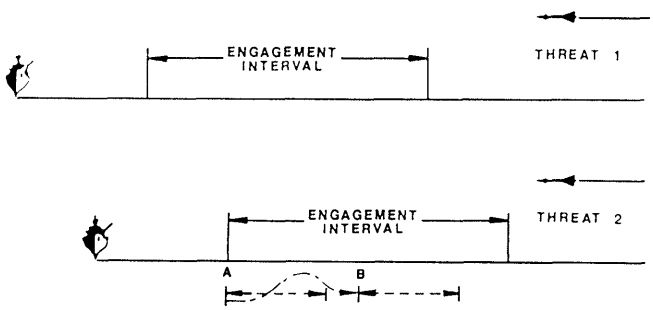
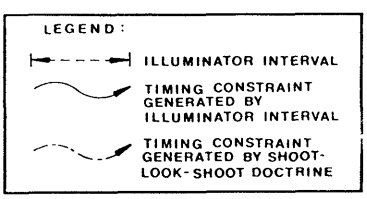


Figure 4d.



Timing Diagrams and Resource Requirement Intervals For A Two Threat Scenario.

Figure 4 (c and d) illustrates the generation of the two final possible engagement sequences. In the last two examples constraints are generated by the shoot-look-shoot doctrine, since these strategies involve launching twice in succession against a given threat.

In this manner, the set of all possible resource intervals for each threat can be generated and projected onto a common time line (figure 5). The resource constraint graph is generated from this set of intervals by identifying each interval with a vertex of the graph, such that there is an arc between two nodes for intervals pertaining to different threats if those intervals intersect on the time line. There is also an edge between two vertices for intervals pertaining to the same threat, if interception at the end of each interval implies a violation of the shoot-look-shoot doctrine. For the purposes of the following discussion, each resource interval and its corresponding vertex is given a two component label $L=(l_1, l_2)$, where l_1 is the number of the threat addressed in the interval represented by vertex L, and l_2 is an index of the resource interval within the set pertaining to threat l_1. The resource interval constraint graph associated with the example of figure 4 is shown in figure 6.

Let G = (V, E) be a graph, with vertex set V and edge set E. A vertex packing in G is a set $P \subseteq V$ of vertices such that no elements of P are adjacent in V. It is clear that any vertex packing in the resource constraint graph corresponds to a legal sequence of interceptions, that is, a sequence of interceptions that does not violate a resource constraint. Although it may not seem as obvious, it can nonetheless be demonstrated that the converse is also true [1]. That is, any legal schedule of interceptions, in an actual scenario, corresponds to a unique vertex packing in the resource constraint graph. Thus, implicit in the resource constraint graph is the set of all valid launch sequences. This fact is applied in the construction of a decision tree used to generate optimal decisions at each point in time.

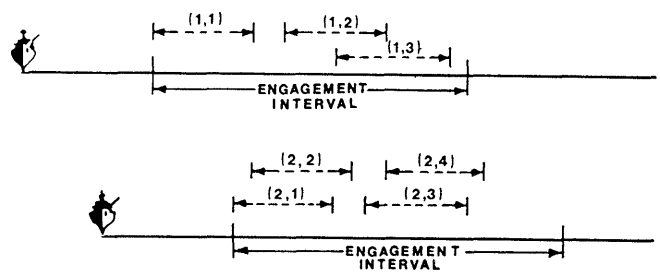


Figure 5. Aggregation Of All Resource Requirement Intervals From The Timing Diagrams Of Figure 4.

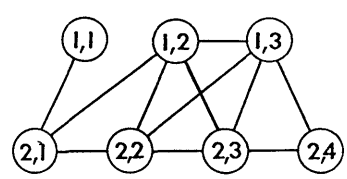


Figure 6. Resource Constraint Graph Representation of the Two Threat Scenario.

Comment: There is the potential that the number of resource intervals will grow very quickly, in fact exponentially, with the number of threats. It has been our experience, however, that this in fact does not happen, primarily because of boundary conditions imposed by (1) the shoot-look-shoot doctrine, and (2) release dates and due dates of the tasks themselves. In practice, further reduction in the number of resource intervals can be realized by rounding the numerical representation of the end points of the resource intervals to, for example, integral values. Then in some cases intervals generated by different launch sequences tend to fall on top of one another and thus need only be represented once.

3. GENERATING THE DECISION TREE

We outlined briefly the mechanics of generating the decision tree. The focus here is to illustrate the way the state of the system is maintained as alternatives are explored, with a particular emphasis on the use of the resource constraint graph.*

Each node in the decision tree represents a particular state of the system. In this case, the current state is represented by a combination including (1) a subgraph of the resource constraint graph, (2) an "activity vector" indicating which threats are currently active (i.e. not yet defeated), and (3) an index into an ordered list of decision/outcome times referred to as an "event list". The purpose of the activity vector is obvious. The subgraph of the resource constraint graph implicitly contains all legal launch sequences which can occur from that particular point in the system evolution. This is a convenient way of maintaining the set of all possible future actions, which is what a search of the decision tree is meant to explore.

There will be two node "types": decision and outcome. A control action of either launch or no launch may be applied to a decision node, and the effect of a previous decision to launch is known as an outcome node. In the general formulation of a Markov decision process, there is only one type of node (decision) and the outcome must be known immediately. In our case, the situation is complicated somewhat by the fact that a decision to intercept a threat in a given resource interval is actually made at the (previous) point in time when the countering surface to air missile must be launched, and in fact, several decision points may be passed before the outcome of a previous decision is known. Because of this, transition probabilities might be conditionally dependent on previous points in time. In order to maintain Markov dynamics, the state space would have to be expanded. It can be shown, however, that this is done implicitly by the manner in which the decision tree is generated, and that Markov dynamics are maintained.

An event list is needed to determine the type (decision/outcome) of the offspring of the node in the decision tree. The list is generated by projecting the left hand side of each resource interval onto a common time line. The (outcome) point which results from the projecting of resource interval (s,t) is labeled "o(s,t)". For each outcome point, o(s,t), a point d(s,t) is generated on the time line which corresponds to the previous point in time at which the decision (launch/no launch) must be made with respect to an interception at point o(s,t). A list is generated from the projected points, ordered as time increases into the past. The time line and event list derived from the example of figure 5 is given in figure 7 (a and b).

Generating the state of the offspring of a node in the decision tree requires (1) generating a subgraph, G', of the graph, G, associated with the state of the parent node, (2) updating the activity vector, and (3) determining which element of the event list "pertains" to that offspring. The manner in which this is done depends on the type of the parent node and the decision made (if the parent is a decision node), or the outcome (if the parent is an outcome node). If the parent is a decision node, a decision to launch against a threat, s, for an interception in interval t can be made only if the vertex (s,t) is in graph G. In this case, the graph associated with the offspring, G', is a subgraph of the graph associated with the parent, G, including vertex (s,t) but excluding all vertices adjacent to (s,t) in G. This guarantees that no future launch decisions will be made which may lead to an resource conflict. The graph G' associated with the offspring of a decision node where the decision is not to launch against threat s, for an interception in resource interval t, is the graph G minus vertex (s,t).

The state of the offspring of an outcome node depends on whether or not the threat, s, has been defeated. If the threat has not been defeated, there is no change in either the graph or the activity vector. If the threat is defeated, the corresponding entry in the activity vector is changed to reflect this, and all vertices corresponding to threat s are removed from the graph.

Updating the index into the event list for an offspring node is accomplished in the following manner: let i be the index into the event list, which is a component of the state of a given node, n, in the decision tree. Let G' be the subgraph associated with the state of one of the offspring, n', of n. To determine the index, i', (into the

---

* A more complete discussion, including detailed examples, will be found in [1].

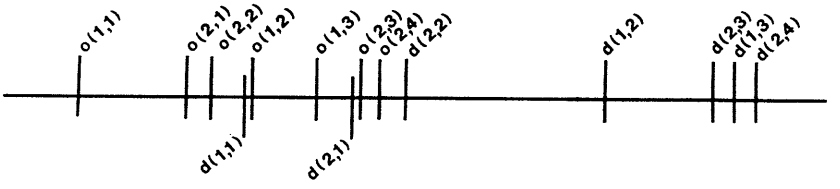


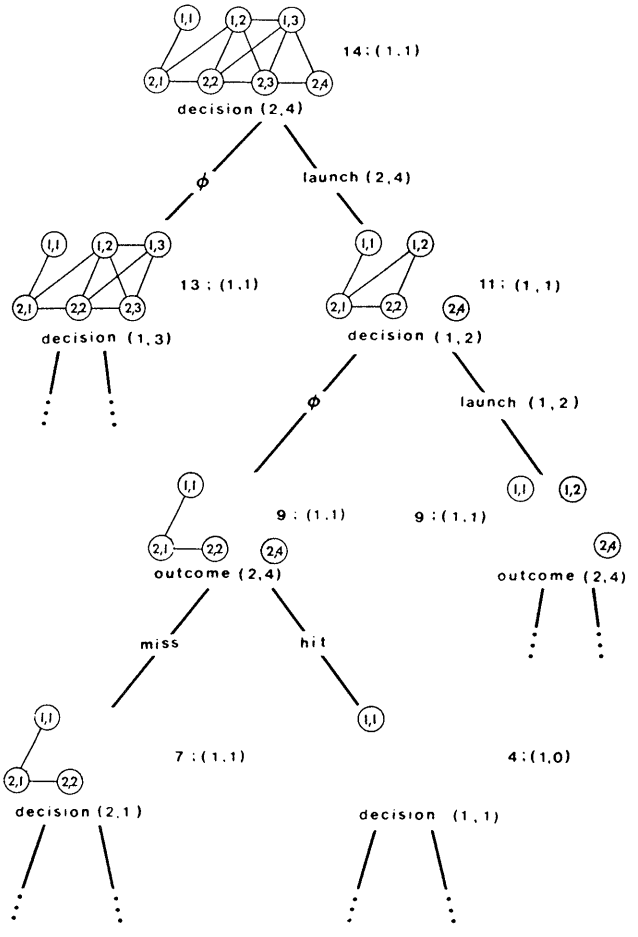
Figure 7a. Labeled Time Line

EVENT LIST

Index	Label
1	o(1,1)
2	o(2,1)
3	o(2,2)
4	d(1,1)
5	o(1,2)
6	o(1,3)
7	d(2,1)
8	o(2,3)
9	o(2,4)
10	d(2,2)
11	d(1,2)
12	d(2,3)
13	d(1,3)
14	d(2,4)

Figure 7b. Associated Event List

event list) of n', examine the labels on the event list elements i-1, i-2, ..., until an element i' with a label o(s,t) or d(s,t) is found such that the vertex (s,t) is in G'. If the label is o(s,t) then the offspring is an outcome node. If the label is d(s,t) then the offspring is a decision node. Figure 8 shows several levels of the decision tree corresponding to the example of figures 4 through 7.



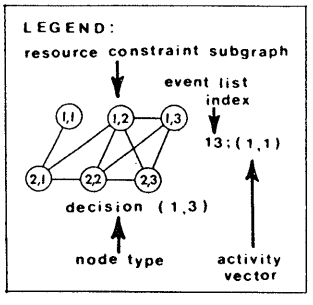


Figure 8. Section Of The Decision Tree For The Two Threat Scenario.

The expected cost associated with any node is defined recursively. For a decision node, it is the maximum of the values of the expected costs associated with its offspring nodes. For an outcome node, it is an average of the values of the expected costs associated with the offspring, weighted by the probability of a successful engagement (for the "hit" offspring) and the probability of an unsuccessful engagement (for the "miss" offspring).

Comment: It would be natural to question the strategy described thus far which first generates the resource constraint graph and then generates the decision tree. After all, since we generate all potential launch sequences in generating the resource intervals, why generate them again by explicitly enumerating the vertex packings in the resource constraint graph. There are several important reasons for this. In the first place, it would be very difficult to maintain the Markov dynamics if one attempted to solve the decision problem at the same time one generated the resource intervals. This is because the complete "structure" of the problem is not known at that time. In the second place, the resource constraint graph can be used explicitly to generate bounds for pruning the search space as discussed in the following section.

4. BOUNDING AND FATHOMING

Enumerating the decision tree as described above provides a finite algorithm for computing an optimum launch policy. However, the number of nodes in the decision tree grows exponentially with the number of resource intervals, and the number of resource intervals grows rapidly with the number of threats. Thus, for multiple threat scenarios, it is important to prune some of the branches of the tree. The approach presented here is based on implicit enumeration [2]. Implicit enumeration approaches attempt to use various tests to disregard some options which the tests show will not provide a better solution than one already known. In our approach, the tests are based on computing upper and lower bounds on expected costs for each node. The resource constraint graph is used to construct one of these bounds, and another graph generated from the resource intervals, the "policy constraint graph" (discussed below), provides the other. The basis for these bounds is to consider first the cost associated with the optimal policies if the problem of countering each threat was treated as an independent subsystem, much in the manner of the single threat case discussed in section 2.1 above. The bounds are then constructed by either over-estimating or underestimating the effect of the interaction of these subsystems.

The resource constraint graph provides a "can do no better than" bound. For example, in a minimization problem, it provides a lower bound on the expected cost associated with a given decision at a decision node. (A minimization problem would result if the objective were defined to "minimize the expected number of hits on ownship", "minimize the probability that the ship will be hit by more than one threat", and so forth). Assign values c(s,t) to the vertices of the resource constraint graph (representing the value of using the illuminator at each interval) by computing the expected cost of having the first interception against threat s occur in interval t, assuming that subsequent launches for all threats will not create illuminator conflicts. This would then guarantee at least that the first interception of each threat would not create an illuminator conflict. The assumption that further

interceptions against a given set of threats (provided that the first interception failed) will not create scheduling conflicts will not hold in general, and hence this provides only a bound.

Deriving this bound, therefore, requires selecting a packing of the resource constraint graph which contains exactly one vertex from the set of vertices associated with each threat, such that the sum of the vertex costs are minimized. This can be represented as a 0-1 integer (vertex packing) problem:

$$\min \; d = \sum_{s=1}^{S} \sum_{t=1}^{T_s} c(s,t) \cdot x(s,t)$$

$$\text{s.t} \quad \sum_{t=1}^{T_s} x(s,t) = 1, \quad s = 1,\ldots,S$$

$$x(s,t) + x(s',t') \leq 1$$

for all (s,t), (s',t') such that launching against threat s in interval t conflicts with a launch against threat s' in interval t'

$$x(s,t) \in \{0,1\}, \quad s=1,\ldots,S, \; t=1,\ldots,T_s.$$

Although combinatorial in nature, deterministic programs are generally much easier to solve than their stochastic counterparts. For this reason, it is common practice to try and find good deterministic approximations for stochastic programs [3]. In fact, 0-1 integer programs of the type given above have been extensively studied and many good algorithms exist for their solution [2,4].

An upper (can do at least as good as) bound is generated by solving a problem of the form given above on a slightly different graph; the "policy constraint graph". The policy constraint graph is a multi-partite graph, with each set of the partition corresponding to a specific threat. Each vertex in a given partition, s, corresponds to a subset of the control intervals pertaining to threat s. A subset will be represented in this set if the interceptions that occur at the end of these control intervals do not violate the shoot-look-shoot doctrine. The policy constraint graph for the example being pursued here is shown in figure 9. This graph consists of two disjoint subsets, one corresponding to threat 1, and the other corresponding to threat 2. Note that each vertex in this graph corresponds to a launch sequence against a specific threat, accounting for the possibility that more than one launch against a given threat may be needed. Thus, each vertex corresponds to a scheduling policy to counter that threat considered independent of the problems created by the other threats.

As an example, vertex $v_{1,5}$ of figure 9 contains intervals (1,1) and (1,3), which correspond to planning the first interception against threat 1 at the end of interval (1,3) and the second interception (should it be required) at the end of interval (1,1). An edge exists between two vertices in this graph if the associated policies imply an illuminator conflict. For example, there is an edge between nodes $v_{1,5}$ and $v_{2,1}$ because there would be a conflict if intercepts were planned for the end points of intervals (1,1) and (2,1). Note that this conflict would not arise if the first engagement of threat 1 at the end of interval (1,3) was successful.

A cost is assigned to each node in the policy constraint graph which corresponds to the expected cost of the policy associated with the given node. As above, the problem becomes one of selecting exactly one vertex from each set in the partition such that no two vertices are adjacent, and which minimizes the total cost. The result is a global policy which is suboptimal (although feasible). It is a conservative policy, since it may reserve an illuminator interval for a second launch at a threat which in fact may be defeated by the first interception. As such, it provides an upper bound (in a minimization problem) on the expected cost for any policy applied to the current state as represented by the current position in the decision tree.

A node in the decision tree is fathomed if all branches emanating from it can be pruned. The upper and lower bounds described above can be used to fathom a node in two ways. First, if the upper bound at that node is equal to the lower bound, then it is fathomed, since the optimum policy from that point on is given by an optimum solution to either of the bounds. In practice, a node can be fathomed if the difference in these bounds is within a reasonable tolerance. Second, if the upper bound of one of the offspring of a decision node is not as good as the lower bound of any other offspring of a decision node, then it is fathomed, since no policy containing it is as good as the policy represented by its sibling.

As the decision tree is generated, two vertex packing problems are solved for each node in an effort to prune the tree to a manageable size. Since the graphs associated with a descendant of a given node are all subgraphs of the graph associated with the parent node, we do not need to derive each bound from scratch. The optimum solution to the parent problem can be used to generate a feasible (and generally, nearly optimal) solution to the offspring problems.

Comment: Experimental results on large scale raids indicate a polynominal growth, with an exponent of approximately 1.8, on the number of nodes visited as a function of raid size. Run times increase faster because of a corresponding increase in the amount of time required to compute the bounds. Obviously these results depend to a large extent on parameters defining the experiment (e.g. the distribution controlling threat arrival rates. These results are meant only to provide a base reference.6.

5. SUMMARY AND CONCLUSIONS

We have presented a technique for defining and solving Markov decision problems on large scale resource scheduling problems. The approach which was presented was based on a two step procedure which involved (1) generating a set of resource intervals and defining a resource constraint graph, and (2) expanding a decision tree. When the problem can naturally be decomposed into interacting subsystems, then a good means for generating bounds for pruning the state space exists. We have demonstrated this approach in the application to the launch scheduling problem, although many other potential applications in the C^3I arena exist.

Several issues of interest were not addressed here. Perhaps most important, it is not realistic to assume that the jump times of the system will be known ahead of time, as assumed above. In general

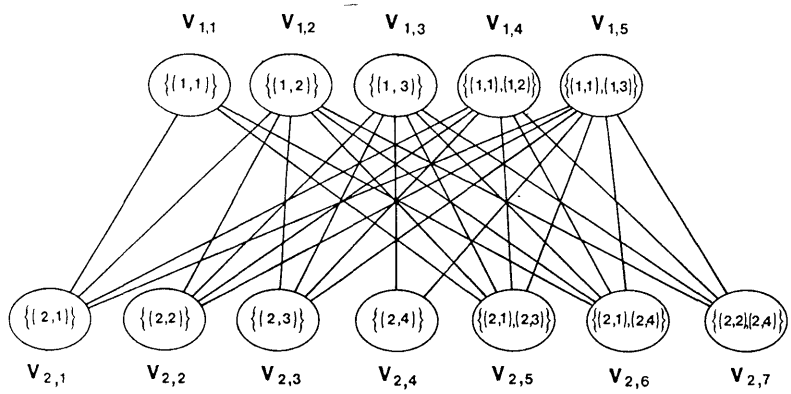


Figure 9. Policy Constraint Graph For The Two Threat Scenario.

the sojourn times of the system in any state would have to be represented by random variables, as would the length of the resource intervals, and the release and due dates of the problems. In a cursory examination of these problems we have determined that it is still possible to define a Markov decision model on such processes. However, the size of the resulting state space would be increased significantly. Thus, the problem of generating efficient bounds based on good techniques for approximating the expected costs associated with decisions at each node in a decision tree is emerging as the central issue to be addressed in further research.

We are currently looking at bounds generated by approximations based on classical stochastic scheduling and bandit models as an alternative to the method suggested above [1,5,6]. The theory of two stage stochastic programs with fixed recourse [1,7,8] can be used to extend the bounds generated above to include second stage effects. Such programs have 0-1 integer first stage variables, which, as above, guarantee that there will be no resource conflict for the initial application of each resource, but which also take into account expected costs due to possible conflicts in the second attempt to solve a given problem. Bounds generated by these two-stage approximations to the multi-stage program described above should be more efficient than those suggested in section 4. However, they will require more computational effort to generate, and it is not known at this time whether this extra effort will be justified.

Other approaches suggest themselves. For example, in any practical application, it may be feasible to use human expertise, experience, and common sense to determine which branches of the decision tree to explore, or what alternatives can be dismissed out of hand. The result would be an expert system or "knowledge based" approach for controlling the search of the state space. Unlike the other expert systems approaches to battle management problems which are currently being pursued, the approach suggested here would be built upon a decision theoretic model of the underlying processes.

References

[1] M.D. Diamond, "On the decision theoretic approach to command and control for naval AAW battle management," FMC Corporation, Northern Ordnance Division Technical Report FMC-NOD-5-XX, forthcoming.

[2] H.M. Salkin, Integer Programming, Reading, Mass.; Addison-Wesley, 1975.

[3] J.R. Birge, "The value of the stochastic solution in stochastic linear programs with fixed recourse," Mathematical Programming, 24 (1982), 314-325.

[4] G.L. Nemhauser, and L.E. Trotter, Jr., "Vertex packings: structural properties and algorithms," Mathematical Programming, 8 (1975), 232-248.

[5] S.M. Ross, Introduction to Stochastic Dynamic Programming, New York; Academic, 1983.

[6] M. Pinedo, "Stochastic scheduling with release dates and due dates," Operations Research, 31 (1983), 559-572.

[7] R.D. Wollmer, "Two stage linear programming under uncertainly with 0-1 integer first stage variables," Mathematical Programming, 19 (1980), 279-288.

[8] D.B. Yudin, and E.V. Tsoy, "Integer valued stochastic programming," Engineering Cybernetics, 12 (1973), 1-8.

COMBINATION OF EVIDENCE IN C^3 SYSTEMS

I.R. Goodman

Command & Control Department
Code 421
Naval Ocean Systems Center
San Diego, California 92152

ABSTRACT

This paper has a threefold thrust:(1) a brief survey is presented of the development of approaches to modeling C^2/C^3 systems as given primarily in this forum-The MIT/ONR Workshop on C^3 Systems;(2) an outline of a theory of C^3 systems is developed which is compatible with previous efforts and which is rich enough for rigid, yet tractable, analysis;(3) as part of this theory, a procedure is exhibited for integrating subjective and objective/probabilistic/numerical information for C^3 system decisionmakers.

1. INTRODUCTION

The C^3 problem is a real-world problem and thus, analogous to theories in Chemistry, Physics, or Biology, a proposed theory for a C^3 system must be based on empirical, as well as sound, logical considerations. In addition, such a theory-following the usual pattern of change for scientific inquiries- will incorporate, overlap to some degree, or otherwise relate with, previously established models. Finally, the author's own biases and predilections will generally be reflected in the degree of detail granted to the various components of the overall model.

Compatible with the above philosophy, the goal of this paper is the development of a general C^3 theory which accounts for a systematic/comprehensive treatment of the combination of subjective information- such as linguistic-based descriptions- with the usual probabilistic or numerical type information. In conjunction with this effort, a literature search was conducted for previous work in this area. In addition to the premier collection of unclassified C^2/C^3 work- these Proceedings over the past eight years- other unclassified sources were also considered, including IEEE publications, Operations Research journals, Psychology publications, and separately published papers and books, among others. A brief survey of that portion of the literature relevant to the task here is presented in the next section. In section 3, general models of warfare and C^3 systems are proposed in the form of networks whose nodes represent decision makers/followers. These networks are also assumed to be time-varying. Section 4 is an abridged analysis of intranodal behavior, utilizing both probabilistic and possibilistic processes, analogous to the previous established PACT (Possibilistic Approach to Correlation and Tracking) program in Ocean Surveillance [55].

2. BRIEF SURVEY OF RELEVANT C^3 WORK

A now extensive C^3 and related discipline literature exists solely within the first seven annual Proceedings of this journal (283 articles). Perhaps because of the great complexity of the overall C^3 problem, relatively few papers have been written establishing quantitative models of generic C^3 systems. Of course, this does not detract from the progress made for various aspects of the problem proper and for related issues. Foremost among the latter is Surveillance, and in particular, multi-target tracking and data association. To a lesser degree, Data Base Management and Communications within C^3 systems have also been extensively treated quantitatively. Similarly, limited portions of the C^3 problem proper have been thoroughly analyzed- under appropriate simplifying conditions- including command decision theory, viewed as a possible multiple player statistical decision game involving, typically, threat situations and system effectiveness reflected in the loss or objective functions, as e.g. in [1], or considering players' mental images of one another together with limited knowledge of rules of play , as in [2]. In a similar vein, distributed or decentralized decision theory appears to be a valuable tool for analyzing C^3 systems which may be spread out geographically or otherwise have loose communications structures. (See, e.g., Tenney [3]-[4'] and Sandell [5] for basic results in this direction.) Complexity of distributed decision problems relative to C^3 was presented in [6] in the form of NP-completeness. Other general results, including asymptotic forms, may be found in Tsitsiklis' general work [7].

Hierarchical games and systems were used as models for parts of C^3 systems by Castanon [8] and others [9]. Later, Castanon [10] applied rational aggregate theory to linear dynamic state processes to obtain sequential (relative to hierarchy level) solutions of systems with hierarchies defined by behavior tempo having also possible uncertain models. (See also Luh et al. [11] for other aspects of hierarchical systems useful in C^3.) Often, C^3 systems have been defined as essentially involving the management of military resources. In conjunction with this, a number of papers have considered resource allocation techniques([12],[13], e.g.) as the prime characterization of C^3 systems. In addition, as mentioned numerous times, C^3 analysis requires multidisciplinary usage. For example, Control Theory could be thought of as central to the problem([13'],e.g.). Many papers have concentrated on the human decision maker-in-the-loop aspect, as a perusal of the last two Proceedings of this journal will show. Such papers can vary in thrust of analysis from input-output node models [14] to various detailed (some, qualitative, others, quantitative in scope) internally analyzed systems as in [15] or Wohl's and others' extended SHOR(Sense, Hypothesize,Option,Response) paradigms [16]-[18], related to Lawson's proposals [19],[20].

Although- as mentioned above- few papers have attempted to analyze the overall C^3 problem quantitatively or qualitatively, those that have, have engendered much controversy. Consider first those qualitatively oriented papers attempting to define or analyze general C^3 systems. Lawson [19],[20] was among the first to propose a general theory of C^3, based to a degree on analogues with thermodynamic principles, motivated by the classic Lanchester equations of force attrition or increase. Later, he emphasized time as a critical factor in all aspects of a C^3 system [21], considered briefly C^3 sys-

tems from a knowledge-based systems viewpoint, among other items [22], and proposed generic experiments for analyzing C^3 systems [23]. Athans also has been active in attempting to define the general C^3 problem, beginning with the First Workshop [24]-[26] and culminating with his view of "expert team of experts" for commanders [27]. Other good qualitative overviews of the problem may be found in [28]-[31'] as well as the short paper [32]. See also the more recent comments of Rona [33] and Metersky [34]. The latter emphasizes expanding Lawson's and others' concepts of C^3 and the integration in some systematic way of subjective and objective information. (This is compatible with section 4 here.) Strack [35] has compiled possibly the most far-reaching of qualitative analyses of C^2 problems in his recent report. In a related direction, development of measures-of-effectiveness (MOE's) for C^3 systems in general began in earnest with Lawson's concern for time/tempo of C^3 operations (such as in [21]) and Harmon and Brandenberg working on internodal and intranodal measures, among other topics [36]. Further work in this area has been carried out by Bouthonnier and Levis [37] (in conjunction with Levis' organizational approach - see below), Linsenmayer's countermeasure-oriented MOE paper [38], and recently, by Karam and Levis [39].

Recently, two additional approaches have been proposed for modeling general C^3 systems, which like Lawson's earlier proposals are most appropriate for large scale system behavior of C^3 components typically representing men in the field and supplies. Anthony [40] proposes four candidate, empirically-derived laws arising from other disciplines as governing C^3 systems. Mayk [41], somewhat similar to Lawson [1], presents a thermodynamics/uncertainty principle approach which regulates the more "irreducible primitive" components of C^3 systems. In addition, Rubin [42], following guidelines in [41], under semi-Markov and Markov assumptions, derived explicit forms for various stochastic processes acting as links among the components of a C^3 system. In particular, Lanchester's equations were shown to be a special case of this model.

The approach taken in this paper (section 3) follows to a degree the general view of Levis et al. [43]-[49]. There, a C^3 system is considered to be a collection of interacting decisionmakers, which as a whole, may follow (under appropriate limiting conditions) macroscopic principles (such as Lawson proposes, e.g.). However, critical to the analysis is the microscopic analysis of each decision maker or node representing a unit of decisionmakers acting through cooperation as a single individual. The structure of each decisionmaker follows the general pattern as the SHOR paradigm or variations. Then a quantitative (normative-descriptive) measure is obtained for each such decisionmaker in the form of the total workload-i.e., entropy-of all internal random variables connected with decision/action and choice of related algorithms, involving also possible interaction with other decisionmakers during this process, as well as accounting for memory. By simple summation over all decisionmakers, an overall C^3 system measure of workload G can be obtained. Alternatively, the overall joint workload can also be used. Another overall performance measure J is assumed obtainable, such as effectiveness of overall system in dealing with the enemy, so that both G and J are assumed to be dependent functionally - in a computable manner - on W, the internal variable strategies of the decisionmakers. Thus, possible tradeoffs or optimizations of G and J can be consider relative to W, subject to natural constraints on W resulting e.g. from bounded rationality involving G(W) and/or satisficing conditions connected with J(W).

The problem of processing and integrating subjective or linguistic-based information occurring within or between decision nodes and stochastic-based data in C^3 systems is an extension of that for the surveillance problem. In both situations, it may not be appropriate to model both types of information stochastically. In place of this, a possibilistic or multi-valued logic-based analysis may be the proper choice. (See [50] for motivations, background, and further details.) Zadeh [51],[52] originally proposed in these Proceedings use of possibilities in place of probabilities only, for decisions that could typically occur in a C^3 system. Similarly, Goodman employed such an approach- tying it in also with the coverage and incidence functions of stochastic set processes (i.e., random sets)- in addressing the data association problem in tracking [53]-[55]. Other approaches to the modeling of subjective information that occurs for C^3 systems have used forms of expert knowledge-based systems [56],[57]. Still others have considered use of neural network theory and the related area of self-organizing systems for C^3 analysis such as H. Szu has done at the most recent (8^{th})MIT/ONR Workshop. (See also [58],[59] for background.)

3. OUTLINE OF A C^3 THEORY

This section outlines a C^3 theory which to some extent follows the spirit of Levis et al. ([46],e.g.) in considering a C^3 system dependent upon its local behaviors and analyzing the latter. (See also the discussion in section 2.)

First consider a warfare process. A warfare process V is a time-indexed process given for convenience as

$$V \triangleq (V_t)_{t\geq 0} \quad , \tag{3.1}$$

where each V_t represents the overall warfare situation for some prescribed region at time t. (Note, that the term "process" and likewise all variables to be introduced below are to be interpreted in possibilistic terms in general, not necessarily probabilistic. Again, see [50] or [52] for background.) In turn, each warfare situation consists of a collection of C^3 systems

$$V_t \triangleq \{C_{t,j} \mid j \text{ in } K_{t,1}\} \ , \tag{3.2}$$

where $K_{t,1}$ is some index set and each $C_{t,j}$ is some C^3 system of interest. These C^3 systems may in a sense (to be explained) overlap, be subsets of each other, or be disjoint, reflecting both the design of the individual systems and the choice of levels of analysis. V_t can be partitioned into

$$V_t = \bigcup_{j \text{ in } K_{t,2}} (V_{t,j}) \text{ (disjointly)} \tag{3.3}$$

where $K_{t,2}$ is the index set of adversaries in conflict,

$$V_{t,j} \triangleq \{C_{t,j'} \mid j' \text{ in } K_{t,1,j}\} \ , \tag{3.4}$$

and

$$K_{t,1} = \bigcup_{j \text{ in } K_{t,2}} (K_{t,1,j}) \tag{3.5}$$

is a corresponding decomposition of index sets.

Often,

$$K_{t,2} = \{1,2\} \ , \tag{3.6}$$

where 1 represents friendly forces and 2 that of hostile ones.

In general, each C^3 system is represented as a type of network through the following ordered quadruple:

$$C_{t,j} \triangleq (N_{t,j}, I_{t,j}, \bar{O}_{t,j}, M_{t,j}), \tag{3.7}$$

where

$$N_{t,j} \triangleq \{N_{t,j,k} \mid k \text{ in } K_{t,3,j}\} \tag{3.8}$$

is the set of all nodes of the network;

$$I_{t,j} \triangleq \{I_{t,j,k} | k \text{ in } K_{t,3,j}\} \qquad (3.9)$$

is the set of all inputs (at t) of the network;

$$\bar{O}_{t,j} \triangleq \{\bar{O}_{t,j,k} | \ k \text{ in } K_{t,3,j}\} \qquad (3.10)$$

is the set of all outputs (at t) of the network; and

$$M_{t,j} \triangleq \{M_{t,j,j',k'} | k \text{ in } K_{t,3,j}, (j',k') \text{ in } K_{t,3,j,k}\} \qquad (3.11)$$

is the set of all media/environment /noise involving any node in the network with any other node (of any other network), where $K_{t,3,j}$ is the index set of all nodes for $C_{t,j}$ and $K_{t,3,j,k}$ is an index set representing those possible nodes outside of $N_{t,j,k}$ to which an initial output can be directed (whether on purpose or due to general radiation patterns, distances, etc.). Thus

$$K_{t,3,j,k} \subseteq \{(j',k') | j' \text{ in } K_{t,2}, k' \text{in } K_{t,3,j'}\} \qquad (3.12)$$

and

$$\bar{O}_{t,j,k} \triangleq \{\bar{O}_{t,j,k,j',k'} | (j',k') \text{ in } K_{t,3,j,k}\} \qquad (3.13)$$

is the decomposition of the output at node $N_{t,j,k}$ into possible outputs directed towards other nodes (for all adversaries.)

Hence, $(I_{t,j,k}, \bar{O}_{t,j,k})$ is the input-output pair for node $N_{t,j,k}$ at t. But the causal or semi-causal relation between inputs and outputs is given as: $I_{t_1,j,k}$ resulting in $\bar{O}_{t_2,j,k}$, for some $t_2 \geq t_1$ through

$$N_{t_1,t_2;j,k} \triangleq (N_{t,j,k})_{t_1 \leq t \leq t_2}, \qquad (3.14)$$

due to processing delays within the node, as some version of the SHOR paradigm is carried out interacting possibly with other decisionmakers, etc. In (3.13), each $\bar{O}_{t,j,k,j',k'}$ is that output from $N_{t,j,k}$ directed towards $N_{t,j',k'}$ through medium $M_{t,j,k,j',k'}$. Thus, typically, the additive-like regression relation holds (where again, note that the values involved may be non-numerical in nature - hence the use of $\oplus$)

$$I_{t_2,j',k'} = f_{t_1,t_2;j,k,j',k'}(\bar{O}_{t_1,j,k,j',k'}) \oplus R_{t_1,t_2;j,k,j',k'}, \qquad (3.15)$$

where f represents some function and R some noise, where possibly the constraint

$$t_1 \leq t_2 \leq t_1 + \Delta t_1 \qquad (3.16)$$

holds.

Next, each node is internally represented as an ordered quadruple

$$N_{t,j,k} \triangleq (S_{t,j,k}, \hat{S}_{t,j,k}, \hat{CS}_{t,j,k}, D_{t,j,k}), \qquad (3.17)$$

where $S_{t,j,k}$ is the true state vector of $N_{t,j,k}$, possibly unknown to the decisionmaker complex $D_{t,j,k}$ of $N_{t,j,k}$ and evolving in time according to possibilistic, or, in particular, probabilistic transition values. Typically, $S_{t,j,k}$ can contain entries (possibly decoupled) for: location and pattern of deployment of individuals within $N_{t,j,k}$, number of personnel there, equation of motion parameter values for that portion of the node involved in movement or going to battle, and weapon descriptions, if any weapons are present at the node. Similarly, $\hat{S}_{t,j,k}$ is the node's estimate of its own state, while $\hat{CS}_{t,j,k}$ is the node's estimate of all remaining relevant state vectors outside of the node. Finally, $D_{t,j,k}$ need not be a decisionmaker(s) in the narrow sense, but may also indicate a follower complex (such as a unit of soldiers ready for combat and following command orders). Use of $D_{t,j,k}$, possibly with $\hat{S}_{t,j,k}$ and $\hat{CS}_{t,j,k}$, if not vacuous, leads to the basic input-output mentioned around eq.(3.14). (One aspect of this will be given in section 4.)

Overall (real- or vector-valued) performance measures $J_{t,j,1}, J_{t,j,2}, \ldots$, can be constructed for each C^3 system $C_{t,j}$, generally through some function, such as addition, numerical averaging, or retaining the joint form of local performance measures at each node. Thus, e.g., one could have

$$J_{t,j,5} \triangleq \sum_{k \text{ in } K_{t,3,j}} (J_{t,j,5,k}), \qquad (3.18)$$

where each $J_{t,j,5,k}$ is considered a function of the internal decision variable possibility functions of $D_{t,j,k}$ through the relation

$$J_{t,j,5,k} = \text{'E'}\,(\mathcal{J}_{t-\Delta,t;j,5,k}(W_{t-\Delta t;j,k} | I_{t-\Delta t;j,k})), \qquad (3.19)$$

where $W_{t-\Delta t;j,k}$ is a collection of internal variables of $D_{t,j,k}$ operating over time interval $[t-\Delta,t]$; similarly for the inputs $I_{t-\Delta t;j,k}$; and where $\mathcal{J}$ is an appropriately chosen function. Quote marks surround the expectation since possibility functions may be involved, in which case a possibilistic measure of central tendency replaces ordinary probabilistic expectation [50].

Thus, as mentioned earlier, one can then determine tradeoffs between various performance measures of a given $C_{t,j}$ or even of V_t through admissible possibility functions, here, corresponding to W.

It is of some interest to determine if under reasonable conditions, as the number of nodes increase indefinitely, behaving in some "random" manner, that the proposed thermodynamic-type C^3 models can be obtained as limiting cases of the model presented here. At present, work is being carried out in this direction. Further details of the general theory presented here will be presented in a later publication. For the present, analysis will concentrate on intranodal use of subjective and objective information, in order to obtain the basic input-output equations.

4. COMBINATION OF EVIDENCE AT NODES

In this section, some quantitative results are derived for intranodal behavior of a C^3 system.

Consider any node $N_{t-\Delta,t;j,k}$ with internal variable set $W_{t-\Delta t;j,k}$ and possible additional input set of variables during processing time $[t-\Delta,t]$, $\bar{I}_{t-\Delta t;j,k}$, as well as original input set $I_{t-\Delta,j,k}$. Without loss of generality, suppose subjective components of the relevant quantities below are indicated by primes as superscripts, while objective/probabilistic ones are denoted by superscripted double primes:

$$\bar{I}_{t-\Delta,t;j,k} \triangleq (\bar{I}'_{t-\Delta,t;j,k}, \bar{I}''_{t-\Delta,t;j,k}) \quad , \quad (3.20)$$

$$W_{t-\Delta,t;j,k} \triangleq (W'_{t-\Delta,t;j,k}, W''_{t-\Delta,t;j,k}) \quad , \quad (3.21)$$

$$I_{t-\Delta,j,k} \triangleq (I'_{t-\Delta,j,k}, I''_{t-\Delta,j,k}) \quad . \quad (3.22)$$

Furthermore, since no time integration will be carried out here (under simplistic assumptions for the current analysis), drop all subscripts in the above equations.

Following the development in [53],[54], all probabilistic information is modeled through some discretization/refinement level of probability density functions i.e., finite probability functions, while all subjective information is treated by possibility functions, which in general are not probability functions (not adding up to unity, since overlapping and vague concepts are being represented [50]). Thus both types of information are now modeled by possibility functions and may be manipulated through finite argument multi-valued logical operators. In particular, conjunction, replacing product for ordinary probability functions, is represented by a large class of operators, the t-norms, which include, as a special case, product. Similarly, disjunction extends the ordinary sum operator relative to probabilities and is represented through the class of t-conorms. Finally, negation or set complements is generalized by use of negation operators which include the more familiar classical operator 1-(·). (Again, see [50] for details.) More specifically, a t-norm $\phi_{\&}:[0,1]^n \to [0,1]$ is non-decreasing in all arguments, continuous, symmetric, associative (so that it may be extended recursively, unambiguously from n=2 arguments to an arbitrary number of), and a t-conorm $\phi_{or}:[0,1]^n \to [0,1]$ has formally the same properties, where both satisfy the boundary conditions for all $0 \le x,y \le 1$, for n=2 (the general case being similar)

$$\phi_{\&}(x,y) \le \min(x,y); \max(x,y) \le \phi_{or}(x,y) \; ; \quad (3.23)$$

$$\phi_{\&}(0,x)=0 \; ; \phi_{\&}(1,x)=x=\phi_{or}(0,x) \; ; \; \phi_{or}(1,x)=1 \quad . \quad (3.24)$$

Also, following the notation in (3.20)-(3.22) and the ensuing remarks, denote for probabilistic and subjective variables involved internally as

$$Z' \triangleq (W', \bar{I}') \quad ; \; Z'' \triangleq (W'', \bar{I}'') \quad . \quad (3.25)$$

It follows that analogous to ordinary probability function relations, denoting possibilities by ϕ [50], and finally noting that $\bar{O}$ as used here is an abbreviation for output $\bar{O}_{t,j,k}$,

$$\phi(\bar{O},W,\bar{I}|I)=\phi_{\&}(F(Z',Z''|I),\phi(Z'|Z'',I)) \quad , \quad (3.26)$$

where

$$F(Z',Z''|I) \triangleq \phi_{\&}(\phi(\bar{O}|W,\bar{I},I),\phi(Z''|I)) = \phi_{\&}(\phi(\bar{O}|Z',Z'',I),\phi(Z''|I)). \quad (3.27)$$

Denote the discretization/refinement (including truncation, if needed) level by index p, so that from the above discussion, replace $\phi(Z'|Z'',I)$ by

$$\phi_p(Z'|Z'',I)=f(Z'|Z'',I)\cdot\Delta_p(Z'), \quad (3.28)$$

where $f(\cdot|\cdot\cdot)$ is a fixed p.d.f., not depending on p, and where the domain of f , assumed to be, say, $\mathbb{R}^m$, ϕ_p has a finitely discrete domain D_p' , so that in any natural sense

$$\lim_{p\to\infty} D_p' = \mathbb{R}^m \quad ; \; \lim_{p\to\infty} \Delta_p(Z') = 0 \text{ (uniform)}, \quad (3.29)$$

In turn, it follows that

$$\phi_p(\bar{O}|I)= \phi_{or}\,(\underset{\text{(all } W,\bar{I})}{}\phi_p(\bar{O},W,\bar{I}|I)) = \phi_{or''}(\underset{\text{(all } Z'')}{}\phi_{or'}\,(\underset{\text{(all } Z')}{}\phi_p(\bar{O},Z',Z''|I))) \quad , \quad (3.30)$$

where as in (3.26),(3.27)

$$\phi_p(\bar{O},Z',Z''|I)=\phi_{\&}(F(Z',Z''|I),\phi_p(Z'|Z'',I)) \quad (3.31)$$

and it is assumed that ϕ_{or} represents a compound combination of $\phi_{or'}$ applied to probabilistic information followed by $\phi_{or''}$ applied to subjective information. In general, the two t-conorms may be different([50],Chp. 10).

With all of this established, the basic question arises as to the behavior of $\phi_p(\bar{O}|I)$ as more and more of the probabilistic information is used in terms of the discretization procedure, i.e., what is $\lim_{p\to\infty} \phi_p(\bar{O}|I)$?

The following theorem has an analogue for the PACT application([50],Chp.9); but differs somewhat in structure from the forms presented there.

Theorem

Suppose that all constructions hold as presented in (3.28) for any index p, where for convenience f is assumed to be also bounded. Suppose also the following:

1. $\phi_{\&}$ as a function of two arguments possesses continuous second order derivatives in some neighborhood of (0,0).

2. $\phi_{or'}$ is an Archimedean t-conorm, i.e., for the two argument case, for example,

$$\phi_{or'}(x,x) > x \text{ ,all } 0<x<1 \quad . \quad (3.32)$$

(Many t-conorms are Archimedean and indeed it can be shown that arbitrary t-conorms can be written as affine types of mixtures (called ordinal sums) of Archimedeans and the non-Archimedean t-conorm max. Again, see[50] - Chp. 2.3.)

3. The corresponding generating function h to $\phi_{or'}$ (see Proof below for discussion) has a continuous second order derivative in some neighborhood $[1-\varepsilon_1,1]$ of 1 , $0<\varepsilon_1<1$.

Then:

$$\lim_{p\to\infty} \phi_p(\bar{O}|I) = \phi_\infty(\bar{O}|I) \triangleq \phi_{or''}(\underset{\text{(all } Z'')}{}\omega(Z''|I)), \quad (3.33)$$

where nondecreasing function ω is given in (3.46) in terms of the ordinary expectation of also nondecreasing function κ of F(Z',Z''|I), with respect to(Z'|Z'',I) now formally a random vector corresponding to p.d.f. f ; κ is given in (3.40).

Proof:

A. A relatively deep theorem from the theory of probabilistic metric spaces [60] shows first that any given Archimedean t-norm, say $\phi_{\&'}$, has an essentially unique generating function $h:[0,1] \to \bar{\mathbb{R}}^+$, where $\bar{\mathbb{R}}^+$ denotes the positive real line with $+\infty$ annexed. That is, h is continuous nonincreasing with

$$h(1)=0 \; ; \; h(0) \le +\infty \quad (3.34)$$

such that for all positive integers n and all $0 \le x_1,\ldots,x_n \le 1$,

$$\phi_{\&'}(x_1,\ldots,x_n) = h^{-1}(\min(h(0), \sum_{j=1}^{n} h(x_j))). \quad (3.35)$$

The definition of an Archimedean t-norm is dual to that in (3.32) :

$$\phi_{\&'}(x,x) < x \quad , \text{ all } 0<x<1 \quad . \quad (3.36)$$

Although any pair of t-norm and t-conorm need not be DeMorgan, any t-conorm can be expressed as the DeMorgan transform of some corresponding t-norm. Furthermore, if one is Archimedean, then so is its DeMorgan transform. Thus, one can let $\phi_{or'_n}$ in (3.32) be written as , for all positive integers n ,etc.

$$\phi_{or'_n}(x_1,..,x_n)=1-\phi_{\&'_n}(1-x_1,..,1-x_n)$$
$$= \psi_h\Big(\sum_{j=1} h(1-x_j)\Big) , \quad (3.37)$$

where

$$\psi_h(x) \triangleq 1-h^{-1}(\min(h(0),x)),\text{all } 0\le x, \quad (3.38)$$

using assumption 2.

B. From assumptions 1 and 3 ,

$$\phi_{\&}(x,y) = \kappa(x)\cdot y + O(y^2;x) \quad (3.39)$$

where

$$\kappa(x) \triangleq (\partial\, \phi_{\&}(x,y) / \partial y)_{y=0} ; \quad (3.40)$$

and

$$h(1-z) = c_h\cdot z + O(z^2) , \quad (3.41)$$

where

$$c_h \triangleq -(dh(z)/dz)_{z=1} > 0 , \quad (3.42)$$

$O(\cdot)$ denotes the usual "order of" relation, and x,y,z are arbitrary such that for $\varepsilon_1,\varepsilon_2$ fixed

$$0\le x,y\le\varepsilon_2\le 1 \; ; \; 1-\varepsilon_1\le z\le 1 . \quad (3.43)$$

C. For any $0\le x_1,..,x_n\le\varepsilon_1$, using (3.37) and (3.41),

$$\phi_{or'_n}(x_1,..,x_n)=\psi_h\Big(c_h\cdot \sum_{j=1}^{n}(x_j+O(x_j^2))\Big). \quad (3.44)$$

Apply (3.39) to (3.31) and (3.28), and then replace each x_j in (3.44) by $\phi_p(\bar{0},Z',Z''|I)$ with index Z' in D'_p replacing j, $j=1,..,n$. This yields

$$\phi_{or'_n}(\phi_p(\bar{0},Z',Z''|I))=\psi_h\Big(c_h\cdot \sum_{(Z' \text{ in } D'_p)}\big(\kappa(F(Z',Z''|I))\phi_p(Z'|Z'',I)\big)$$
$$(\text{all } Z' \text{ in } D'_p)$$
$$+ O\Big(\sum_{Z' in D'_p}\Delta_p(Z')^2\Big)\Big), \quad (3.45)$$

The main result then follows using (3.45) in (3.30), where

$$\lim_{p\to\infty}\Big(\phi_{or'_n}(\phi_p(\bar{0},Z',Z''|I))\Big) \triangleq \omega(Z''|I)$$
$$= \psi_h\Big(c_h\cdot E_{(Z'|Z'',I)}(\kappa(F(Z',Z''|I))|Z'',I)\Big) , \quad (3.46)$$

Thus, up to essentially increasing transforms, the approach taken here to combining subjective information with objective information involves taking an expectation of the latter and using a multi-valued logical procedure on the former in a unified way, up to the level of discretization/refinement used for the probabilistic information. As a final remark, it should be noted that most common t-norms and t-conorms satisfy the rather mild analytic conditions required in the hypotheses of the theorem and max can also be used in place of an Archimedean form for $\phi_{or'_n}$, with appropriate modifications of the proof. (See Appendix A for an important example of this.)

5. CONCLUSIONS

An attempt at modeling the overall C^3 problem as a network of nodes has been outlined. Key to this is the local modeling, i.e., the modeling of input-output behavior at each node. A procedure was presented, analogous to the PACT algorithm for multi-target data association which treats in a unified manner subjective and objective information. Future efforts will elaborate further on both global and local aspects of combining such information.

ACKNOWLEDGEMENTS

This work has been supported in part by the NOSC IR (Independent Research) and IED (Independent Exploratory Development) Programs. In addition , the author wishes to express appreciation to M.C. Mudurian of NOSC, Code 421, for his many valuable suggestions and stimulating discussions.

REFERENCES

In the following references, abbreviations are made for the Proceedings of the MIT/ONR Workshop on C^3 Systems by use of the notation $C^3(k)$ or $C^3(k,j)$, where k corresponds to the number of the Proceedings and j denotes the volume (in Roman numerals), if more than one was issued. Years are omitted for such references, noting the correspondences:

1978 ↔ k=1	1981 ↔ k=4	1984 ↔ k=7
1979 ↔ k=2	1982 ↔ k=5	1985 ↔ k=8
1980 ↔ k=3	1983 ↔ k=6	

1. Castanon,D.A.,Delaney,J.R.,Kramer,L.C.,Athans,M."A mathematical framework for the study of battle group position decisions", $C^3(5)$, 105-110.

2. Teneketzis,D.,Castanon,D.A.,"Information aspects of a class of subjective games of incomplete information", $C^3(6)$, 226-231.

3. Tenney,R.R., Distributed Decision Making Using a Distributed Model, Ph.D.Diss.(LIDS-TH-938),Sept.,1979.

4. Tenney,R.R.,"Distributed decision making with limited communication", $C^3(2,IV)$, 830-868.

4. Tenney,R.R.,"A case study of distributed decision making", $C^3(4,IV)$, 235-244.

5. Sandell,N.R., "Distributed decision making processes in C^3", $C^3(3,II)$, 1-38.

6. Tsitsiklis,J.,Athans,M.,"On the complexity of distributed decision problems", $C^3(6)$, 232-237.

7. Tsitsiklis,J.N., Problems in Decentralized Decision Making and Computation,Ph.D.Diss.(LIDS-TH-1424),Dec,1984.

8. Castanon,D.A.,"Goal coordination for hierarchical structures in game theory",$C^3(3,II)$, 155-184.

9. Papavassilopoulos,G.P., "Directions and recent results in hierarchical decision making via game theoretic methods", $C^3(3,II)$, 185-198.

10. Castanon,D.A., "Games with uncertain models", $C^3(4,IV)$, 61-80.

11. Luh,P.B.,Ning,T.,"Dynamic, hierarchical decision problems", $C^3(6)$, 249-254.

12. Lauer,G.S.,Bertsekas,D.,Sandell,N.R.,"Solution of large dynamic resource allocation problems",$C^3(3,III)$,171-224.

13. Loparo,K.,Walker,B.,Griffiths,B.,"A framework for the design of survivable systems-Part II:Control and information structure", $C^3(4,IV)$, 295-308.

13. Tenney,R.R.,"Optimal decentralized control of finite deterministic systems", $C^3(6)$, 238-242.

14. Sen,P.,Drenick,R.F.,"Information processing in man-machine systems", $C^3(4,IV)$, 81-102.

15. Kleinman,D.L.,Pattipati,K.B.,"Results toward developing a model of human decision making in C^3 systems", $C^3(3,III)$, 29-64.

16. Wohl,J.G.,"Battle management decisions in Air Force Tactical Command", C3(3,II), 87-126.

17. Wohl,J.G.,Entin,E.E.,"Modeling human decision processes in Command and Control",C3(5), 124-126.

18. Wohl,J.G.,"Rate of change of uncertainty as an indicator of Command and Control Effectiveness",C3(4,IV),1-18.

19. Lawson,J.S.,"A unified theory of Command and Control", C3(1), 10-125.

20. Lawson,J.S.,"The state variables of a Command Control system", C3(2,II),462-510.

21. Lawson,J.S.,"The role of time in a Command Control system, C3(4,IV), 19-60.

22. Lawson,J.S.,"Data bases and decisions",C3(5), 1-7.

23. Lawson,J.S.,"Doing C2 experiments using war games", C3(6), 3-9.

24. Athans,M.,"Systems aspects of C3 problems",C3(1),126-141.

25. Athans,M.,"Overview of MIT/LIDS research in Naval C3 systems", C3(2,I), 1-34.

26. Athans,M.,"System theoretic challanges and research opportunities in military C3 systems, C3(3,I), 1-8.

27. Athans,M.,"The expert team of experts approach to C2 organizations", C3(4,IV), 207-234.

28. Brick,D.B.,Jarvis,T., "C3:systems,functions, problems,..", C3(1), 142-214.

29. Blinkenberg,R.,"Command,Control, and Communication (C3) design implications for distributed data analysis and fusion activities", C3(2,I), 134-157.

30. Schutzer,D.M.,"Command,Control and Communications-some design aspects", C3(2,III), 741-782.

31. Shanahan,E.J., Teates,H.B.,Wise,B.B.,"Defining and measuring C2", C3(3,III), 229-282.

31. Tenney,R.R.,"Modelling the C3 decision process", C3(3,II), 39-80.

32. Conley,R.E.,"A command and control theory",C3(3),81-86.

33. Rona,T.P.,"C3 over the past 3-5 years- a personal learning experience", C3(5), 8-13.

34. Metersky,M.E., "A C2 process and an approach to design and evaluation", C3(7), 11-22.

35. Strack,C.W.,Elements of C2 Theory, Report for DSI (McLean,Virginia), Jan. 30, 1985.

36. Harmon,S., Brandenburg,R., "Command, Control and Communications (C3) systems model and measures of effectiveness (MOE's)", C3(4,IV), 181-206.

37. Bouthonnier,V.,Levis,A.H., "Effectiveness analysis of C3 systems", C3(5), 80-87.

38. Linsenmayer,G.R., "Measures-of-effectiveness for Command, Control, and Communications Countermeasures", C3(5), 88-93.

39. Karam,J.,Levis,A.H., "Effectiveness assessment of the METANET demonstration", C3(7), 61-70.

40. Anthony,R.W.,"Holistic patterns in command,control, communication and information systems".C3(7), 71-78.

41. Mayk,I., Rosenstark,S., Frank,J,, "Analysis of C3 systems based on a proposed canonical reference model", C3(7), 45-54.

42. Rubin,I.,Queueing and Decision Theoretic Modeling and Mathematical Analysis of C3 Systems and Networks , IRI Report No. IRI-CEC-REP-8501 (Tarzana,Cal.) Mar.,1985.

43. Levis,A.H.,Boettcher,K.L.,"Organization theory and C3 systems", C3(3,II), 127-154.

44. Boettcher,K.L., Levis, A.H., "Modeling the interacting decisionmaker with bound rationality",C3(4,IV),103-136.

45. Levis,A.H.,Boettcher,K.L.,"Decisionmaking organizations with acyclical information structures", C3(5), 94-104.

46. Levis,A.H.,"Information processing and decisionmaking organizations: a mathematical description", C3(6), 30-38.

47. Hall,S.A., Levis,A.H.,"Information theoretic models of memory in human decisionmaking models",C3(6),67-75.

48. Tomovic,M.M., Levis,A.H., "On the design of organizational structures for command and control", C3(7) , 131-138.

49. Tabak,D., Levis,A.H., "Petri net representation of decision models", C3(7), 121-130.

50. Goodman,I.R., Nguyen,H.T., Uncertainty Models for Knowledge-Based Systems, North Holland Co., 1985.

51. Zadeh,L.A.,"Possibility theory and its applications to decision analysis", C3(1), 215-256.

52. Zadeh,L.A., "Fuzzy probabilities and their role in decision analysis", C3(4,IV), 159-180.

53. Goodman,I.R., "An approach to the data association problem through possibility theory", C3(5), 209-215.

54. Goodman,I.R., "A unified approach to modeling of evidence through random set theory", C3(6), 42-47.

55. Goodman,I.R., "Modeling natural language information for use in the combination of evidence problem", C3(7), 173-178.

56. Aldrich,J.R.,"Applications of knowledge-based approaches to C2I processes", C3(3,III), 143-170.

57. Dillard,R.A., "Integration of Artificial Intelligence into tactical C3 systems", C3(7), 79-84.

58. Grossberg,S., Studies of Mind and Brain , Reidel Co., Boston, 1982.

59. Kohonen,T., Self Organization and Associative Memory, Springer-Verlag Co., Berlin, 1984.

60. Schweizer,B., Sklar,A., Probabilistic Metric Spaces North-Holland Co., 1983.

APPENDIX A.

A large and conveniently parameterized family of DeMorgan transform pair of Archimedean t-norms and t-conorms is due to Frank originally (see [50],Chp.2.3 for additional discussion) and satisfies uniquely the modular relation $\phi_{or}(x,y)=x+y-\phi_{\&}(x,y)$,all $0\le x,y\le 1$. (A-1)

The solution is given as, using parameter-index s,

$$\phi_{\&,s}(x_1,..,x_n)=\log_s\left(1+\left(\prod_{1\le j\le n}(s^{x_j}-1)/(s-1)^{n-1}\right)\right), \quad (A-2)$$

$$\phi_{or,s}(x_1,..,x_n)=1-\phi_{\&,s}(1-x_1,..,1-x_n), 0\le x_1,..,x_n\le 1, \quad (A-3)$$

where $0<s\le+\infty$; s otherwise any real number. In a limiting sense, it is natural to define for the non-Archimedean pair min, max , (A-4)

$$\phi_{\&,0}(x_1,..,x_n)=\min(x_1,..,x_n);\phi_{or,0}(x_1,..,x_n)=\max(x_1,..,x_n)$$

and to note the special cases s=1,s=+∞ in (A-2),(A-3):

$$\phi_{\&,1}(x_1,...,x_n)=\prod_{1\le j\le n}(x_j);\phi_{or,1}(x_1,..,x_n)=1-\prod_{1\le j\le n}(1-x_j), \quad (A-5)$$

$$\phi_{\&,+\infty}(x_1,..,x_n)=\max\left(\sum_{1\le j\le n}(x_j)-(n-1),0\right);\phi_{or,+\infty}(x_1,..,x_n)=\min\left(\sum_{1\le j\le n}(x_j),1\right). \quad (A-6)$$

It then follows that Frank's family satisfies the hypotheses of the theorem in section 3, for all s>0 , with s=0 also treatable as a special case. For all s>0, generator function h_s and (3.38),(3.40),(3.42) become:

$$h_s(x)=-\log((s^x-1)/(s-1));c_{h_s}=(s\log(s))/(s-1) ; \quad (A-7)$$

$$\psi_{h_s}(x)=1-\log_s(1+((s-1)\cdot e^{-x}));\kappa_s(x)=(s^x-1)/(s-1). \quad (A-8)$$

ON COMBINING UNCERTAIN MESSAGES USING BELIEF FUNCTIONS

A. P. Dempster and Augustine Kong

Department of Statistics
Harvard University
Cambridge, Massachusetts 02138

ABSTRACT

Information at a source t_0 is transmitted partially and uncertainly to one or more nodes $t_1, t_2, \ldots$ which in turn may transmit information to latter nodes, and so on. The nodes are assumed to form a tree with t_0 as its root. Belief functions are used to model the transmission mechanisms. An efficient algorithm for computing the beliefs about the original message at source t_0 given the messages received at the terminal nodes is provided.

1. INTRODUCTION

The elements of the theory of belief functions were set forth in Dempster (1967) and subsequently were extended and enriched in Shafer's 1976 book A Mathematical Theory of Evidence. Although Shafer (1981, 1982a, 1982b) has sought to define the set of attitudes and thought patterns which encompass the intended uses of the theory, the body of published material on applications remains very small. In this paper we study a model applicable to information networks of successive and concurrent messages which could arise in the gathering and processing of intelligence data. We start by reviewing the elements of the theory which are essential to model building.

1.1 Elements of the Theory

An ordinary probability model has two basic elements: the sample space and the probability measure over the sample space. For present purposes, there is no need to be concerned with mathematical details introduced by infinite sample spaces, so we suppose that the sample space

$$\Omega = \{\theta_1, \theta_2, \ldots, \theta_n\} \tag{1.1}$$

has n elements, and the probability measure over Ω is defined by the corresponding probabilities $p_1, p_2, \ldots, p_n$ where $0 \le p \le 1$ and $\sum_1^n p_i = 1$. The theory of belief functions depends on exactly the same form of mathematical structure, but with a twist which greatly expands its range of applicability.

Similar to ordinary probability models, belief function models are characterized by a sample space Ω and a probability assignment associated with Ω. Shafer adopted the term frame of discernment for Ω, indicating that Ω is deliberately constructed by the user of the technology to define the factual structure of some small world under analysis. Given a finite frame Ω, the completion of a belief function model requires an assignment of basic probability numbers m(A) for each $A \subset \Omega$, where

$$0 \le m(A) \le 1 \qquad \forall A \subset \Omega$$
$$m(\phi) = 0, \tag{1.2}$$

and

$$\sum_{A \subset \Omega} m(A) = 1.$$

Sets A such that $m(A) > 0$ are called the focal elements of the model. For any $B \subset \Omega$, we define

$$BEL(B) = \sum_{B \supset A} m(A). \tag{1.3}$$

The idea behind (1.2) and (1.3) is that m(A) is a piece of probability which can move freely among $\theta_i \in A$, whence BEL(B) is the total probability committed to $\theta_i \in B$ in the sense that it cannot move outside B. We call BEL(B) the belief in B or simply the probability of B. A companion to (1.3) is given by

$$PL(B) = \sum_{A \cap B \neq \phi} m(A) \tag{1.4}$$

which is the largest amount of probability which can move to some $\theta_i \in B$. Evidently $PL(B) \ge BEL(B)$ for all $B \subset \Omega$. PL(B) is called the plausibility of B. It is easily checked that $BEL(B^c) = 1 - PL(B)$ and $PL(B^c) = 1 - BEL(B)$ where B^c is the complement of B, or $B^c = \Omega - B$. Also, $BEL(\phi) = PL(\phi) = 0$ and $BEL(\Omega) = PL(\Omega) = 1$. For practical purposes, BEL(B) and PL(B) can be interpreted respectively as the lower and upper probability of B.

Example 1: Drawing a Ball from Urns

Consider two urns containing balls which are either red or white. All the balls in Urn I are red. Nothing is known about the proportions of red balls and white balls in Urn II. It is possible that all the balls are red or all the balls are white. A ball is to be drawn from one of the urns. There is a probability .7 that the ball will be drawn from Urn I and probability .3 that the ball will be drawn from Urn II. Our knowledge about the color of the ball to be drawn is represented by the belief function

$$m(\{R\}) = .7$$
$$m(\{R, W\}) = .3 \tag{1.5}$$

over the frame $\Omega = \{R, W\}$, where R and W stand for red and white, respectively. It follows that $BEL(\{R\}) = .7$, $PL(\{R\}) = .1$, $BEL(\{W\}) = 0$ and $PL(\{W\}) = .3$.

The special class of belief function models for which the focal elements are restricted to the class of single member subsets $A_i = \{\theta_i\}$ are equivalent to ordinary probability models. Definitions (1.3) and (1.4) now yield the same quantity for all B, so we may as well use P(B) instead of BEL(B) or PL(B). ■

A pair of frames Ω and Ω' such that the members of Ω' are related one-to-one with the members of a partition of Ω, both mathematically and logically, are respectively refined and coarsened relative to each other. That is, the more refined Ω is able to represent more detail than the coarser Ω' about the small world formally represented by the analysis. (An example will be a product space $\Omega = \Omega_1 \times \Omega_2$. The frame Ω is a

refinement of both Ω_1 and Ω_2. An element θ_1 of Ω_1 corresponds to the subset $\{\theta_1\} \times \Omega_2$ of Ω.) If a general belief function is constructed over Ω', then the minimal extension to Ω is defined by using the same probability numbers but replacing each focal element by the corresponding union of partition subsets of Ω. The frame Ω' is often called a margin of Ω. Starting from any belief function over Ω we may define a marginal belief function over Ω' whose BEL'$(\cdot)$ values for subsets $B' \subset \Omega'$ are identical to BEL$(\cdot)$ for $B \subset \Omega$ where B is the union of partition sets in B'. Thus, minimal extension and marginalization are both operations which allow us to propagate beliefs from one frame to another based on their logical relationships. The difference is that we can always recover a belief function from its minimal extension by marginalization, but it is not always possible for us to reconstruct a belief function from its margins.

The basic technical device giving the theory scope to represent complex information structures is the direct sum operation used for combining independent sources of evidence. Suppose that BEL_1 and BEL_2 are belief functions over the same Ω with basic probability numbers denoted by m_1 and m_2, respectively. The direct sum BEL of BEL_1 and BEL_2, or

$$BEL = BEL_1 \oplus BEL_2 \tag{1.6}$$

is also a belief function over Ω whose basic probability numbers m are defined as follows from m_1 and m_2. Suppose that $P(A)$ denotes the class of pairs (A_1, A_2) where A_i is a focal element of BEL_i for $i = 1, 2$ and $A = A_1 \cap A_2$, and suppose that $m_\oplus$ is defined to be

$$m_\oplus(A) = \sum_{P(A)} m_1(A_1)\, m_2(A_2). \tag{1.7}$$

Then A is defined to be a focal element of BEL = $BEL_1 \oplus BEL_2$ if and only if $P(A)$ is nonempty and $A \neq \phi$, and the basic probability numbers for BEL are given by

$$m(A) = m_\oplus(A)/(1 - m_\oplus(\phi)). \tag{1.8}$$

Note that $m_\oplus(A)$ sums to unity over all $A \subset \Omega$, while m(A) sums to unity over all $A \neq \phi$ as required by (1.8). The reasoning behind (1.7) is that A_1 and A_2 are like random messages independently selected with probabilities $m_1(A_1)$ and $m_2(A_2)$ whose combined message is $A = A_1 \cap A_2$ according to ordinary Boolean algebra. Hence, the combined message A is like a random message with probability proportional to $m_\oplus(A)$ given by (1.7).

Example 1 Revisited.

Let us look at the urn example discussed earlier from a different angle. Assume that we are interested in both the color of the ball drawn and the urn the ball is drawn from. The natural frame of discernment is

$$\Omega = \{(I, R), (I, W), (II, R), (II, W)\} \tag{1.9}$$

which is the product of $\Omega_1 = \{I, II\}$ and $\Omega_2 = \{R, W\}$. For example, the pair (I,R) stands for the proposition that a red ball is drawn from Urn I. Our knowledge that the balls in Urn I are all red can be represented by the belief function

$$m_1(\{(I, R), (II, R), (II, W)\}) = 1. \tag{1.10}$$

Our knowledge about which urn the ball is going to be drawn from can be represented by a belief function over Ω_1 with basic probability numbers

$$\begin{aligned} m_2(\{I\}) &= .7 \\ m_2(\{II\}) &= .3, \end{aligned} \tag{1.11}$$

which can be minimally extended to Ω as

$$\begin{aligned} m_2(\{(I, R), (I, W)\}) &= .7 \\ m_2(\{(II, R), (II, W)\}) &= .3. \end{aligned} \tag{1.12}$$

Combining (1.10) and (1.12) gives us the belief function

$$\begin{aligned} m(\{(I, R)\}) &= .7 \\ m(\{(II, R), (II, W)\}) &= .3. \end{aligned} \tag{1.13}$$

Note that (1.5), the belief function we constructed earlier, is the marginal belief function of (1.13) over the frame $\Omega_2 = \{R, W\}$.

2. BELIEF FUNCTION ANALYSIS OF A GENERAL INFORMATION NETWORK WITHOUT LOOPS

Let Ω_0 be a finite set of propostions which are mutually exclusive and exhaustive. Messages regarding the true proposition, θ_0, are transmitted through a communications network. The goal is to construct a belief function over Ω_0 based on the observations of some of those messages. We limit ourselves to the study of communication networks resembling a one root-single parent-family tree. Terminologies needed for describing such a tree are developed below.

A tree is represented by a set of nodes $T = \{t_0, t_1, \ldots, t_N\}$. Each node stands for a specific member of a "family". The node t_0, the root of the tree, is the "ancestor" of all the other nodes. The second generation would be a set of nodes which are the "sons" of t_0. The sons may have their own sons, and so on, which builds up the overall structure of the tree. In general, a father may have any number of sons, but a son has exactly one father. A node which does not have any son is called a terminal node. Graphically, the father-son relationship is represented by a line connecting the father-node and the son-node. To avoid the confusion of mixing up the father and the son, the nodes are ordered in a way such that if t_i is the father of t_j, then $i < j$. Figure 2.1 displays a tree which has all the features described.

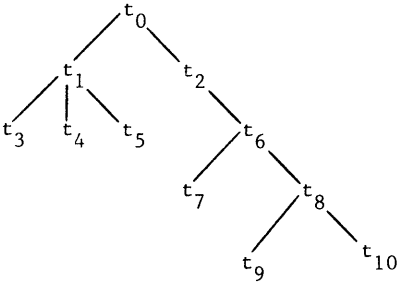


Figure 2.1

The terms "ancestor" and "descendant" would be used in the most obvious manner. For example, in Figure 2.1, t_6 is the ancestor of t_7, t_8, t_9, and t_{10}, and they are all his descendants. For notation, let

$$\begin{aligned} \tilde{T} &= \{t_i \mid t_i \text{ is a terminal node}\} \\ S(i) &= \{t_j \mid t_j \text{ is a son of } t_i\} \\ D(i) &= \{t_j \mid t_j \text{ is a descendant of } t_i\} \\ C(i) &= \{t_j \mid t_j \text{ is an ancestor of } t_i\} \\ \tilde{T}(i) &= \begin{cases} \{t_i\} & \text{if } t_i \in \tilde{T} \\ \tilde{T} \cap D(i) & \text{otherwise} \end{cases} \end{aligned}$$

Also we would let $t_{f(i)}$ to stand for the father of t_i. For example, in Figure 2.1, we have $\tilde{T} = \{t_3,t_4,t_5,t_7, t_9,t_{10}\}$, $S(1) = \{t_3,t_4,t_5\}$, $D(6) = \{t_7,t_8,t_9,t_{10}\}$, $C(7) = \{t_0,t_2,t_6\}$, $\tilde{T}(2) = \{t_7,t_9,t_{10}\}$ and $t_{f(6)} = t_2$.

The nodes of a tree represent agents who receive and transmit messages. The agent t_0 is a witness who knows exactly which proposition in Ω_0 is true. He transmits messages to agents who are members of the set S(0). These agents then transmit messages to other agents. In general, for $i > 0$, agent t_i receives a message from agent $t_{f(i)}$ and transmits messages to agents t_j, $t_j \in S(i)$. Agents t_i, $t_i \in \tilde{T}$, are assumed to be agents on our side and it is the messages received by them which we actually observed. We are interested in constructing the conditional belief function over Ω_0 based on the observation of these messages.

All the messages received are in the form of "the true proposition is in A" where A is a subset of Ω_0. Our overall frame of discernment will be the cross-product of Ω_0 and the message space. This can be represented by $\Omega = \Omega_0 \times 2^{\Omega_1} \times 2^{\Omega_2} \times \ldots \times 2^{\Omega_N}$, where Ω_i, $0 < i \leq N$, are isomorphic to Ω_0 and 2^{Ω_i} is the set of nonempty subsets of Ω_i. A typical point in this frame of discernment will be a vector $(\theta_0,A_1,A_2,\ldots,A_N)$ which represents the case where the true proposition is θ_0 and the message reviewed by agent t_i is "the true proposition is in A_i". To construct a belief function over Ω which would represent our knowledge about the transmission process as a whole, we start by constructing belief functions over its margins based on our knowledge about how individual messages are transmitted. We will then minimally extend these belief functions to Ω and combine them using the rule of combination.

We assume that each agent transmits messages based on the message he has received. Although agent t_0 does not actually receive a message from anyone, we may think of him as receiving a totally reliable message telling him that "the true proposition is in A_0" where $A_0 = \{\theta_0\}$. Based on the message received by agent $t_{f(i)}$, we assume that the message received by agent t_i is generated by one of the following methods:

Method 1: A Reliable Report

We say that the report received by agent t_i is reliable if it is generated by a process which guarantees that $A_{f(i)} \subset A_i$. Notice that if $A_{f(i)}$ contains the true proposition, then A_i will also contain the true proposition.

Method 2: An Irrelevant Report

The report received by agent t_i is said to be irrelevant if A_i has nothing to do with $A_{f(i)}$. This may be caused by some kind of misunderstanding or communications error. When the report is irrelevant, no information will be transmitted.

Note that a reliable report does not require $A_{f(i)} = A_i$. This is because we believe that information tends to disperse even when it is not distorted.

Assume that we do not know exactly which method is used to transmit the report to agent t_i, but we know the relative frequencies of the two methods. To represent this piece of evidence we can construct the belief function BEL_i over the marginal frame of discernment, which is either $\Omega_0 \times 2^{\Omega_i}$ or $2^{\Omega_{f(i)}} \times 2^{\Omega_i}$ depending on whether t_i is an element of S(0) or not. For $t_i \in S(0)$, the belief function BEL_i will be

$$\begin{aligned} m(\{(\theta_0, A_i) \mid \theta_0 \in A_i\}) &= r_i \\ m(\Omega_0 \times 2^{\Omega_i}) &= 1 - r_i \end{aligned} \tag{2.1}$$

where r_i is the probability that the report is reliable, and $1 - r_i$ is the probability that the report is irrelevant. For $t_i \notin S(0)$, BEL_i will be

$$\begin{aligned} m(\{(A_{f(i)}, A_i) \mid A_{f(i)} \subset A_i\}) &= r_i \\ m(2^{\Omega_{f(i)}} \times 2^{\Omega_i}) &= 1 - r_i. \end{aligned} \tag{2.2}$$

Note the simplicity of the belief function, at least relative to an ordinary probability assignment which would need to assign a numerical probability to each A_i for each given $A_{f(i)}$.

The minimal extensions of (2.1) and (2.2) to Ω are

$$\begin{aligned} m(\{(\theta_0, A_i) \mid \theta_0 \subset A_i\} \times \underset{(j>0, j\neq i)}{X} 2^{\Omega_j}) &= r_i \\ m(\Omega) &= 1 - r_i \end{aligned} \tag{2.3}$$

and

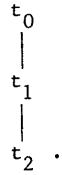
$$\begin{aligned} m(\Omega_0 \times \{(A_{f(i)}, A_i) \mid A_{f(i)} \subset A_i\} \times \underset{(j>0, j\neq i, f(i))}{X} 2^{\Omega_j}) &= r_i \\ m(\Omega) &= 1 - r_i \end{aligned} \tag{2.4}$$

respectively. Let the minimal extension of BEL_i be referred to as BEL'_i. By assuming that the messages are transmitted independently, the belief function $BEL_T = \bigoplus_{(i>0)} BEL'_i$ will represent all the evidence we have about the transmission process.

When the terminal messages are actually observed to be "the true proposition is in A_i^*", $t_i \in \tilde{T}$, our beliefs about the true proposition can be represented by the marginal belief function over Ω_0 corresponding to $BEL_T(\cdot \mid A_i = A_i^*, t_i \in \tilde{T})$. Consider the following example:

Example 2.1: Combined Reliability of Successive Agents

Assume that the transmission process can be represented by the following tree:



From (2.3) and (2.4), BEL'_1 and BEL'_2 are

$$\begin{aligned} m(\{(\theta_0, A_1) \mid \theta_0 \in A_1\} \times 2^{\Omega_2}) &= r_1 \\ m(\Omega) &= 1 - r_1 \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} m(\Omega_0 \times \{(A_1, A_2) \mid A_1 \subset A_2\}) &= r_2 \\ m(\Omega) &= 1 - r_2 \end{aligned} \tag{2.6}$$

respectively. Combining (2.5) and (2.6) using the rule of combination and conditioning on $A_2 = A_2^*$ we get

$$
\begin{aligned}
m(\{(\theta_0, A_1) \mid \theta_0 \in A_1, A_1 \subset A_2^*\} \times \{A_2^*\}) &= r_1 r_2 \\
m(\{(\theta_0, A_1) \mid \theta_0 \in A_1\} \times \{A_2^*\}) &= r_1(1 - r_2) \\
m(\Omega_0 \times \{A_1 \mid A_1 \subset A_2^*\} \times \{A_2^*\}) &= r_2(1 - r_1) \\
m(\Omega_0 \times 2^{\Omega_1} \times \{A_2^*\}) &= (1 - r_1)(1 - r_2).
\end{aligned} \tag{2.7}
$$

To find the marginal belief function over Ω_0, we have to study the focal elements listed in (2.7) and find out what they tell us about the relationships between θ_0 and A_2^*. The first focal element requires that $\theta_0 \in A_1$ and $A_1 \subset A_2^*$. This implies that $\theta_0 \in A_2^*$. Focal elements 2 to 4 require no relationship between θ_0 and A_2^*. Thus, the marginal belief function over Ω_0 corresponding to (2.7) is

$$
\begin{aligned}
m(\{\theta_0 \mid \theta_0 \in A_2^*\}) &= r_1 r_2 \\
m(\Omega_0) &= 1 - r_1 r_2.
\end{aligned} \tag{2.8}
$$

Belief function (2.8) implies that the second report is relevant only if both reports are reliable. This result is not surprising and applies to more complicated tree structures. In general, a report, whether terminal or not, received by agent t_i carries relevant information only if the reports received by agents t_j, $t_j \in C(i)$, are all reliable. For a tree where every node except the terminal node has only one son, the belief function constructed based on the terminal message received by agent t_N has the following basic probability assignments:

$$
\begin{aligned}
m(\{\theta_0 \mid \theta_0 \in A_N^*\}) &= \prod_{j=1}^{N} r_j \\
m(\Omega_0) &= 1 - \prod_{j=1}^{N} r_j.
\end{aligned} \tag{2.9}
$$

This result can be generalized even further. For any tree structure, the marginal belief function over Ω_0 corresponding to $BEL_T(\cdot \mid A_i = A_i^*)$, where t_i is any terminal node, has basic probability assignments

$$
\begin{aligned}
m(\{\theta_0 \mid \theta_0 \in A_i^*\}) &= r_i \prod_{t_j \in C(i)} r_j \\
m(\Omega_0) &= 1 - r_i \prod_{t_j \in C(i)} r_j
\end{aligned} \tag{2.10}
$$

where r_0 is assumed to be 1. This belief function represents our conditional beliefs over Ω_0 based on one single terminal message. In general, the marginal belief function over Ω_0 corresponding to $BEL_T(\cdot \mid A_i = A_i^*, t_i \in \tilde{T})$ is not equal to the combination of the marginal belief functions over Ω_0 corresponding to $BEL_T(\cdot \mid A_i = A_i^*)$, $t_i \in \tilde{T}$. This implies that different terminal messages may not be independent pieces of evidence about Ω_0. Later, in example (2.2) we will again discuss necessary conditions for two different terminal messages to be independent pieces of evidence about Ω_0. ■

As demonstrated in example (2.1) the credibility of a report received by an agent depends not only on the reliability of that report itself, but also depends on the reliability of the reports received by the ancestors of the agent. For this reason we define the indicator functions X_i and Y_i as follows:

$$
X_i = \begin{cases} 1 \text{ if the report received by agent } t_i \text{ is reliable} \\ 0 \text{ if the report received by agent } t_i \text{ is irrelevant} \end{cases} \tag{2.11}
$$

and

$$
Y_i = \begin{cases} 1 \text{ if } X_i = 1 \text{ and } X_j = 1 \text{ for all } j \text{ such that } t_j \in C(i) \\ 0 \text{ otherwise} \end{cases} \tag{2.12}
$$

$i = 0,1,\ldots,N$. Each X_i takes values 1 and 0 with probabilities r_i and $1 - r_i$, respectively. In the case of X_0, we assume $r_0 = 1$.

The reports are assumed to be sent independently, meaning that the X_i are distributed independently. It follows that

$$
P(Y_i = y_i) = \begin{cases} r_i \prod_{t_j \in C(i)} r_j & \text{for } y_i = 1 \\ \left(1 - r_i \prod_{t_j \in C(i)} r_j\right) & \text{for } y_i = 0 \end{cases} \tag{2.13}
$$

The report received by agent t_i is said to be absolutely credible if $Y_i = 1$. If a report is absolutely credible, then it must be true.

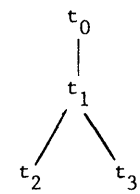
In example (2.1), there is another important implication to notice. For any tree, let t_i be a terminal node and let t_k be an ancestor of t_i. If $r_j = 1$ for $j = k$ and for all j such that $t_j \in C(k)$ then (2.10) becomes

$$
\begin{aligned}
m(\{\theta_0 \mid \theta_0 \in A_i^*\}) &= r_i \prod_{t_j \in C(i) \cap D(k)} r_j \\
m(\Omega_0) &= 1 - r_i \prod_{t_j \in C(i) \cap D(k)} r_j
\end{aligned} \tag{2.14}
$$

Notice that we get the same result if we assume that agent t_k knows exactly which is the true proposition. So as far as drawing information from a single subsequent message is concerned, making the assumption that the report received by agent t_k is absolutely credible has the same effect as assuming that agent t_k knows the absolute truth. Let us investigate whether this is still true when we are trying to draw information from two terminal messages which are both relayed through agent t_k.

Example 2.2:

Consider a transmission process represented by the following tree:



Assuming that $r_1 = 1$, the belief functions BEL'_1, BEL'_2 and BEL'_3 have basic probability assignments

$$
m(\{(\theta_0, A_1) \mid \theta_0 \in A_1\} \times 2^{\Omega_2} \times 2^{\Omega_3}) = 1 \tag{2.15}
$$

and

$$
\begin{aligned}
m(\Omega_0 \times \{(A_1, A_2) \mid A_1 \subset A_2\} \times 2^{\Omega_3}) &= r_2 \\
m(\Omega) &= 1 - r_2
\end{aligned} \tag{2.16}
$$

and

$$
\begin{aligned}
m(\Omega_0 \times 2^{\Omega_2} \times \{(A_1, A_3) \mid A_1 \subset A_3\}) &= r_3 \\
m(\Omega) &= 1 - r_3
\end{aligned} \tag{2.17}
$$

respectively. When we combine (2.15), (2.16) and (2.17) and condition on $A_2 = A_2^*$ and $A_3 = A_3^*$, the resulting belief function has basic probability assignments:

$$\begin{aligned} m(\{(\theta_0,A_1)|\theta_0\in A_1, A_1\subset A_2^*\cap A_3^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_2r_3 \\ m(\{(\theta_0,A_1)|\theta_0\in A_1, A_1\subset A_2^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_2(1-r_3) \\ m(\{(\theta_0,A_1)|\theta_0\in A_1, A_1\subset A_3^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_3(1-r_2) \\ m(\{(\theta_0,A_1)|\theta_0\in A_1\}\times\{A_2^*\}\times\{A_3^*\}) &= (1-r_2)(1-r_3). \end{aligned} \tag{2.18}$$

The first focal element listed in (2.18) does not exist if $A_2^* \cap A_3^* = \phi$. When this happens, the probability mass allocated to it must be reallocated to other focal elements, as in (1.8). Let us not worry about this at this moment and take a closer look at the focal elements to figure out the marginal belief function over Ω_0. It is very easy to see that the marginal belief function over Ω_0 corresponding to (2.18) is the same as the marginal belief function over Ω_0 corresponding to the belief function constructed over $\Omega_0 \times 2^{\Omega_2} \times 2^{\Omega_3}$ with the following basic probability assignements:

$$\begin{aligned} m(\{\theta_0|\theta_0\in A_2^*\cap A_3^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_2r_3 \\ m(\{\theta_0|\theta_0\in A_2^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_2(1-r_3) \\ m(\{\theta_0|\theta_0\in A_3^*\}\times\{A_2^*\}\times\{A_3^*\}) &= r_3(1-r_2) \\ m(\Omega_0\times\{A_2^*\}\times\{A_3^*\}) &= (1-r_2)(1-r_3). \end{aligned} \tag{2.19}$$

Notice that (2.19) is exactly the belief function we would construct if we assume that the two terminal messages come directly from agent t_0. It is also obvious that the marginal belief function over Ω_0 is equal to the combination of the two belief functions constructed over Ω_0 with the following basic probability assignments:

$$\begin{aligned} m(\{\theta_0|\theta_0\in A_2^*\}) &= r_2 \\ m(\Omega_0) &= 1-r_2 \end{aligned} \tag{2.20}$$

and

$$\begin{aligned} m(\{\theta_0|\theta_0\in A_3^*\}) &= r_3 \\ m(\Omega_0) &= 1-r_3 \end{aligned} \tag{2.21}$$

Belief function (2.20) is belief function (2.8) with $r_1 = 1$ and belief function (2.21) is just belief function (2.20) with r_2 replaced by r_3. Since belief function (2.8) represents our beliefs over Ω_0 based on one terminal message the equivalence between the marginal belief function over Ω_0 corresponding to (2.18) and (2.19) and the combination of (2.20) and (2.21) implies the following:

(1) If two terminal agents have the same father and if we know that the message received by the father is absolutely credible, then the messages received by the two terminal agents can be considered as independent pieces of evidence about Ω_0. (2.22)

In fact (2.22) can be generalized as

(1') If the report received by the nearest common ancestor of two terminal agents is absolutely credible, then the reports received by the two terminal agents can be considered as independent pieces of evidence about Ω_0. (2.23)

The logic behind statement (1') is very simple. If t_k is the common ancestor of t_i and t_j, $t_i, t_j \in \tilde{T}$, then the agents responsible for relaying the message from agent t_k to agent t_i are completely different from the agents responsible for relaying the message from t_k to t_j. This implies that, under the assumption that the report received by agent t_k is absolutely credible the credibility of the messages received by t_i and t_j are independent of each other. ■

In the two examples above we have demonstrated that we can find the marginal conditional belief function over Ω_0 by operating on BEL_T directly. On the other hand, for a more complicated tree structure where N is reasonably large, BEL_T will be huge and it will take a lot of computational effort for us to find $BEL_T(\cdot|A_i = A_i^*, t_i \in \tilde{T})$. Fortunately, because the transmission process has a tree structure, there is an alternative approach for calculating the conditional beliefs over Ω_0 which does not involve the actual computation of $BEL_T(\cdot|A_i = A_i^*, t_i \in \tilde{T})$. The alternative approach involves the concept of discounting which will be discussed below.

Let BEL be a belief function constructed over Ω_0. For $0 < \alpha < 1$, BEL^α is said to be a discounted version of BEL if

$$\begin{cases} m^\alpha(A) = (1-\alpha)\, m(A) & \text{for } A \neq \Omega_0 \\ m^\alpha(\Omega_0) = (1-\alpha)\, m(\Omega_0) + \alpha\ . \end{cases} \tag{2.24}$$

The number α is called the discounting factor. The idea of discounting was originally introduced by Shafer (1976). A belief function is discounted when we have doubts about the credibility of the item of evidence from which the belief function BEL is constructed. The value $(1-\alpha)$ can be interpreted as the probability that BEL is actually relevant to our inference. For $A \subset \Omega_0$, define BEL_A as the belief function over Ω_0 with a single focal element A. The belief function (2.8) can be written as $BEL_{A_2^*}^{1-r_1r_2}$. The interpretation is as follows. The belief function $BEL_{A_2^*}$ summarizes our evidence if the message received by agent 2 is absolutely credible. Since the probability of the message being so is only r_1r_2, we have to discount the belief function by the factor $(1-r_1r_2)$.

An alternative method for constructing the conditional belief function over Ω_0

Here we consider the terminal messages as different pieces of evidence about Ω_0. Instead of constructing belief functions over the produce space Ω, conditional belief functions over Ω_0 would be constructed. For notations, let $E_i, t_i \in \tilde{T}$, represents the evidence that agent t_i has received the message "the true proposition is in A_i^*". The belief function constructed over Ω_0 based on all available evidence is written as $BEL(\cdot|\{E_i|t_i \in \tilde{T}\})$. In theory, $BEL(\cdot|\{E_i|t_i \in \tilde{T}\})$ should be equal to the marginal belief function over Ω_0 corresponding to $BEL_T(\cdot|A_i = A_i^*, t_i \in \tilde{T})$. The alternative method for constructing $BEL(\cdot|\{E_i|t_i \in \tilde{T}\})$ can be summarized by the following theorem and its corollary.

Theorem 2.1.

$$\begin{aligned}&\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}\})\\ &=\bigoplus_{t_j\in S(0),t_j\notin\tilde{T}}\mathrm{BEL}^{1-r_j}(\cdot\,|Y_j=1,\{E_i|t_i\in\tilde{T}(j)\})\\ &\quad\oplus\left(\bigoplus_{t_j\in S(0),t_j\in\tilde{T}}\mathrm{BEL}_{A_j^*}^{1-r_j}\right),\end{aligned}\tag{2.25}$$

where E_i represents the evidence that agent t_i, $t_i\in\tilde{T}$, has received the message "the true proposition is in A_i^*".

Corollary:

$$\begin{aligned}&\mathrm{BEL}(\cdot\,|Y_j=1,\{E_i|t_i\in\tilde{T}(j)\})\\ &=\bigoplus_{t_k\in S(j),t_k\notin\tilde{T}}\mathrm{BEL}^{1-r_k}(\cdot\,|Y_k=1,\{E_i|t_i\in\tilde{T}(k)\})\\ &\quad\oplus\left(\bigoplus_{t_k\in S(j),t_k\in\tilde{T}}\mathrm{BEL}_{A_k^*}^{1-r_k}\right).\end{aligned}\tag{2.26}$$

Instead of proving Theorem 2.1 and its corollary rigorously, which requires much heavy mathematical notation and preliminary theorems about belief functions over product spaces with a tree structure, we choose a more descriptive approach which involves the extension of some of the results shown in Example (2.1) and Example (2.2).

Proof of Theorem 2.1.

The set $\{E_i|t_i\in\tilde{T}\}$ represents the total amount of evidence we have about Ω_0. The sets $\{E_i|t_i\in\tilde{T}(j)\}$, $j\in S(0)$, form a partition of $\{E_i|t_i\in\tilde{T}\}$. Two different sets, $\{E_i|t_i\in\tilde{T}(j_1)\}$ and $\{E_i|t_i\in\tilde{T}(j_2)\}$, $j_1,j_2\in S(0)$, $j_1\neq j_2$, represent independent collections of evidence about Ω_0. This is because the nearest common ancestor of t_{i_1} and t_{i_2}, $t_{i_1}\in\tilde{T}(j_1)$ and $t_{i_2}\in\tilde{T}(j_2)$, is t_0. It follows that

$$\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}\})=\bigoplus_{t_j\in S(0)}\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}(j)\}).\tag{2.27}$$

For $t_j\in S(0)$, let us consider two different cases.

Case 1: $t_j\in\tilde{T}$

When $t_j\in\tilde{T}$, $\{E_i|t_i\in\tilde{T}(j)\}$ is just $\{E_j\}$. From (2.10), we know that $\mathrm{BEL}(\cdot\,|\{E_j\})$ is equal to

$$\mathrm{BEL}_{A_j^*}^{1-r_j}\tag{2.28}$$

Case 2: $t_j\notin\tilde{T}$

When $t_j\notin\tilde{T}$, the terminal messages received by agents t_i, $t_i\in\tilde{T}(j)$, are all relayed through agent t_j. Since the messages provide relevant information only if $Y_j=1$, $P(Y_j=1)=P(X_j=1)=r_j$ implies that

$$\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}(j)\})=\mathrm{BEL}^{1-r_j}(\cdot\,|Y_j=1,\{E_i|t_i\in\tilde{T}(j)\}).\tag{2.29}$$

From (2.28) and (2.29), (2.27) can be rewritten as (2.25) and the theorem is proved.

Proof of the Corollary to Theorem 2.1.

Given that $Y_j=1$, we know for sure that the message received by agent t_j is absolutely credible. In Example (2.2) it has been shown that as far as drawing information from subsequent messages is concerned, we can actually assume that agent t_j knows exactly which proposition in Ω_0 is true. Realizing this we can think of $\{t_j\}\cup D(j)$ as a subtree, with t_j as the root and $\tilde{T}(j)$ as the set of terminal nodes, so that (2.26) is just an analogue to (2.25).

In general, by applying Theorem 2.1 and then its corollary recursively, $\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}\})$ can be expressed in terms of $\mathrm{BEL}_{A_i^*}$, $t_i\in\tilde{T}$.

Although the theorem and its corollary tells us how to look at the tree from the top to the bottom, the actual calculation of $\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}\})$ is from the bottom to the top. The procedure can be summarized as follows:

Step 1: Start at the terminal nodes and construct the belief functions $\mathrm{BEL}_{A_i^*}$, $t_i\in\tilde{T}$.

Step 2: Go up one level. When going from node t_j to $t_{f(j)}$, we discount the corresponding belief function by $(1-r_j)$ and combine with other belief functions which merge into the same node. In general, every edge of the tree represents a discounting operation and every node having more than one son represents a combination operation.

Step 3: Apply step 2 recursively until reaching the top (root) of the tree. The final belief function will be $\mathrm{BEL}(\cdot\,|\{E_i|t_i\in\tilde{T}\})$.

In general the number of combinations that has to be performed is equal to $|\tilde{T}|-1$.

Extensions. Dempster and Kong (1984) study a model where each communication channel may be used for a deliberate lie ($A_{f(i)}\subset A_i^c$) as well as for a true statement or a meaningless statement. Theory for models with more general graphical structures appears in Kong (1986).

Acknowledgement. This work was facilitated in part by ARO Contract DAAG 29-84-K-0120 and ONR Contract No. N00014-85-K-0490.

References

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping, *Ann. Math. Statist.* 38, 325-339.

Dempster, A. P. and Kong, A. (1984). Belief functions and communications networks, Research Report S-98, Department of Statistics, Harvard University.

Kong, A. (1986). *Graphical Belief Function Models*, Ph.D. thesis, Department of Statistics, Harvard University (in preparation).

Shafer, G. (1976). *A Mathematical Theory of Evidence*, Princeton University Press.

Shafer, G. (1981). Constructive probability, *Synthese* 48, 1-60.

Shafer, G. (1982a). Lindley's paradox, *J. Amer. Statist. Assoc.* 77, 325-351.

Shafer, G. (1982b). Belief functions and parametric models, *J. Roy. Statist. Soc. B* 44, 322-339.

MAXIMUM LIKELIHOOD DETECTION AND ESTIMATION OF JUMPS IN DYNAMIC SYSTEMS

by

R.C. Morgan, M.S. Asher, D.W. Porter, and W.S. Levine
Business and Technological Systems, Inc.
10210 Greenbelt Road, Suite 440
Seabrook, Maryland 20706

ABSTRACT

We propose that the maximum likelihood method be used to detect and estimate jumps in dynamical systems. The primary advantage of ML over hypothesis testing methods, such as GLR, is that estimate errors are obtained for the jump times as well as for the jump amplitudes.

1.0 INTRODUCTION

There are several practical applications where it is necessary to estimate impulsive jumps in the state of a dynamic system. One application is the estimation of instrument shifts in inertial navigation systems. The biases and drift rates associated with accelerometers and gyroscopic attitude references can suddenly change value because of unmodeled physical effects. Another application is the tracking of maneuvering targets, where target accelerations or even control surface deflections can be modeled as jump processes.

One of the most widely used algorithms for accommodating such jumps is the generalized likelihood ratio (GLR) technique and its variations [4]. Some of these algorithms process the data in one large batch while others search for jumps in a moving window. These algorithms attempt to do two things: 1) detect jumps and 2) estimate parameters associated with the jumps, i.e., time and amplitude. In accomplishing the second goal, these algorithms approximate the problem of estimating the jump times of a continuous dynamic system by an hypothesis testing problem in which the jumps are allowed to assume only a finite number of values.

The present paper proposes that the jump times be treated as what they really are: continuous parameters. The jump times and amplitudes can then be estimated by a maximum likelihood method. Two advantages arise from this treatment. Firstly, gradient and Newton searches can be used to maximize the likelihood function. This can converge much faster than exhaustive tests of all hypothesized jump times. Secondly, a measure of the uncertainty in the estimates of the jump time and the correlation with jump amplitude estimate errors is produced via the inverse of the Fisher information matrix. In real-time tracking applications, the advantage of the first improvement is obvious in that it can greatly reduce the computational burden. Using the full maximum likelihood estimate error covariance matrix (including jump time estimate uncertainty) to increase the Kalman filter covariance can result in a real-time tracking filter that is more responsive to target maneuvers.

2.0 MODEL

Consider the discrete-time stochastic linear system

$$\underline{x}_k = \Phi_{k-1}\underline{x}_{k-1} + G_k\underline{w}_k + \sum_{j=1}^{NJ} \tilde{\Phi}_k(\tau_j)\, s_j \underline{\nu}_j \qquad (2.1)$$

$$y_k = H_k\underline{x}_k + r_k \qquad (2.2)$$

where

$\underline{x}$ is the n-dimensional state vector at time t_k;

$\underline{x}_0 \sim N(\underline{\mu}, \Sigma)$;

y_k is the scalar measurement at time t_k;

$\{\underline{w}_1, \ldots, \underline{w}_M\}$ and $\{r_1, \ldots, r_M\}$ are zero mean, Gaussian, white noise sequences that are mutually independent and independent of $\underline{x}_0$;

H_k is the n-dimensional row vector representing the measurement sensitivity;

$\Phi_{k-1} = \Phi(t_k; t_{k-1})$ is the transition matrix for the differential equation $\dot{\underline{x}} = A(t)\,\underline{x}$;

$$\tilde{\Phi}_k(\tau_j) = \begin{cases} \Phi(t_k; \tau_j) & , \quad \tau_j \in (t_{k-1}, t_k] \\ 0 & , \quad \text{otherwise} \end{cases} ; \text{ and}$$

$S_j\underline{\nu}_j$ is the jump in the state at time τ_j.

The unknown (deterministic) parameters of the problem are the jump times $\{\tau_1, \ldots, \tau_{NJ}\}$ and, what we will call the jump amplitudes $\{\underline{\nu}_1, \ldots, \underline{\nu}_{NJ}\}$. We assume that each "selector" matrix S_j is given. A more important assumption that we currently make is that the number of jumps, NJ, is known. In Section 6.0 we will discuss briefly the possibility of including NJ as an unknown parameter. Note that it is possible to consider vector measurements in (2.2). We consider only scalar measurements because this simplifies the notation.

3.0 MAXIMUM LIKELIHOOD

We will use the maximum likelihood (ML) method, along with the scoring optimization technique, to determine the optimal values of the jump parameters in (2.1). Our derivation of the negative log likelihood function will follow the derivation in [4]. The state $\underline{x}_k$ can be decomposed into two parts: one that is independent of the jump parameters and one that depends solely on the jump parameters. This decomposition applies also to the measurements, and consequently to the residuals that are derived when passing the measurements through a jump free Kalman filter. We will denote by γ_k^{NJ} that part of the residual, when processing the $k^{\underline{th}}$ measurement, that is independent of the jump parameters, and by $\gamma_k^{NJ}(\theta)$ that part of the residual that depends on the jump parameters $\underline{\theta}^T = (\tau_1, \ldots, \tau_{NJ}, \underline{\nu}_1^T, \ldots, \underline{\nu}_{NJ}^T)$.

We can now write the $k^{\underline{th}}$ residual γ_k in the form $\gamma_k = \gamma_k^{NJ} + \gamma_k^J(\underline{\theta})$. (Note that γ_k, and not γ_k^{NJ}, is the residual that is generated by the jump free filter.) It follows that γ_k^{NJ} is a zero-mean white noise sequence and that $\gamma_k^J(\underline{\theta})$ is deterministic. If we denote

$$\mathrm{cov}(\gamma_k^{NJ}) = B_k$$

then

$$\gamma_k \sim N(\gamma_k^J(\underline{\theta}), B_k)$$

Because $\{\gamma_1, \ldots, \gamma_M\}$ is a white Gaussian sequence, the negative log likelihood function can now be written as

$$J(\underline{\theta}) \equiv -\log p(\gamma_1, \ldots, \gamma_M \mid \underline{\theta}) \tag{3.1}$$

$$= \frac{M}{2}\log 2\pi + \frac{1}{2}\sum_{k=1}^{n}\log B_k + \frac{1}{2}\sum_{k=1}^{M}\frac{1}{B_k}(\gamma_k - \gamma_k^J(\underline{\theta}))^2$$

where

$$\gamma_k^J(\underline{\theta}) \equiv \gamma_k^J(\tau_1, \ldots, \tau_{NJ}, \underline{\nu}_1, \ldots, \underline{\nu}_{NJ}) \tag{3.2}$$

$$= \sum_{j=1}^{NJ} H_k \Psi_k(\tau_j)\, S_j \underline{\nu}_j$$

and

$$\Psi_k(\tau) = \begin{cases} 0 & , \ t_k < \tau \\ \Phi(t_k;\tau) & , \ t_{k-1} < \tau \le t_k \\ \Phi_{k-1}(I - K_{k-1}H_{k-1})\,\Psi_{k-1}(\tau) & , \ \tau \le t_{k-1} \end{cases} \tag{3.3}$$

where K_k is the Kalman gain that arises when processing the $k^{\underline{th}}$ measurement, and H_k is the measurement sensitivity given in (2.2). Combining (3.1) and (3.2) yields

$$J(\tau_1, \ldots, \tau_{NJ}, \underline{\nu}_1, \ldots, \underline{\nu}_{NJ}) \tag{3.4}$$

$$= \ldots + \frac{1}{2}\sum_{k=1}^{M}\frac{1}{B_k}(\gamma_k - \sum_{j=1}^{NJ} H_k \Psi_k(\tau_j)\, S_j \underline{\nu}_j)^2 \quad .$$

The scoring method is a variant of Newton's method in which the Hessian is replaced with its expectation. For our problem, the $\ell^{\underline{th}}$ iterate becomes

$$\underline{\theta}_\ell = \underline{\theta}_{\ell-1} + F^{-1}(\underline{\theta}_{\ell-1})\,\underline{\nabla J}(\underline{\theta}_{\ell-1})$$

where

$$(\underline{\nabla J})_i = \frac{\partial J}{\partial \theta_i}$$

and

$$F_{ij} = E\left[\frac{\partial^2 J}{\partial\theta_i\,\partial\theta_i}\right] \quad .$$

We will develop the formula for $(\underline{\nabla J})_i$ explicitly in terms of $\tau_1, \ldots, \tau_{NJ}$ and $\underline{\nu}_1, \ldots, \underline{\nu}_{NJ}$, since we will refer to it later, but will leave the formula for F_{ij} in terms of $\underline{\theta}$.

It follows from $\frac{\partial\Phi}{\partial\tau}(t_k;\tau) = -\Phi(t_k;\tau)\,A(\tau)$ and (3.3) that

$$\frac{d\Psi_k}{d\tau}(\tau) = -\Psi_k(\tau)\,A(\tau) \qquad (\tau \neq t_k) \quad . \tag{3.5}$$

Thus, for $i = 1, \ldots, NJ$,

$$\frac{\partial J}{\partial \tau_i} = \sum_{k=1}^{M}\frac{1}{B_k}(\gamma_k - \sum_{j=1}^{NJ} H_k \Psi_k(\tau_j)\, S_j \underline{\nu}_j)\, H_k \Psi_k(\tau_i)\, S_i \underline{\nu}_i \tag{3.6}$$

$$\frac{\partial J}{\partial \underline{\nu}_i} = -\sum_{k=1}^{M}\frac{1}{B_k}(\gamma_k - \sum_{j=1}^{NJ} H_k \Psi_k(\tau_j)\, S_j \underline{\nu}_j)(H_k \Psi_k(\tau_i)\, S_i)^T \quad . \tag{3.7}$$

The expectation of the Hessian of J is equal to the expected value of the covariance of $\underline{\nabla J}$, as shown in any standard text, [3]. Also, note that $\gamma_k \sim N(\gamma_k^J(\underline{\theta}), B_k)$, that $\{\gamma_1, \ldots, \gamma_M\}$ is an independent sequence, and that in equation (3.1) only γ_k depends on the measurements. Consequently,

$$F_{ij} = E\left[\frac{\partial^2 J}{\partial\theta_i\,\partial\theta_j}\right] = E\left[\frac{\partial J}{\partial\theta_i}\,\frac{\partial J}{\partial\theta_j}\right]$$

$$= \sum_{k,\ell} \frac{1}{B_k B_\ell} E[(\gamma_k - \gamma_k^J(\underline{\theta}))(\gamma_\ell - \gamma_\ell^J(\underline{\theta}))] \frac{\partial \gamma_k^J}{\partial \theta_i} \frac{\partial \gamma_\ell^J}{\partial \theta_j}$$

$$= \sum_{k=1}^{M} \frac{1}{B_k} \frac{\partial \gamma_k^J}{\partial \theta_i}(\underline{\theta}) \frac{\partial \gamma_k^J}{\partial \theta_j}(\underline{\theta})$$

4.0 CONTINUITY OF THE LIKELIHOOD AND ITS GRADIENT

Our goal in this section is to state conditions that will guarantee that the likelihood function and its gradient are continuous functions. This will be done by stating conditions under which the functions γ_k^J for $k = 1, \ldots$ and their derivatives are continuous.

Since γ_k^J is a linear function of the jump amplitudes, we will concentrate on the smoothness of γ_k^J and its derivatives, as functions of the jump times. Recall that $\phi(t; \tau)$ is the transition matrix for the differential equation $\dot{\underline{x}} = A(t)\,\underline{x}$. We assume that A is a bounded piecewise smooth matrix valued function. Thus, $\tau \to \frac{\partial \Phi}{\partial \tau}(t_{k+1}; \tau) = -\Phi(t_{k+1}; \tau)\, A(\tau)$ will have a jump discontinuity whenever A is discontinuous. A consequence of (3.2) and (3.3) is that γ_k^J and its derivatives with respect to the jump times are at worst, piecewise smooth functions of the jump times; the possible points of discontinuity are the measurement times and points of discontinuity of A and its derivaties.

We will assume that there is only one jump. The conditions that guarantee continuity in this case can then be applied to the parameters of each jump when there is more than one jump. The function γ_k^J for $k = 1, \ldots, M$, is defined by

$$\gamma_k^J(\tau, \nu) = H_k \Psi_k(\tau)\, S\underline{\nu} \tag{4.1}$$

where H_k is an n-dimensional row vector, Ψ_k is an nxn matrix defined by (3.3), $\underline{\nu}$ is an m-vector, and the selector matrix S is an nxm matrix.

Conditions that guarantee the continuity of γ_k^J and its derivatives are:

THEOREM: If $H_k S = \underline{0}^T$ for all $k = 1, \ldots, M$, then γ_k^J and $\frac{\partial \gamma_k^J}{\partial \underline{\nu}}$ for $k = 1, \ldots, M$, are continuous functions. If in addition, $H_k A(t_k)\, S = \underline{0}^T$ for all $k = 1, \ldots, M$, and A is continuous everywhere, then $\frac{\partial \gamma_k^J}{\partial \tau}$ for $k = 1, \ldots, M$, is continuous.

A proof is given in the appendix.

5.0 NUMERICAL EXAMPLE

The use of maximum likelihood, in which scoring is the optimization technique, is illustrated in this section. We consider a 4-state model whose dynamics are given by

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 0 & c_1 & 0 & c_3 \\ 0 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{Bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 0 \\ f(t) \end{Bmatrix} \tag{5.1}$$

where

$$f(t) = \sum_{j=1}^{NJ} \delta(t - \tau_j)\, \nu_j \tag{5.2}$$

and the initial state satisfies $x_{i0} \sim N(0, 1)$ for $i = 1, 2, 3, 4$. There are 200 measurements of the form

$$y_k = x_{1k} + r_k \tag{5.3}$$

where $r_k \sim N(0, .01^2)$. Note that the selector matrices in this example are 4x1 matrices equal to $(0, 0, 0, 1)^T$, each jump amplitude ν_j is a scalar, and the measurement sensitivity matrix $H_k = (1, 0, 0, 0)$. A consequence of the theorem in Section 4.0 is that the negative log likelihood function associated with (5.1) - (5.3), and its derivatives with respect to the jump amplitudes are continuous functions because $H_k S_j = 0$. Furthermore, the derivatives with respect to the jump times will be continuous between measurement times because $A(t)$ is a constant matrix, but may be discontinuous at each measurement time because $H_k A(t)\, S_j = c_3 \neq 0$.

The results presented here are for the case in which $NJ = 2$ and $\tau_1 = 75$, $\nu_1 = 6.67$, and $\tau_2 = 125$, $\nu_2 = -6.67$. Figure 5-1 is a graph of the negative log likelihood as a function of the two jump times, with the jump amplitudes fixed at their true values. It is evident that the initial estimate of τ_1 and τ_2 must not be too far from the true values, in order to converge to the true values, because the scoring method may easily converge to a local minimum rather than the global minimum. Table 5.1 gives the final estimate of the jump parameters based on several initial estimates. The column labelled NLL contains the value of the negative log likelihood (minus the $\frac{1}{2} M \log 2\pi$ term) at the corresponding values of the jump parameters. One can see that care must be exercised, as with any optimization scheme, when selecting the initial estimates. The initial estimate $\tau_1 = 65$, $\nu_1 = 6$ and $\tau_2 = 105$, $\nu_2 = -6$ does not appear to by any worse than the other initial estimates in Table 5.1, however the final estimate is far from the true values. In fact, $\tau_2 = 200$ is at the end of the data stream.

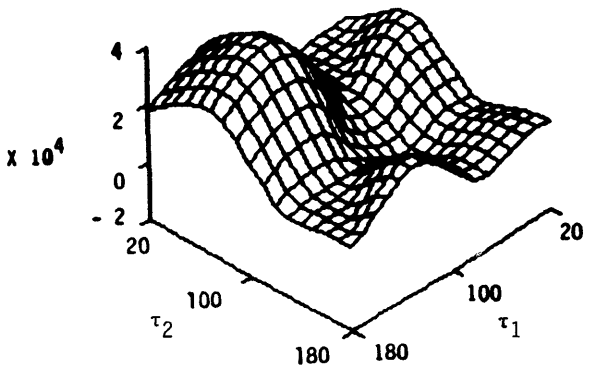


FIGURE 5-1

TABLE 5.1

INITIAL					FINAL				
τ_1	ν_1	τ_2	ν_2	NLL	τ_1	ν_1	τ_2	ν_2	NLL
65	6.0	105	-6.0	4539	70.4	10.0	200	-16.67	1360
85	7.0	130	-1.33	3663	74.9	6.60	125	- 6.80	-1270
95	6.33	100	- .677	8930	74.9	6.60	125	- 6.80	-1270
70	6.33	120	-7.00	- 287	74.9	6.60	125	- 6.80	-1270

6.0 FUTURE RESEARCH

Several areas in which we feel further work is needed include determining the effects of the lack of smoothness of the likelihood function, deriving a more efficient optimization scheme, determining the number of jumps, and selecting initial estimates of the jump parameters. Perhaps the most important from a practical point of view are the determination of the number of jumps and the selection of initial estimates. Clearly, these are closely related problems. The Akaike information criterion [1] can be used to help determine the number of jumps. It requires that the number of jump parameters in the problem be added to the negative log likelihood, and then, that this new function be minimized.

However, the Akaike information criterion does not provide a method of selecting, a priori, the number of jumps or types of jumps. One way in which we envision this criterion being used is as follows. First, select more jumps than seems reasonable for the problem, and find the optimum values of the jump parameters. Then, based on some measure of the optimum values, remove some of the jump parameters, and optimize again. One measure that has given reasonably good results with higher order problems is the ratio of the estimated jump amplitude to its one-sigma uncertainty as derived from the inverse of the Fisher information matrix. This procedure will continue until the negative log likelihood, augmented by the Akaike information criterion, no longer diminishes in value. Clearly, determining a viable method to remove jump parameters is a non-trivial task.

Concerning the optimization method, one might try an update method or a trust region method [2] instead of scoring. On the other hand, it might also be beneficial to rearrange formulas (3.6) and (3.7), the zeros of which, most optimization methods will attempt to find. Noting that (3.7) is linear in the jump amplitudes $\underline{\nu}_j$, it is possible to solve the linear system $\frac{\partial J}{\partial \underline{\nu}_i} = 0$, $i = 1, \ldots, NJ$ for $\underline{\nu}_i$ in terms of the jump times $(\tau_1, \ldots, \tau_{NJ})$. This fact was used to eliminate the jump amplitudes in the hypothesis testing problem described in [4]. Formally, each $\underline{\nu}_i$ can then be substituted into (3.6), and the optimization procedure can then seek the zeros of $\frac{\partial J}{\partial \tau_i}$, $i = 1, \ldots, NJ$ as functions of the jump times only. The primary advantage is that the sizes of the matrices in the optimization procedure can be substantially smaller than when all of the jump parameters are being considered at once. However, it may be necessary to run the Kalman filter twice per iteration; once to solve for the jump amplitudes in terms of the jump times and once when determining the jump times. After the optimum jump parameters have been obtained, the Kalman filter can be run one more time in order to compute the Fisher information matrix, the inverse of which gives, asymptotically, the error covariance.

Finally, the lack of smoothness should not affect the solvability, but will affect the rate of convergence. Furthermore, it is important to know how the lack of smoothness affects the nice properties, such as efficiency and consistency, that are associated with maximum likelihood estimators.

APPENDIX A

We want to prove the theorem in Section 4.0. For a given function g, we will state its continuity in terms of $g(\tau^+) - g(\tau^-) \equiv \lim_{t \to \tau} g(t) - \lim_{t \to \tau} g(t)$. A consequence of (3.3) is

$$\Psi_k(t_{k-1}^+) = \Phi_{k-1} \quad \text{and} \quad \Psi_k(t_k^-) = I \quad . \qquad (A.1)$$

Also, define

$$G_{k\ell} = \begin{cases} I & , \ \ell = k \\ \Phi_{k-1}(I-K_{k-1}H_{k-1})\ldots\Phi_\ell(I-K_\ell H_\ell), & \ell = 1,\ldots,k-1 \end{cases} .$$

Next, we will show that

$$\Psi_k(t_\ell^+) - \Psi_k(t_\ell^-) = \begin{cases} G_{k,\ell+1}\,\Phi_\ell\,K_\ell\,H_\ell & , \ \ell \le k-1 \\ I & , \ \ell = k \\ 0 & , \ \ell \ge k+1 \end{cases} . \qquad (A.2)$$

For $\ell \le k-1$, it follows from (3.3) and (A.1) that

$$\begin{aligned} \Psi_k(t_\ell^+)-\Psi_k(t_\ell^-) &= G_{k,\ell+1}(\Psi_{\ell+1}(t_\ell^+)-\Psi_{\ell+1}(t_\ell^-)) \\ &= G_{k,\ell+1}(\Psi_{\ell+1}(t_\ell^+)-\Phi_\ell(I - K_\ell H_\ell)\Psi_{\ell+1}(t_\ell^-)) \\ &= G_{k,\ell+1}\,\Phi_\ell K_\ell H_\ell \quad . \end{aligned}$$

The cases $\ell = k$ and $\ell \geq k+1$ follow in a similar manner.

We now prove the theorem that is stated in Section 4.0. It follows from (4.1) and (3.3) that $\gamma_k^J(\tau, \underline{\nu})$ is continuous for all τ not equal to a measurement time and for all $\underline{\nu}$. If $\tau = t_\ell$ then by (A.2),

$$\gamma_k^J(t_\ell^+, \underline{\nu}) - \gamma_k^J(t_\ell^-, \underline{\nu}) = \begin{cases} H_k G_{k,\ell+1} \Phi_\ell K_\ell H_\ell S\underline{\nu} & , \ \ell \leq k-1 \\ H_k S\underline{\nu} & , \ \ell = k \\ 0 & , \ \ell \geq k+1 \end{cases}$$

is true, and by our assumption this is 0 for all values of ℓ. Thus, γ_k^J is a continuous function.

The continuity of $\frac{\partial \gamma_k^J}{\partial \underline{\nu}}$ is proven the same way since

$$\frac{\partial \gamma_k^J}{\partial \underline{\nu}}(\tau, \underline{\nu}) = H_k \Psi_k(\tau)\, S$$

and

$$\frac{\partial \gamma_k^J}{\partial \underline{\nu}}(t_\ell^+, \underline{\nu}) - \frac{\partial \gamma_k^J}{\partial \underline{\nu}}(t_\ell^-, \underline{\nu}) = \begin{cases} H_k G_{k,\ell+1} \Phi_\ell K_\ell H_\ell S\nu & , \ \ell \leq k-1 \\ H_k S\nu & , \ \ell = k \\ 0 & , \ \ell \geq k+1 \end{cases}$$

Finally, it follows from (A.2), that

$$\frac{\partial \gamma_k^J}{\partial \tau}(\tau, \nu) = -H_k \Psi_k(\tau)\, A(\tau)\, S\underline{\nu} \quad ,$$

and then from (3.5) and (3.3) that

$$\frac{\partial \gamma_k^J}{\partial \tau}(t_\ell^+, \nu) - \frac{\partial \gamma_k^J}{\partial \tau}(t_\ell^-, \nu) = \begin{cases} H_k G_{k,\ell+1} \Phi_\ell K_\ell H_\ell A(t_\ell) S\underline{\nu} & , \ \ell \leq k-1 \\ H_k S A(\tau_\ell) S\underline{\nu} & , \ \ell = k \\ 0 & , \ \ell \geq k+1 \end{cases}$$

because A is continuous. Since $H_\ell A(t_\ell) S = \underline{0}^T$, by assumption, it follows that $\frac{\partial \gamma_k^J}{\partial \tau}$ is continuous.

REFERENCES

[1] Akaike, H., "A new look at the statistical model identification," IEEE Trans. on AC, vol. AC-19, 1974, pp. 716-722.

[2] Dennis, J.E. and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentics-Hall, 1983.

[3] Sorenson, H.W., Parameter Estimation, Marcel Dekker, Inc., 1982.

[4] Willsky, A.S. and H.L. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," IEEE Trans. on AC, vol. AC-21, no. 1, February 1976, pp. 108-112.

A MARKOV TWO SENSOR CORRELATION PERFORMANCE MODEL

M. A. KOVACICH

COMPTEK RESEARCH, INC., 100 CORPORATE PLACE, SUITE B
VALLEJO, CALIFORNIA 94590

ABSTRACT. A Markov model describing the performance of a simple two sensor track-to-track correlation/decorrelation algorithm is discussed. Tracks are generated by two radar sensors covering the same surveillance area. Track arrivals are modeled as a Poisson process with track lives determined by an exponential distribution. Nearest neighbor correlation processing is performed on new tracks and positional separation decorrelation testing is performed on paired tracks. The performance of the correlation/decorrelation (C/D) is modeled as a continuous time, finiate state Markov process, and the Chapman-Kolmogorov equation is developed. Solution of the Chapman-Kolmogorov equation for the transient or limiting state probabilities proved to be exceedingly difficult even for this fairly simple C/D process. Further theoretical work will be required to develop the tools needed to obtain results from this analytical approach to modeling the dynamical behavior of the full correlation/decorrelation process.

INTRODUCTION

Track-to-track correlation and decorrelation is a complex process whose performance has generally been determined through simulation, and through analytical modelling of pieces of the process. Type I and Type II correlation error rates i.e., miscorrelation and dual designation rates, have been determined analytically for certain decision rules in well defined situations, but little work has been done to characterize the performance of the combined correlation and decorrelation process as they operate through time. This paper is an attempt to determine the dynamical performance of the full track-to-track correlation and decorrelation (C/D) process through analytical means for a simplified C/D process.

The case considered here concerns two radar systems that are generating tracks in a common surveillance area. Track-to-track correlation is performed to pair tracks from the two radar systems in order to generate a nonredundant merged track file and decorrelation testing is performed to monitor the continued validity of previously made pairings to determine if they should be kept or broken. A simple correlation process is considered, namely a 2-D positional gating algorithm that employs a single pass, nearest neighbors algorithm for ambiguous situations. The decorrelation process operates by decorrelating tracks whose positional separation exceeds a specified threshold. This paper presents the dynamical behavior of this correlation/decorrelation process by means of a finite state, continuous time Markov model.

The Markov state of the C/D algorithm consists of the C/D decision/outcome counts, namely, the number of correct correlations (CORRECT CORRS), correct no correlations (CORRECT NO-CORRS), incorrect correlations (INCORRECT CORRS) and incorrect no correlations (INCORRECT NO-CORRS). Transitions between states occur when an event arrives at the C/D process (a NEW track, an UPDATE or a DROP track message), a decision is made (CORRELATE, NO-CORRELATE, DECORRELATE, KEEP) and an outcome determined (CORRECT and INCORRECT). The dynamical behavior of the C/D system is obtained by determining the transient and limiting state probability function using the Chapman-Kolmogorov forward equations.

SETTING

The setting for this problem consists of a surveillance area covered by two radar systems, radar A and radar B, that generate tracks that are sent to the C/D function for correlation and decorrelation testing. The characteristics of these three components: surveillance area, sensors, and C/D system are presented in Table 1.

TABLE 1 PROBLEM SETTING

1. ENVIRONMENT
   - Surveillance area covered by both sensors
   - Area = πR^2
   - Track density is uniform over the surveillance area
2. TRACKERS
   - No false tracks
   - No maneuver biases or intersensor biases
   - New tracks arrive according to a Poisson process with arrival rate ν_A (ν_B) tracks/sec for TRACKER A (TRACKER B)
   - Track updates occur according to a Poisson process with update rate r_A (r_B) for TRACKER A (TRACKER B)
   - Track holding time is modeled by the exponential distribution with average track life $1/\mu_A$ ($1/\mu_B$) for tracker A (TRACKER B)
   - Infinite track file capacity.

3. C/D ALGORITHM

- Correlation gates are 2-D positional gates of area $\pi\alpha$
- Ambiguities are resolved using a single pass, nearest neighbors logic
- Decorrelation is done when the paired track falls outside a circular gate with area $\pi\beta$
- No correlation or decorrelation action is done on updates on unpaired tracks.
- The probability that the correct pairing already exists for a new track from radar A (B) is given by the apriori value γ_A (γ_B).

MATHEMATICAL MODEL OF C/D PROCESS

It is assumed that the performance of the C/D process can be modeled as a Markov process with state space given by C/D decision/outcome counts of correct and incorrect CORRS and NO-CORRS, and with transitions occurring when a decision is made and outcome determined. The key element in this assumption is that the count of correct and incorrect CORRS and NO-CORRS fully describes the current state of the C/D process, that is, no previous historical information about how the tracks were correlated or decorrelated is required to determine the probability that the C/D State will transition to another state when a decision is made. This point requires further discussion. The C/D process can arrive at or leave from a given system state along many different sample paths. (The sample path considered here consists of the tracks, their positions and velocities and the current C/D decision/outcome counts.) In a specific sample path realization, it is clear that the transition probabilities from a given state will depend on more than the C/D decision/outcome counts. For example, the decorrelation rate for miscorrelated tracks will depend upon the relative vehicle separation. The Markov nature is retrieved by integrating over relative vehicle separations thereby eliminating it as a state variable. In general, the Markov nature of the C/D process is obtained using the C/D decision/outcome counts by integrating or randomizing other system dependencies.

The system state is given by

$$N = (N_1, N_{2A}, N_{2B}, N_3, N_{4A}, N_{4B})$$

where:

N_1 = # CORRECT CORRS

N_{2A} = # CORRECT NO-CORRS FROM A

N_{2B} = # CORRECT NO-CORRS FROM B

N_3 = # INCORRECT CORRS

N_{4A} = # INCORRECT NO-CORRS FROM A

N_{4B} = # INCORRECT NO-CORRS FROM B

with the condition that

$$N_{2B} = N_{4B}$$

when $N_3 = 0$. This condition states that when no miscorrelations exist, each incorrect NO-CORR from source A is paired with an incorrect NO-CORR from source B.

The total number of tracks in the track file supported by radar A is

$$N_A = N_1 + N_{2A} + N_3 + N_{4A}$$

and from radar B,

$$N_B = N_1 + N_{2B} + N_3 + N_{4B}.$$

The total number of tracks in the track file is given by

$$N_{TOT} = N_1 + N_{2A} + N_{2B} + N_3 + N_{4A} + N_{4B}.$$

The limiting or steady state average number of tracks from radar A is given by

$$\bar{N}_A = (\nu_A/\mu_A) \quad (1)$$

and of radar B,

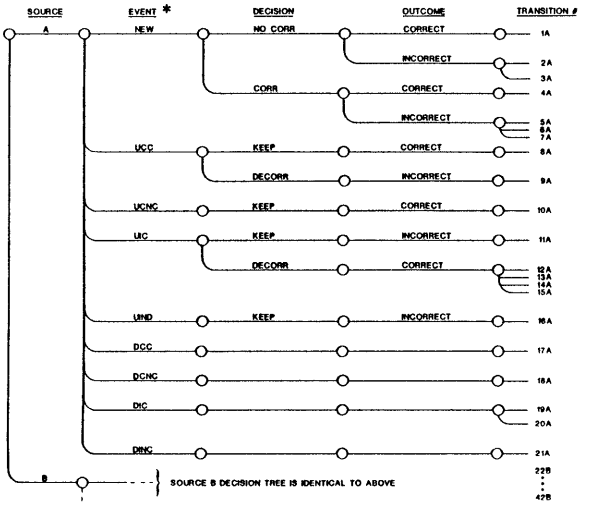
$$\bar{N}_B = (\nu_B/\mu_B) \quad (2)$$

Transitions from one state to another are made when source A or source B generates an event that results in a decision whose outcome is determined. The decision tree that describes the possible transitions is shown in Figure 1. The transition probabilities for each transition in the decision tree are shown in Table 2. Each transition probability is composed of two terms: the first is the event rate which are detailed in Table 3, and the second is the Decision/Outcome probabilities detailed in Table 4.

(Table 4 appears at the end of the report.)

FIGURE 1.

CORRELATION/DECORRELATION DECISION TREE



* The event terminology is discussed in Table 2.

TABLE 2 CORRELATION MODEL TRANSITION TABLE

START STATE: (N_1) (N_{2A}) (N_{2B}) (N_3) (N_{4A}) (N_{4B})

TRANSITION #	TRANSITION PROBABILITY	END* STATE	TRANSITION #	TRANSITION PROBABILITY	END* STATE
1	PNEWA P1A	$(N_{2A}+1)$	22	PNEWB P1B	$(N_{2B}+1)$
2	PNEWA P2A	$(N_{2B}-1)(N_{4B}+1)(N_{4A}+1)$	23	PNEWB P2B	$(N_{2A}-1)(N_{4A}+1)(N_{4B}+1)$
3	PNEWA P3A	$(N_{4A}+1)$	24	PNEWB P3B	$(N_{4B}+1)$
4	PNEWA P4A	$(N_1+1)(N_{2B}-1)$	25	PNEWB P4B	$(N_1+1)(N_{2A}-1)$
5	PNEWA P5A	$(N_3+1)(N_{2B}-1)$	26	PNEWB P5B	$(N_3+1)(N_{2A}-1)$
6	PNEWA P6A	$(N_3+1)(N_{4B}-1)$	27	PNEWB P6B	$(N_3+1)(N_{4A}-1)$
7	PNEWA P7A	$(N_3+1)(N_{2B}-2)(N_{4B}+1)$	28	PNEWB P7B	$(N_3+1)(N_{2A}-2)(N_{4A}+1)$
8	PUCCA P8A	----	29	PUCCB P8B	----
9	PUCCA P9A	$(N_{4B}+1)(N_1-1)(N_{4A}+1)$	30	PUCCB P9B	$(N_{4A}+1)(N_1-1)(N_{4B}+1)$
10	PUCNCA P10A	----	31	PUCNCB P10B	----
11	PUICA P11A	----	32	PUICB P11B	----
12	PUICA P12A	$(N_3-1)(N_{2A}+1)(N_{2B}+1)$	33	PUICB P12B	$(N_3-1)(N_{2B}+1)(N_{2A}+1)$
13	PUICA P13A	$(N_3-1)(N_{2A}+1)(N_{4B}+1)$	34	PUICB P13B	$(N_3-1)(N_{2B}+1)(N_{4A}+1)$
14	PUICA P14A	$(N_3-1)(N_{4A}+1)(N_{2B}+1)$	35	PUICB P14B	$(N_3-1)(N_{4B}+1)(N_{2A}+1)$
15	PUICA P15A	$(N_3-1)(N_{4A}+1)(N_{4B}+1)$	36	PUICB P15B	$(N_3-1)(N_{4B}+1)(N_{4A}+1)$
16	PUINCA P16A	----	37	PUINCB P16B	----
17	PDCCA P17A	$(N_1-1)(N_{2B}+1)$	38	PDCCB P17B	$(N_1-1)(N_{2A}+1)$
18	PDCNCA P18A	$(N_{2A}-1)$	39	PDCNCB P18B	$(N_{2B}-1)$
19	PDICA P19A	$(N_3-1)(N_{2B}+1)$	40	PDICB P19B	$(N_3-1)(N_{2A}+1)$
20	PDICA P20A	$(N_3-1)(N_{4B}+1)$	41	PDICB P20B	$(N_3-1)(N_{4A}+1)$
21	PDINCA P21A	$(N_{4A}-1)$	42	PDINCB P21B	$(N_{4B}-1)$

* only counts that change are listed

TERMINOLOGY:

NEW	=	NEW TRACK	DCC	=	DROP CORRECT CORR
UCC	=	UPDATE ON CORRECT CORR	DCNC	=	DROP CORRECT NO-CORR
UCNC	=	UPDATE ON CORRECT NO-CORR	DIC	=	DROP INCORRECT CORR
UIC	=	UPDATE ON INCORRECT CORR	DINC	=	DROP INCORRECT NO-CORR
UINC	=	UPDATE ON INCORRECT NO-CORR			

TABLE 3 EVENT PROBABILITIES

PNEWA	=	ν_A	PNEWB	=	ν_B
PUCCA	=	$N_1 \tau_A$	PUCCB	=	$N_1 \tau_B$
PUNCA	=	$N_{2A} \tau_A$	PUNCB	=	$N_{2B} \tau_B$
PUICA	=	$N_3 \tau_A$	PUICB	=	$N_3 \tau_B$
PUINCA	=	$N_{4A} \tau_A$	PUINCB	=	$N_{4B} \tau_B$
PDCCA	=	$N_1 \mu_A$	PDCCB	=	$N_1 \mu_B$
PDCNCA	=	$N_{2A} \mu_A$	PDCNCB	=	$N_{2B} \mu_B$
PDICA	=	$N_3 \mu_A$	PDICB	=	$N_3 \mu_B$
PDINCA	=	$N_{4A} \mu_A$	PDINCB	=	$N_{4B} \mu_B$

A number of probabilities in Table 4 should be discussed.

- P[No Incorrect Tracks in Correlation Gate]
  $= 1 - \exp(-\lambda_B G_A)$ (3)
  where:
  $$\lambda_B = (N_{2B} + N_{4B})/\text{Area}$$
  - Tracks are randomly distributed.
- P[Correct Track Falls in the Correlation Gate]
  $= 1 - \exp(-\alpha_A/2)$ (4)
  where:
  - Track position is assumed to be normally distributed.

- P[Correct Pairing is Closest Track]

$$= PCL_A(\alpha,\lambda) = \int\int P_1(r)\ P_2(r)\ r\,dr\,d\alpha$$

$$= \frac{[1- \exp(-.5+\lambda_B\pi)\alpha]}{[1 + 2\lambda_B\ ][1 - \exp(-\alpha/2)]} \quad (5)$$

where:

$P_1(r)$ = P[Correct Pairing is r units Away]
= Normal with mean = 0 and variance = 1.0.

$P_2(r)$ = P[No other tracks in area πr^2].
$= 1 - \exp(-\lambda\pi r^2)$

- P[Correct Pairing in Decorrelated]

$$= \exp(-\beta_A/2) \quad (6)$$

- P[Incorrect Pairing is Decorrelated]

$$= PDCR_A(\beta) = \int P_1(r)\ P_2(r)\ dr \quad (7)$$

where:

$P_1(r)$ = P[Incorrect Pairing is r units Away]
= Uniform Density over [0, 9.0]

$P_2(r)$ = P[Incorrect Pairing has a Report Outside the Decorrelation Gate $\pi\beta$ /Incorrect Pairing is r units Away]
= Noncentral Chi-Square with 2 Degrees of Freedom and Noncentrality Parameter r.

The Chapman-Kolmogorov forward equation is given by:

$$\frac{d\,P_N(t)}{d\,t} = -\ C_N(t)\ P_N(t) + \sum_{\substack{M \\ M \neq N}} P_{N,M}(t)\ P_M(t) \quad (8)$$

where:

$P_N(t)$ = The probability that the C/D process is in system state N at time t,

$C_N(t)$ = The probability rate that the C/D process will transition out of state N in the next increment of time at time t,

$P_{N,M}(t)$ = The probability rate that the C/D process will transition from state M to state N at time t.

The term, $C_N(t)$, is obtained by summing over all transitions in Table 2 which result in a new end state. The resultant sum is given by

$$C_N(t) = \nu_A + \nu_B + r_A\ (N_1 esp(-\beta_A/2) + N_3\ PDCR_A) + r_B\ (N_1 esp(-\beta_B/2) + N_3\ PDCR_B) + \mu_A\ N_A + \mu_B\ N_B \quad (9)$$

The first two terms are the new track arrival rate; the second two terms describe the rate at which decorrelations occur and the last two terms reflect the rate at which tracks are dropped.

The transition probability rates, $P_{N,M}(t)$, are obtained by consulting the transitions in Table 2 and considering

$$N = (N_1, N_{2A}, N_{2B}, N_3, N_{4A}, N_{4B})$$

as the end state, and

$$M = (M_1, M_{2A}, M_{2B}, M_3, M_{4A}, M_{4B})$$

as the start states where $M \neq N$. The total number of such transitions is 34.

This Markov process is homogeneous i.e., the transitions probabilities are time independent, and ergodic. Ergodicity is obtained since the number of system states is finite and every state cam be reached from every other state with positive probability [1]. Any state can be reached from any other state through a sequence of NEW, UPDATE and DROP track events with appropriate decisions and outcomes. Because the process is ergodic, the limiting distribution exist and can be obtained by setting the left hand side of (8) to zero.

The solution to (8) for the transient and limiting state probabilities $P_N(t)$, has proven to be especially difficult. The dimensionality of the state space, the non-linear dependence of the transition probabilites on the system state and the inability to lump the Markov process into a lower dimensional process have precluded solving (8) at this time. Further theoretical work will be required to address these difficulties.

SUMMARY

This report has shown that the correlation/decorrelation process can be modelled as a Markov process. Solution of the Chapman-Kolmogorov equation for the transient and limiting state probabilities has proven to be exceedingly difficult due in part to the state space dimensionality and to the non-linear, non-lumping character of the transition probabilities. New theoretical tools will be required to solve the Chapman-Kolmogorov equation even for the simple C/D process modeled in this paper.

REFERENCES

[1] Takacs, Lajos. Stochastic Processes. Science Paperbacks, Chapman and Hall, London, England, 1966.

TABLE 4 DECISION/OUTCOME PROBABILITIES

DECISION/OUTCOME PROBABILITY	DESCRIPTION OF TERM
$P1A = (1 - \gamma_A) \times \exp(-\lambda_B G_A)$	Correct pairing does not exist; and no other track is in the gate.
$P2A = \gamma_A \times N_3/(N_3 + N_{2B}) \times \exp(\lambda_B G_A)$	Correct pairing exists; but is already incorrectly paired, and no other track is in the gate.
$P3A = \gamma_A \times N_{2B}/(N_3 + N_{2B}) \times \exp(-\alpha_A/2) \times \exp(-\lambda_B G_A)$	Correct pairing exists; and is not correlated; but is not in the gate; and no other track is in the gate.
$P4A = \gamma_A \times N_{2B}/(N_3 + N_{2B}) \times (1 - \exp(\alpha_A/2) \times PCL_A$	Correct pairing exists; and is not correlated; and is in the gate; and is the closest.
$P5A = (1 - \gamma_A) \times (1 - \exp(-\lambda_B G_A)) \times N_{2B}/(N_{2B} + N_{4B})$	Correct pairing does not exist; but another track is in the gate; and was a correct no-corr.
+	
$\gamma_A \times N_3/(N_3 + N_{2B}) \times 1 - \exp(-\lambda_B G_A) \times N_{2B}/(N_{2B} + N_{4B})$	Correct pairing exists; but is incorrectly correlated; and another track is in the gate; and was a correct no-corr.
+	
$\gamma_A \times N_{2B}/(N_3 + N_{2B}) \times \exp(-\alpha_A/2) \times (1 - \exp(-\lambda'_B G_A)) \times N_{4B}/(N_{2B} + N_{4B}-1)$	Correct pairing exists; and is not correlated; but is not in the gate; but another track is in the gate; and was an incorrect no-corr.
+	
$\gamma_A \times N_{2B}/(N_3 + N_{2B}) \times (1 - \exp(-\alpha_A/2)) \times (1 - PCL_A) \times N_{4B}/(N_{4B} + N_{2B}-1)$	Correct pairing exists; and is not correlated; and is in the gate; but is not closest; and the closest was an incorrect no-corr.
$P6A = (1 - \gamma_A) \times (1 - \exp(-\lambda_B G_A)) \times N_{4B}/(N_{2B} + N_{4B})$	Correct pairing does not exist; but another track is in the gate; and track was an incorrect no-corr.
+	
$\gamma_A \times N_3/(N_3 + N_{4B}) \times (1 - \exp(-\lambda_B G_A)) \times N_{4B}/(N_{2B} + N_{4B})$	Correct pairing exists; but is incorrectly correlated; and another track is in the gate; and the track was an incorrect no-corr.
$P7A = \gamma_A \times N_3/(N_3 + N_{2B}) \times \exp(-\alpha_A/2) \times (1 - \exp(-\lambda_B G_A)) \times N_{2B}/(N_{4B} + N_{2B})$	Correct pairing exists; but is incorrectly correlated; and another track is in the gate; and track was an incorrect no-corr.
+	
$\gamma_A \times N_{2B}/(N_3 + N_{2B}) \times (1 - \exp(-\alpha_A/2)) \times (1 - PCL) \times (N_{2B}-1)/(N_{4B} + N_{2B}-1)$	Correct pairing exists; and is not correlated; and is in the gate; but is not the closest; and other track is a correct no-corr.
$P8A = 1 - \exp(-\beta_A/2)$	Keep a correct pairing.
$P9A = \exp(-\beta_A/2)$	Decorrelate correct pairing.
$P10A = 1.00$	Keep a correct no-corr.
$P11A = 1\text{-}PDCR_A$	Incorrectly keep a miscorrelated track.
$P12A = PDCR_A/4$	Correctly decorrelate leaving a correct no-corr and a correct no-corr.
$P13A = PDCR_A/4$	Correctly decorrelate leaving a correct no-corr and an incorrect no-corr.

DECISION/OUTCOME PROBABILITY	DESCRIPTION OF TERM
P14A = $PDCR_A/4$	Correctly decorrelate leaving an incorrect no-corr and a correct no-corr.
P15A = $PDCR_A/4$	Correctly decorrelate leaving an incorrect no-corr and an incorrect no-corr.
P16A = 1.00	Keep an incorrect no-corr.
P17A = 1.00	Drop a correctly correlated track.
P18A = 1.0	Drop a correctly no correlated track.
P19A = .5	Drop an incorrectly correlated track leaving a correct no-corr.
P20A = .5	Drop an incorrectly correlated track leaving an incorrect no-corr.
P21A = 1.00	Drop an incorrectly no correlated track.

TERMS:
- $\lambda_B^1 = (N_{2B} + N_{4B} - 1) / AREA$
- $\lambda_B = (N_{2B} + N_{4B})/AREA$
- γ_A = Probability correct pairing from source B exists in the track file.
- $G_A = \alpha_A \pi$ = Gate area.
- β_A = Decorrelation threshold.
- $PCL(\alpha_A, \lambda_B)$ = Probability correct pairing is closest.
- $PDCR(\beta_A)$ = Probability incorrect pairing is decorrelated.

*NOTE: Analogous decision/outcome probabilities exist for the source B events: p22B through p42B with all terms with subscript A replaced with subscript B and vice versa.

The Need for Multiple Hypothesis Data Correlation Methods in Multi-Target Tracking

B. Belkin and W. R. Stromquist

Daniel H. Wagner, Associates

It is often the case in multi-target tracking problems that insufficient target signature information is available to resolve the individual target identities (the *discrimination* problem) or even to associate sensor reports unambiguously into constructed target tracks (the *correlation* problem). In some cases, the difficulty is that the targets are intrinsically similar. In the ocean surface surveillance context multiple warships of the same or closely related classes may be operating in close proximity as part of a battle group. In some applications deceptive decoys may be employed as a counter-surveillance measure. In other situations the indistinguishability of the targets may be more a consequence of the limitations of the detection sensor than the similarilty of the targets. A radar return from a merchant ship may look exactly like that from a warship.

False alarms, at least to the extent that they tend to be isolated and statistically uncorrelated events, constitute a different aspect of the correlation problem. Here the difficulty is to avoid the degraded tracking performance that can result from spurious correlations between false alarms and valid target tracks. False alarms in significant numbers can add greatly to the number of feasible correlations and hence to the combinatorial growth of the correlation problem.

The aspect of the uncertain data correlation problem of interest to us here is the unavoidable trade-off between probabilistic accuracy and computational complexity. Our main thesis is twofold. First, methods based on saving only a single target track hypothesis at each processing stage may involve a significant loss of information resulting in seriously inaccurate track estimates. Second, methods based on processing multiple data association hypotheses in parallel are available which are computationally tractable and largely avoid these inaccuracies.

In the first section below we briefly describe a particular type of single hypothesis method sometimes referred to as probabilistic data association (see [1]) developed for tracking a single valid target in a background of clutter. We present two examples which show that this method, like other single hypothesis methods may produce unsatisfactory results when applied to a problem involving multiple valid targets. In the second section we describe a multiple hypothesis approach using the concept of hypothesis *clustering* first introduced by Reid in [2]. Various successful implementations of this approach have been developed for ocean surveillance applications including the Multiple Association Tracker/Correlator (MATCH) algorithm developed by the authors (see [3]), the ISATS algorithm developed by SAIC ([4]), and an over-the-horizon tracker/correlator developed at the Johns Hopkins University Applied Physics Laboratory ([5]).

1. The Inadequacy of Single Hypothesis Methods

A data association or correlation hypothesis is a tentative matching of a set of sensor reports with their assumed target of origin. When the number of targets generating sensor reports is unknown and when the possibility of false alarms is present, the potential exists for a very large number of correlation hypotheses to be generated. The role of a correlation algorithm is to screen the list of correlation hypotheses rapidly and efficiently and, using appropriate measures of report-to track consistency, discard the ones that are judged infeasible or to have too little likelihood to be worth preserving. The algorithm should then assign relative weights to the hypotheses that survive the screening process based on statistical or other measures of internal consistency. The most widely used consistency criterion is position/velocity, but many others are possible depending on the precise nature of the sensor information. Most correlation schemes use Bayesian methods to calculate these weights.

In practice, most correlation algorithms are recursive. After each stage of processing, estimates are made of each target's position, velocity, and possibly other characteristics, along with their associated uncertainties. In the most general situation, what is retained for each target is something we will call a *target map*, which gives a probability distribution for the target's state in some arbitrary state space (position, velocity, etc.). Sometimes target maps are called *tracks*, and the memory area they occupy is called a *track file* or *track data base*. The basic operation of the algorithm is to update the track data base, based on each new sensor report or group of reports.

A single hypothesis algorithm is one which, after each stage in processing, retains only one target map for each target. A multiple-hypothesis algorithm, as the name implies, retains multiple hypotheses or *scenarios*, each of which contains a target map for each target. The scenarios may be very different; e.g., they may contain different numbers of targets. Along with each scenario, the algorithm retains a probability (or relative probability) of that scenario being correct.

Most research-oriented correlation algorithms are multiple-hypothesis algorithms, but most operational surveillance systems are single-hypothesis. A hazard in designing large systems is that top-level designers, being unaware of the limitations of single-hypothesis algorithms and doubting the feasibility of multiple-hypothesized algorithms, will commit themselves to a single-hypothesis design before getting down to the details of the correlation problem itself. Our thesis is that users of correlation algorithms should be aware of the multiple-hypothesis alternative.

Two kinds of single-hypothesis algorithms. We identify two kinds of single-hypothesis algorithms: maximum likelihood data association (MLDA) and probabilistic data association (PDA). Both methods begin each processing stage by listing the plausible report-to-track correlation hypotheses, and determining their relative probabilities (or weights). The MLDA method proceeds by selecting the most likely association hypotheses, updating all target maps on the basis of this correlation hypothesis, and discarding all other possibilities. The PDA method updates each target map by forming a weighted composite of the updates that would result from each correlation hypothesis considered. The PDA method, if implemented without too many compromises, has the advantage of producing the best possible target map for each target, given that only one target map can be retained in memory.

We have little to say about the MLDA method; although widely used, it has obvious limitations. Sooner or later a situ-

ation will arise in which the most likely association hypothesis is not the true one. All further processing is then built on a false foundation. Subsequent reports, which ought to clarify the situation, may appear in stark contradiction to the only list of target maps retained in memory. The errors may compound, leading to lost targets, mislocated targets, and spurious new targets.

The problems with PDA are more subtle and are best demonstrated by example. We will give two examples: one using continuous target distributions, and one using discrete distributions.

Example 1. Assume that multiple targets are being tracked in two-dimensional space, so that target state corresponds to location only. Assume further that each target map consists of a Gaussian distribution in two-space, corresponding to the conventional notion of an uncertainty ellipse for each target. Essentially, each target is represented by a location mean and covariance, updated at each stage for assumed target motion and for new sensor reports using the PDA method. This is essentially the algorithm in which PDA was introduced in [1], and it is a natural one because it allows the recursive updating of tracks by Kalman filtering. An extension is to let target state consist of both position and velocity, and to admit a motion model in which velocity is constant or slowly changing; this formulation still supports Kalman filtering.

Let $\mathbf{x}$ and Σ represent the mean and covariance for a particular target, after the motion update for a particular stage in processing. If there is no ambiguity in report-to-track association, and if we associate with this target a report with mean $\mathbf{y}$ and covariance Δ, then the target map is updated as follows:

$$\begin{cases} \Sigma' = \left(\Sigma^{-1}+\Delta^{-1}\right)^{-1} & \text{(new mean)} \\ \mathbf{x}' = \Sigma'\left(\Sigma^{-1}\mathbf{x}+\Delta^{-1}\mathbf{y}\right) & \text{(new covariance).} \end{cases} \quad (1)$$

Now suppose, instead, that several reports $(\mathbf{y}_1,\Delta_1),\ldots,(\mathbf{y}_M,\Delta_M)$ are candidates for association with the given target, and suppose that their association probabilities are $\alpha_1,\ldots,\alpha_M$. (Essentially, α_i is the sum of the weights, or relative probabilities, of all of the report-to-track scenarios which cause report $(\mathbf{y}_i, \Delta_i)$ to be associated with this target.) Let α_0 be the probability that no current report corresponds to this target, and normalize so that $\alpha_0 + \alpha_1 + \ldots + \alpha_M = 1$.

The PDA method applied in the present context would then proceed according to the following steps:

(1) Compute the *conditional* posterior target state vector and error covariance estimates for each possible current-stage association.

(2) Weight each conditional posterior estimate by the corresponding α_i.

(3) Compute the unconditioned state vector and error covariance estimates.

(4) Approximate the posterior target state distribution as Gaussian with mean and variance as determined below.

The M + 1 conditional updates, corresponding to the M possible reports and the no-report hypothesis, are as follows:

$$\begin{cases} \Sigma_i' = \left(\Sigma^{-1}+\Delta_i^{-1}\right)^{-1} \\ \mathbf{x}_i' = \Sigma_i'\left(\Sigma^{-1}\mathbf{x}+\Delta_i^{-1}\mathbf{y}_i\right), \text{ for } i = 1,\ldots,M. \end{cases} \quad (2)$$

and

$$\begin{cases} \Sigma_0' = \Sigma \\ \mathbf{x}_0' = \mathbf{x}. \end{cases}$$

The mean and covariance of this unconditioned (weighted) distribution are given by

$$\begin{cases} \mathbf{x}' = \sum_{i=0}^{M} \alpha_i \mathbf{x}_i' \\ \Sigma' = \sum_{i=0}^{M} \alpha_i \left[\Sigma_i' + (\mathbf{x}_i'-\mathbf{x}')(\mathbf{x}_i'-\mathbf{x}')^T\right]. \end{cases} \quad (3)$$

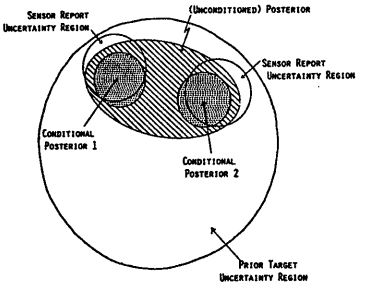
Note that the combined covariance Σ' has both a within-components part and a between-components part. The latter can dominate, leading to a final covariance which is much larger than the covariance from any of the individual updates.

In a PDA algorithm using Gaussian distributions, only the values of $\mathbf{x}$ and Σ are saved to represent the target in the next stage of processing, and for future calculations of association weights, motion updates, etc., the target location distribution is assumed Gaussian.

A schematic representation of the special case M = 2 is shown in Figure 1. The prior target location (two standard deviation) uncertainty region is large compared to the uncertainty regions associated with each of the two sensor reports. The resulting conditional posterior uncertainty regions are shaded and the composite unconditioned posterior uncertainty region is hatched.

FIGURE 1

EXAMPLE 1: MODIFIED KALMAN FILTER UPDATE SCHEMATIC REPRESENTATION



Two important observations are to be made based on Figure 1:

(1) The mode of the posterior target location distribution is an unlikely target location no matter which of the two sensor reports is actually associated with the target.

(2) The size of the posterior uncertainty region significantly overstates the size of the region that one would have to search to contain 86.5% of the target location probability mass (corresponding to a 2σ-containment probability).

One concludes based on this example that the PDA method of [1] is not likely to produce satisfactory results when applied to multiple valid target tracking problems.

Example 2. Our interest now is in the behavior of single hypothesis methods over the course of repeated updates for target motion and for locational sensor data. For this purpose we shift from a continuous target space representation to a discrete one. The specific problem conditions are as follows:

(1) Target motion follows a symmetric random walk (equal probabilities for each of the possible transitions to adjacent cells) on a 10 x 10 cellular grid.

(2) Sensor reports always localize the detected target in a single cell.

(3) No false alarms.

(4) Target location distributions are initialized as independent and uniform over the entire grid.

The assumed method of processing is the discrete space analogue of the continuous space PDA scheme described in Example 1. Focus attention on a specific target and define

$f_{k|k}(x)$ = k^{th} processing stage posterior location distribution of given target evaluated at cell x.

Suppose, again, that there are M simultaneous reports to be processed at stage k, and that by one means or another relative association weights $\alpha_k^{(j)}$, j = 0,1,...,M have been calculated. Let

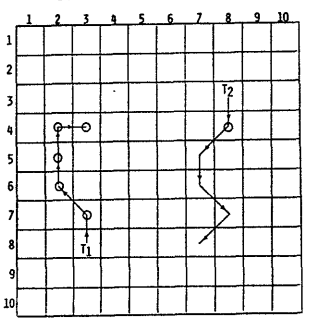
$f_{k|k}^{(j)}(x)$ = k^{th} processing stage *conditional posterior* location distribution given association of the j^{th} sensor report with the target (j=0 if no association is made).

(Note that $f_{k|k}^{(0)} = f_{k|k-1}$.) Then the appropriate PDA update formula is

$$f_{k|k}(\mathbf{x}) = \sum_{j=0}^{M} \alpha_k^{(j)} f_{k|k}^{(j)}(\mathbf{x}). \qquad (4)$$

For our example we specialize to the case of two targets T_1 and T_2 as shown in Figure 2. Here a target track is simply a succession of five cell-to-cell transitions, at the rate of one transition per update stage. The circled target positions indicate the occurrence of sensor reports, each of which serves to mark the presence of a target (but not its identity) in the associated cell. Note that target T_1 is detected during each of the five update stages. Target T_2 is somehow less detectable and is detected only once, during stage k = 1.

FIGURE 2
EXAMPLE 2: ASSUMED TARGET TRACKS



NOTE: CIRCLED POSITIONS INDICATE SENSOR REPORTS.

Tables 1(a) and 1(b) show the target location distribution at the end of the five processing stages, first based on the PDA method and second, based on the multiple hypothesis method which we will introduce in the next section.

TABLE 1(a)
EXAMPLE 2: POSTERIOR TARGET LOCATION DISTRIBUTION
(PDA METHOD)

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1
3	32	34	32	2	0	1	1	1	1	1
4	38	41	536	4	1	1	1	2	2	1
5	44	50	41	7	1	1	1	1	1	1
6	13	18	12	6	2	1	1	1	1	1
7	9	12	8	5	1	1	0	0	0	0
8	3	4	3	2	1	0	0	0	0	0
9	1	2	2	1	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

TABLE 1(b)
EXAMPLE 2: POSTERIOR TARGET LOCATION DISTRIBUTION
(MULTIPLE HYPOTHESIS METHOD)

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	1	3	5	7	6	4
2	0	0	0	1	3	8	13	16	15	9
3	0	0	0	1	5	12	20	24	23	14
4	0	0	500	1	6	14	23	28	27	17
5	0	0	0	1	5	12	20	24	23	14
6	0	0	0	1	3	8	12	15	14	9
7	0	0	0	0	1	3	5	6	6	4
8	0	0	0	0	0	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

NOTES: (1) DIVIDE EACH ENTRY BY 1000 TO OBTAIN CELL CONTAINMENT PROBABILITY.
(2) BY THE INHERENT SYMMETRY OF THE PROBLEM THE ABOVE DISTRIBUTIONS APPLY TO BOTH TARGET T_1 AND T_2.

One sees in Table 1(a) further evidence of the target mislocalization exhibited by the PDA method. It is known for certain from the Stage 1 sensor data that there are at least two targets present in the region and that at the end of processing Stage 5 there must be a target within four distance units (i.e., four cell-to-cell transitions) of cell (4,8). It is apparent from Table 1(a), however, that this condition is violated and that most of the target mass has been drawn to the vicinity of the more detectable of the two targets, T_1. A search plan based on the target location distributions in Table 1(a) would assign no effort at all to cell (8,7), the actual location of T_2.

The problem with the PDA method is that the repeated hybridization implicit in iterated applications of Equation (4) entails a systematic loss of memory. Most of the detection activity is in the left part of the surveillance region and that is where Bayes' theorem will put increasing amounts of target mass if the constraint that a target must be present in the right part of the region is ignored. There is, of course, nothing wrong with Bayes' theorem; the problem lies in the way it is applied in the PDA method.

The situation in Table 1(b) shows a dramatic contrast with 1(a). When the multiple hypothesis method is used, the target location distributions of T_1 and T_2 as represented are precisely as one would expect. The probability mass in cell (4,3), which is known to contain a target, is exactly 1 (shared equally between the two constructed targets). The upper right portion of the region also contains a total of one unit of target mass reflecting the fact that there must be a target within four cell-to-cell transitions of cell (4,8). The distribution of target probability over the upper right portion of the region is fairly uniform as one expects based on the assumed symmetric random walk model for target motion.

In summary, then, the examples we have presented argue strongly in favor of multiple hypothesis methods over single hypothesis methods. We now go on to discuss some of the implementational aspects of multiple hypothesis processing.

2. Feasibility of Multiple Hypothesis Methods

Multiple hypothesis methods retain in memory, after each stage of processing, more than one *scenario*, along with the relative probabilities that each scenario is correct. Each scenario contains a target map for each target under surveillance, and hence represents a coherent description of the target population, but the various scenarios may be in sharp disagreement with each other.

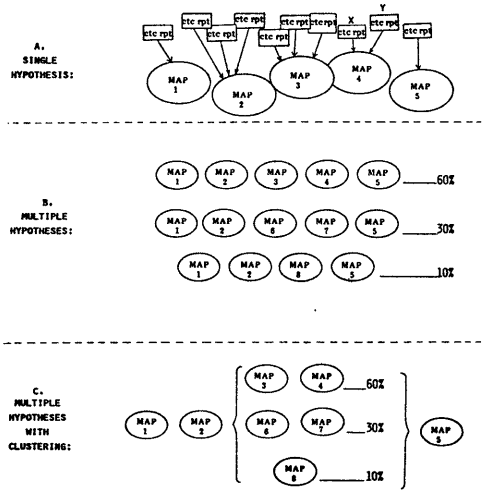
At each stage of processing, as in single-hypothesis algorithms, a list is made of plausible report-to-track correlation hypotheses, with their relative probabilities, although now a distinct list is required for each scenario. A new scenario is generated for each old scenario and each plausible report-to-track correlation hypotheses. The result is a combinatorial explosion in the list of scenarios: left alone, the number of scenarios in memory will tend to grow at least exponentially in the number of processing stages. Because of this, all multiple-hypothesis algorithms include various devices for pruning the scenario list, and for merging scenarios which are substantially identical by some standard. The success of the algorithm depends on the ability of these devices to control the growth of the scenario list without discarding scenarios with reasonable likelihood.

We believe that the success of a multiple-hypothesis algorithm depends on efficient representation of the knowledge contained in the track data base; i.e., in the list of scenarios. In this section we will describe a simple but powerful technique called *clustering*, which we believe to be the most useful technique for this purpose. Clustering was introduced in the context of multi-target tracking by Reid [4]. The basic idea is to exploit the fact that virtually all multi-target tracking problems can be statistically decoupled into subproblems, each of which can be solved independently. The complexity of the problem is determined not by its overall size, but the size of the largest subproblem.

Figure 3 illustrates the use of clustering in simplifying a hypothetical scenario list. Figure 3A shows how a single-hypothesis algorithm (with MLDA) might combine a handful of sensor reports into a single scenario with five target maps (presumably in several processing stages.) Suppose that all associations are unambiguous except that report x might actually belong with map 3, and that if so, then report y might belong there also (meaning that there would be only four targets present). Then a multiple-hypothesis algorithm will retain a scenario list such as in Figure 3B, with some list of relative probabilities as shown.

FIGURE 3

REPRESENTING KNOWLEDGE OF MULTIPLE TARGETS

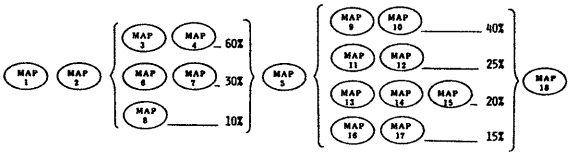


The key to clustering is to recognize that the unambiguous target maps do not need to be duplicated in memory, even to the extent of duplicate pointers. Thus the same information can be stored as suggested schematically in Figure 3C.

The advantage of clustering appears when there are multiple clusters, as in Figure 4. Some number from 7 to 9 of targets are being tracked, with two distinct areas of ambiguity, each represented in Figure 4 by a cluster. A total of seven *local scenarios* are stored, with seven relative probabilities. The same information could be stored without clustering, but more memory would be required: there would be twelve global scenarios, with twelve relative probablities.

FIGURE 4

HOW CLUSTERING COMBATS "COMBINATORIAL EXPLOSION"



WITH CLUSTERING: 7 LISTS, 7 PROBABILITIES

WITHOUT CLUSTERING: 12 LISTS, 12 PROBABILITIES

With clustering, adding more areas of ambiguities causes memory and processing time to increase more-or-less linearly. Without clustering, the requirements grow exponentially.

MATCH (reference [3]), a tracker-correlator algorithm developed for research purposes by the authors, incorporates a multiple-hypothesis methodology made practical by clustering. As in the case of other algorithms based on the same principles, it has been necessary to meet various technical challenges: methods have been found for forming, merging, and simplifying clusters, introducing new target maps and deleting targets, and updating the entire scenario list for new sensor reports. In a test using the ECAP I data base, a realistic simulation developed by NRL of ELINT sensor data for ocean surveillance, MATCH has dramatically out-performed an operational single-hypothesis algorithm. Other multiple-hypotheses algorithms have performed similarly.

The lessons we have learned in ocean-surveillance applications of MATCH are that success depends on efficient representation of knowledge in the track data base; that multiple hypotheses are required; and that efficient representation of multiple hypotheses requires clustering.

References

[1] Y. Bar-shalom, "Tracking Methods in a Multi-target Environment," IEEE Trans. Automatic Control AC-23 1978.

[2] D. B. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Automatic Control AC-24, 843-854 (1979).

[3] W. R. Stromquist, B. Belkin, and N. L. Gerr, Multiple Association Tracker Correlator (MATCH) Program Design Documents, Daniel H. Wagner, Associates Memorandum Report to Naval Research Laboratory, February 15, 1984.

[4] K. Askin, C. Osgood, and B. Kaufman, "ISATS Testing Using ECAP II Data," Draft Report (1985).

[5] G. Mitzel, P. Barnett, B. Kuehne, and S. Sommerer, "Wide-Area Correlation and Tracking of Surface Ships Using Multiple Sensors," Johns Hopkins APL Technical Digest, January-March 1984, Vol. 5, No. 1.

LIMITED SENSING ALGORITHMS FOR COMMUNICATION NETWORKS USING CARRIER-SENSE MULTIPLE ACCESS CHANNELS

Lazaros Merakos and P. Papantoni-Kazakos

Electrical Engineering and Computer Science Dept.
University of Connecticut
Storrs, CT. 06268

ABSTRACT

We consider the random multiple access of a packet-switched, broadcast channel, for the Poisson user model in a local area network environment, where "carrier-sensing" techniques are possible due to small propagation delays. We examine a representative algorithm from a new class of stable algorithms with "limited sensing" characteristics. Limited sensing algorithms require that users sense the channel only while they have a packet to transmit, and, therefore, they have practical advantages over algorithms that require continuous channel sensing, especially when users are mobile. The algorithm's throughput-delay characteristics are obtained using results from the theory of regenerative processes and infinite dimensionality linear systems.

1. INTRODUCTION

Random-access algorithms (RAA's) allow large numbers of independent, dispersed, bursty users to exchange messages over a shared communication channel under distributed control. Applications include: packet-radio networks, local area networks, cellular radio local distribution systems, satellite communication networks, and computer communication networks. When many users share a single channel with no central coordination, simultaneous transmissions leading to message "collisions" are inevitable. Colliding messages are treated as transmission errors and each user strives to retransmit its message till it is correctly received. All users employ the same algorithm for this purpose, and have to resolve their conflicts by utilizing no other source of information except for the one acquired by having listened to the past transmissions. This information is called channel feedback information, or simply feedback. In slotted (synchronous) channels, feedback is usually modelled as providing the "outcome" of the transmission in each slot, where various such outcomes may be distinguished, depending on the physical environment. The most commonly assumed feedback is the ternary, which distinguishes between empty, busy with one message (or successful), and busy with at least two messages (collision) slots.

The existence of feedback broadcast (per slot) and its level is an entirely different issue than its sensing by the users. Different system characteristics induce various feedback sensing limitations, and the latter are instrumental in the development of RAA's. In fact, different levels of feedback sensing limitations induce different classes of algorithms. Two such prominent classes that represent two extreme cases in feedback sensing limitations are described by the "ALOHA-type" and the "tree-search-type" RAA's.

The "ALOHA-type" RAA's [1] require the minimal level of feedback sensing, among all the RAA's that utilize feedback broadcast. In particular, each user monitors the feedback, only each time that he attempts transmission. Unfortunately, such minimal feedback does not allow for the development of stable algorithms, for the Poisson user model, (effectively infinite number of users whose cumulative traffic is modeled by a Poisson process). Stability, here, means maintaining the total data traffic generated by the users with finite average delay. To attain stability, the knowledge of some channel"history" is necessary. Thus, while the feedback sensing required by the "ALOHA-type" algorithms is attainable in most systems, it leads to severe instabilities.

The "tree-search-type" RAA's [2,3] have been proposed as remedies to the instability of the "ALOHA-type" algorithms. The "tree-search-type" algorithms are indeed stable, and perform better than the "ALOHA-type" algorithms. However, they imply a quite severe drawback. They require that the users sense the feedback continuously (full feeback sensing), even when they are not in the sysetm. This last requirement is in fact an integral part in the operation of the "tree-search-type" algorithms which operate synchronously. Such extreme synchronization is clearly unrealizable in most of the systems that the "tree-search-type" algorithms address. Such is the case in mobile radio and in multi-hop systems, for example, where users move in and out the feedback broadcast range. The "ALOHA-type" with retransmission control algorithms [4] require full feedback sensing as well; they thus imply the same level of extreme synchronization as the "tree-search-type" algorithms do.

A user in a random-access system can almost always sense the feedback broadcast continuously, during active (for himself) time periods. Such a time period extends from the time instant when the user generates some data packet to the time when this packet is successfully transmitted. The above feedback sensing level is indeed feasible in local area networks, in mobile radio environments, and in multi-hop systems (unless in the last two, a user moves out of the broadcast range before successful transmission, in which case he stops being part of the system), and it is called limited feedback sensing. Fortunately, it is possible to devise stable random-access algorithms, subject to limited feedback sensing considerations. We call the class of those protocols, limited sensing class, and we point out that it was initiated in the Soviet Union in 1980 [5]. The limited sensing class has high potential, due to its very attractive properties in terms of performance, operational complexity, and applicability.

In this paper we examine a RAA which is representative of the limited sensing class. The algorithm is designed to operate in a local area network environment, where "carrier sensing" techniques are possible due to small propagation delays. The algorithm of this paper also serves as an example for the illustration of a systematic method for evaluating the delay performance characteristics of a large class of RAAs. The method is based on a powerful theorem referring to regenerative processes, in conjuction with results from the theory of infinite dimensionality linear systems.

2. USER AND CHANNEL MODEL

We assume that an infinite population of independent, bursty, packet-transmitting users share a common communication channel. We model the packet arrival process as homogeneous Poisson with intensity λ packets per unit of time. For convenience, we assume that packets are of fixed length, and we take the packet transmission time to correspond to our unit of time We also assume that the propagation delay between any two users in the network is at most α, where $\alpha<1$. For simplicity in analysis, we assume that the time axis is slotted, where the slot size is equal to the maximum propagation delay α. Users may initiate a packet transmission only at the beginning of a slot.

We consider limited channel sensing and ternary feedback. That is, each user senses the channel continuously, from the time instant when he generates a packet, to the time instant when this packet is successfully transmitted, and he can distinguish without error among the following channel states: a) idle (no transmission) b) success (transmission of a single packet) c) collision (simultaneous transmission of at least two packets). We assume that a collision results in complete loss of the information included in all the involved packets; thus, retransmission is then necessary.

Without loss of generality, we assume that a user who senses the channel can distinguish between transmission (success or collision) and no transmission (idle) instantaneously. However,the time required to distinguish a collision from a successful transmission (collision detect time) is a system characteristic whose value depends on the maximum propagation delay, the transmission medium, the packet encoding and modulation techniques, and the method used to detect collisions. In this paper it will be assumed that it takes β units of time before the transmitting users determine the interference and abort their transmissions, where $\alpha\leq\beta\leq1$. We refer to the parameter β as the "conflict truncation time"; for packet-radio networks it is common to assume that $\beta=1$, since users cannot listen to the channel while they are transmitting; for cable networks, where users have early collision detection capabilities, it is commonly assumed that $\beta<1$.

3. THE ALGORITHM AND ITS GENERAL OPERATION

In this section we describe a limited channel sensing algorithm, which allows users to communicate with each other in a carrier-sensing environment satisfying the assumptions specified in the previous section.

The algorithm is implemented by each "busy" user in a distributed fashion. A user is defined to be busy from the moment it generates a new packet for transmission until the moment after the same packet is successfully transmitted, otherwise, the user is said to be idle. The time instant that a user generates a packet, (i.e., when he becomes busy), he starts sensing the channel and he simultaneously initializes the algorithm; he continues to sense the channel until the successful transmission of his packet, (i.e., until he becomes idle); upon the occurrence of this event, he stops sensing the channel and simultaneously he terminates the algorithm.

For the implementation of the algorithm the user uses a counter, whose indication at time t is denoted by CI_t. The indications of the counter dictate the operation of the algorithm, which is described as follows:

Rule 1 -- Counter initialization

Let the user generate a new packet at time t_o, and let k_o denote the first slot boundary, after t_o. Also, let k_1 denote the first slot boundary after t_o at which the user senses the channel idle. Then at k_1, the user initializes his counter as follows:

$$CI_{k_1} = \begin{cases} 1 ; & \text{if } k_1 = k_o \\ M ; & \text{if } k_1 \neq k_o \end{cases}$$

where M is a random variable uniformly distributed on $\{1,2,\ldots,m\}$,and the integer m, $m\geq1$, is an algorithm parameter.

Rule 2 -- Transmission rule

The user transmits at the beginning of the slots at which his counter indication equals "1".

Rule 3 -- Counter updating

After the user has initialized his counter he updates it only at the slot boundaries at which he senses the channel idle.

Let k_2, k_3 ... denote these slot boundaries in accordance with their occurrence. Let the user be busy at k_i(i=1, 2,3,...), with $CI_{k_i}\geq1$. Then, at time k_{i+1}he updates his counter as follows:

a) If $CI_{k_i}\geq1$, then

$$CI_{k_i\,1} = \begin{cases} CI_{k_i}-1 & \text{if, during } (k_i, k_{i+1}), \text{ he senses the channel idle} \\ CI_{k_i}+m-1 & \text{if, during } (k_i, k_{i+1}), \text{ he senses the channel busy with a successful transmission} \\ CI_{k_i}+m+n-1 & \text{if, during } (k_i, k_{i+1}), \text{ he senses the channel busy with a collision} \end{cases}$$

where the integer n, $n\geq2$, is an algorithmic parameter.

b) If $CI_{k_i}=1$ and, during (k_i, k_{i+1}), he senses the channel busy with a collision, then

$$CI_{k_{i+1}} = m + J$$

where J is a random variable uniformly distributed on $\{1,2,\ldots,n\}$.

If $CI_{k_i}=1$ and, during (k_i, k_{i+1}), he senses the channel busy with a successful transmission, then his packet has been successfully transmitted and he becomes idle.

The integers m and n used in the description of the algorithm are design parameters, subject to optimization for throughput maximization.

The general operation of the algorithm is perhaps better illustrated by introducing the concept of a "stack". A stack is an abstract storage device consisting of an infinite number of cells, labelled 1,2,3,... . The number of packets that a cell can accommodate is unrestricted. At each time t during the operation of the algorithm, users with counter value $CI_t=r$ can be thought of as having stored their packets in cell #r of the stack. A packet is transmitted whenever it enters cell #1 of the stack. Packets are, eventually, successfully transmitted after moving through the cells of the stack in accordance with the algorithmic rules described above.

The execution of the algorithm by each busy user induces on the time axis an alternate sequence of transmission periods (successful or unsuccessful) and idle periods. Let t_i(i=0,1,2,...) denote the consecutive slot boundaries at which the channel is idle. The interval $[t_i, t_{i+1})$ will be referred to as the ith algorithm step. If during an algorithm step the channel is idle, busy with a successful transmission, or busy with a collision, then the algorithm step will be called idle, successful, or unsuccessful, respectively. An idle algorithm step lasts α units of time; a successful algorithm step lasts for $1+\alpha$ units of time, one unit of time to place the packet onto the channel and α units of time for this packet to clear the channel due to propagation delay; an unsuccessful algorithm step lasts for $\beta+\alpha$ units of time, β units of time for the transmitting users to detect the collision and abort their transmission

and α units of time for the packet fragments to clear the channel.

The description of the general operation of the algorithm and its analysis are greatly facilitated if one considers how the state of the stack evolves at the beginning of consecutive algorithm steps. In figure 1 the stack is imbedded at t_i and t_{i+1} to show how packets move through the cells of the stack, (i.e., how users update their counters), as well as to show how new packets arriving between t_i and t_{i+1} place themselves in the cells of the stack, (i.e., how users initialize their counters), depending on whether the algorithm step was idle, successful or unsuccessful.

As it can be seen from figure 1, the operation of the algorithm is based on the "divide and conquer" philosophy that characterizes most RAAs; specifically, the algorithm spreads the incoming traffic into the first m cells of the stack to, a priori, avoid collisions, when the new traffic is heavy, (e.g., after a successful transmission). Furthermore, to resolve collisions, it uniformly splits the group of collided packets into n cells of the stack. The algorithm described in this section will be referred to as the LAN Stack algorithm (LANSA).

4. DELAY ANALYSIS

To analyze the performance of the LANSA we introduce the concept of a session. A session is a sequence of consecutive algorithm steps that begins and ends at two consecutive algorithm renewal instants. These instants are denoted by R_n, $n\geq 1$, and are determined by means of a conceptual marker that operates on the stack. The first session begins with the beginning of the first algorithm step, at $R_1=t_1$, with the marker placed at cell #2. During the session, the marker's position in the stack is adjusted at the beginning of each algorithm step. At t_i, let the marker be at cell $\#c_i$, $c_i\geq 2$; then, at t_{i+1} the marker is placed at cell $\#c_{i+1}$, with

$$c_{i+1} = \begin{cases} c_i - 1 & \text{if the i-th algorithm step is idle} \\ c_i + m - 1 & \text{if the i-th algorithm step is successful} \\ c_i + m + n - 1 & \text{if the i-th algorithm step is unsuccessful} \end{cases}$$

where the integers $m\geq 1$, $n\geq 2$ are as defined in the LANSA description.

The second renewal instant, R_2, is the instant at which the marker drops to cell #1 for the first time; that is, $R_2=\min\{t_i>R_1 : c_i=1\}$; this signifies the end of the first session. Instantaneously, at R_2, the marker is then adjusted to cell #2 and the second session begins. This process continues indefinitely.

Denote by S_i the number of packets successfully transmitted during the ith session. Since all the packets arriving while a session is in progress are successfully transmitted by its end, S_i also represents the number of arrivals during the ith session. From the definition of a session, in conjuction with the memoryless property of the Poisson arrival process, we have that $\{S_i\}$ is a sequence of i.i.d. random variables.

Let the arriving packets be labelled n=1,2,..., according to the order of their arrival instant. Let $\mathcal{D}_n$ denote the delay experienced by the nth packet (time interval between the time of its arrival and the time of its destination). Also, let M_i denote the total number of packets that were successfully transmitted during the first i sessions. The sequence $\{M_i\}_{i\geq 0}$ is a renewal sequence, since $M_0=0$, and $M_{i+1}=M_i+S_{i+1}$, $i\geq 0$. From the renewal properties of the algorithm, it can be seen that, for every $i\geq 0$, the process $\{\mathcal{D}_{M_i+n}\}_{n\geq 1}$, is a probabilistic replica of the process $\{\mathcal{D}_n\}_{n\geq 1}$. Thus, the process $\{\mathcal{D}_n\}_{n\geq 1}$ is regenerative with respect to the renewal sequence $\{M_i\}_{i\geq 0}$, with common regenerative cycle, S, the number of packets successfully transmitted over a session. The next theorem is a combination of theorem 2 and corollary 2 of [6].

Theorem 3 If S is not periodic, with $S\triangleq E(S)<\infty$, and if $T\triangleq E(\sum_{i=1}^{S}\mathcal{D}_i)>\infty$, then there exists a real number D such that

$$D = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\mathcal{D}_i = \lim\frac{1}{n}E(\sum_{i=1}^{n}\mathcal{D}_i) \text{ with probability 1}$$

Furthermore, $\mathcal{D}_n$ converges in distribution to a random variable $\mathcal{D}_\infty$, and

$$D = E(\mathcal{D}_\infty) = T/S < \infty \qquad (1)$$

From the rules of the algorithm and the definition of the session, it can be seen that sessions always end with an idle algorithm step. Thus, at the beginning of session all the cells of the stack are empty except from cell #1, which contains K packets, where K is a random variable with $P(K=k)=p_k(\lambda\alpha)^k\exp(-\alpha\lambda)/(k!)$. Let S_k, $k\geq 0$, denote the expected number of successful transmissions during a session that starts with k packets in the first cell of the stack, (such a session will be referred to as a session of multiplicity k). Then,

$$S = \sum_{k=0}^{\infty} p_k S_k \qquad (2)$$

The quantity T in (1) represents the mean cumulative delay experienced by all packets transmitted during a session. If we let T_k denote the mean cumulative delay of a session of multiplicity k, $k\geq 0$, then

$$T = \sum_{k=0}^{\infty} p_k T_k \qquad (3)$$

The dynamics of the algorithm yield the following for the S_k's and T_k's.

$$S_o=o,\ S_k = f_k + \sum_{i=1}^{\infty} a_{ki}S_i,\ k>1 \qquad (4)$$

$$T_o=o,\ T_k = g_k + \sum_{i-1}^{\infty} a_{ki}T_i,\ k>1 \qquad (5)$$

where the nonnegative coefficients a_{ki}, f_k, g_k are determined by the system characteristics. (Their expressions can be found in [7]).

From (1), (2), (3), (4) and (5) we see that the mean packet delay induced by the algorithm can be computed from the solution of the infinite dimensional linear systems (4) and (5). To determine a solution (if any) to a system such as (4) or (5) we follow the following steps.

Step 1: Find conditions under which the system has a unique, nonnegative solution.
Step 2: Show that the algorithmic sequence of interest, i.e. $\{S_k\}$, or $\{T_k\}$ coincides with the unique solution.
Step 3: Develop arbitrarily tight upper and lower bounds on the solution.

The bounds developed in step 3 are then used in (2), (3), and (1) to obtain bounds on the mean packet delay.

We, next, proceed following the steps outlined above.

Step 1

For convenience, we rewrite an infinite linear system in an operator form. Specifically, let E be

the space of sequences $X=\{x_\nu\}: A\to R$, where A is a countable set. Also let E^L be the subspace of E for which

$$\sum_{\nu\in A} |C^L_{\mu\nu} X_\nu| < \infty \ , \ \mu\in A$$

We define the operator $L=\{L_\mu(X)\} : E^L \to E$ as follows:

$$L_\mu(X) = b^L_\mu + \sum_{\nu\in A} C^L_{\mu\nu} X_\nu, \quad \mu\in A, \ X\in E^L$$

In this notation, systems (4), (5) can be written in the form

$$Z^L = L(Z^L), \ Z^L \in E^L \tag{6}$$

We are interested in the existence and uniqueness of nonnegative points $Z^L\in E^L$, that satisfy (6); such points will be referred to as fixed points of L, and represent solutions to the corresponding infinite linear system of equations. It can be shown that to establish the existence of a nonnegative fixed point Z^L, it suffices to find a point $X^o\in E^L$, such that

$$0\le L(X^o) \le X^o \tag{7}$$

A point X^o, satisfying (7), also serves as an upper bound on Z^L. Furthermore, to establish a lower bound on Z^L, it suffices to find a point $Y^o\in E^L$, such that,

$$0<Y^o\le L(Y^o) \le X^o \tag{8}$$

Thus, under (7) and (8), we have that,

$$0<Y^o \le S^L \le X^o \tag{9}$$

System (4) - Existence of solution: System (4) corresponds to an operator L_1 with $b^{L_1}_\mu = f_\mu$, $C^{L_1}_{\mu\nu} = a_{\mu\nu}$, $\mu, \nu \in N$, where N is the set of natural numbers. Let $X^o = \{\gamma k+\delta\}_{k\ge 1}$ and $Y^o = \{\gamma' k+\delta'\}_{k\ge 1}$. It can be shown that if $\lambda<\lambda_o(\alpha,\beta,m,n)$, then there exist $\gamma,\delta,\gamma',\delta'$ such that (7) and (8) are satisfied; $\lambda_o(\alpha,\beta,m,n)$ is the unique solution of the equation

$$\frac{1-(1+m\alpha)\lambda}{1-m(1-S)} - \lambda \frac{\beta+(m+n)\alpha}{(1-r)m+(1-q)n-1} = 0 \tag{10}$$

;where, $S=\exp(-(m^{-1}+\alpha)\lambda)$, $r=\exp(-(\beta m^{-1}+\alpha)\lambda)$, $q=(1-n^{-1})^2\exp(-\alpha\lambda)$. Thus, if $\lambda<\lambda_o(\alpha,\beta,m,n)$, system (4) has a solution $Z^{L_1}=\{z^{L_1}_k\}$, and by (9)

$$0<\gamma' k+\delta' \le z^{L_1}_k \le \gamma k+\delta \ , \ k\ge 1 \tag{11}$$

The coefficients γ,δ,γ', and δ' are bounded functions of λ, whose expressions can be found in [7].

System (5) - Existence of solution: Following the same procedure as for system (4), it can be shown that if $\lambda<\lambda_o(\alpha,\beta,m,n)$, system (5) has a solution $Z^{L2}=\{z^{L2}_k\}$, such that,

$$0<\zeta' k^2+\eta' k+\theta' \le z^{L2}_k \le \zeta k^2+\eta k+\theta, \ k\ge 1 \tag{12}$$

where the coefficients of the quadratic bounds are bounded functions of λ, whose expressions can be found in [7].

System (4) and (5) - Uniqueness of solution: It can be shown that the solutions Z^{L1}, Z^{L2} are unique in the class

$$E_2 = \{X: \sup_{k\in N} \frac{|x_k|}{k^2} < \infty\}$$

The main tools used in the proof can be found in [8].

Step 2

In step 1, we have established a sufficient condition for the existence of nonnegative solutions to the systems of interest, and we have identified a class of sequences in which these solutions are unique. Next we must show that the algorithmic sequences $\{S_k\}_{k\ge 1}$, $\{T_k\}_{k\ge 1}$, belong to the identified class, and, therefore, coincide with the unique solution in the class. The proof is essentially the same as the one used in [5], and is omitted.

Step 3

The bounds found in step 1 can be improved by solving finite systems of linear equations that are truncations of the original infinite systems. In systems (4), (5), we replace all the unknowns for $i>i_o$, i_o a constant, with the known upper (lower) bounds given by (11),(12),respectively. It can be shown that the solution of the resulting finite system is a new, tighter upper (lower) bound to the solution of the original system. Furthermore, these bounds can be made arbitrarily tight by increasing i_o. In our computations we used $i_o = 20$. The computed bounds on S_k, T_k, for $k\le 20$, in conjuction with the bounds given by (11), (12) for $1\le k<20$, were then used in (4), (5), and, finally, in(1) to obtain an upper bound D_u and a lower bound D_ℓ on the mean packet delay D. The bounds D_ℓ, D_u for $\beta=1$ and $\beta=\alpha$ and various α choices, are plotted in figures 2, and 3, respectively. (The difference between D_ℓ and D_u is indistinguishable since they coincide up to the fourth decimal point.) Given α,β, let $\eta(\alpha,\beta)$ denote the maximum stable throughput attained by the LANSA, defined as $\eta(\alpha,\beta)\triangleq \sup\{\lambda: D<\infty\}$. Then, since $\lambda<\lambda_o(\alpha,\beta,m,n)$ is only a sufficient condition for finite mean packet delay, we have that

$$\eta(\alpha,\beta) \ge \bar{\lambda}(\alpha,\beta)$$

where $\bar{\lambda}(\alpha,\beta) \triangleq \sup_{m,n} \lambda_o(\alpha,\beta,m,n) = \lambda_o(\alpha,\beta,m^*,n^*)$.

The values of $\bar{\lambda}(\alpha,\beta)$ as well as of the optimal parameter choices m^*,n^*, for representative values of α and β, are given in table 1. Comparing the results shown in figures 2, 3 and table 1 to the results reported in [4] reveals that the LANSA out performs the optimally controlled version of the traditional nonpersistent CSMA and CSMA-CD algorithms for all values of the propagation delay α.

5. CONCLUSIONS

We presented and analyzed a representative "limited sensing" random access algorithm for the carrier-sense multiple access channel. The analysis revealed that the limited sensing class provides algorithms that combine modest channel sensing requirements with inherently stable operation and high performance. The delay analysis method used in this paper is of independent interest, since it can be applied to a wide class of RAA's whose operation exhibits a regenerative character.

REFERENCES

[1] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - Carrier sense multiple access modes and their throughput-delay characteristics," IEEE Trans. Comm., vol. COM-23, pp. 1400-1416, Dec. 1975.

[2] J.I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Trans. Inform. Theory, vol. IT-25, pp. 505-515, Sept. 1979.

[3] D. Towsley and G. Venkatesh, "Window random access protocols for local computer networks," IEEE Trans. Comp., vol. C-31, pp. 715-722, Aug. 1982.

[4] J. S. Meditch, and C.T.A. Lea, "Stability and optimization of the CSMA and CSMA/CD channels," IEEE Trans. Comm., vol. COM-31, no. 6, pp. 763-773, June 1983.

[5] B. S. Tsybakov and N.D. Vvedenskaya, "Stack algorithm for random multiple access," Problemy Peredachi Informatsii, vol. 15 no. 3, pp. 80-94, Jul./Sept., 1980.

[6] S. Stidham, Jr., "Regenerative processes in the theory of queues, with applications to the alternating-priority queue," Adv. Appl. Prob., vol. 4, pp. 542-577, 1972.

[7] L. Merakos, "Stack algorithms for CSMA and CSMA-CD channels," Tech. Rep. TR-84-6, EECS Dept., Univ. of Connecticut, Storrs, CT, Apr. 1984.

[8] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A unified method for delay analysis of random multiple access algorithms," Tech. Rep. TR-85-8, EECS Dept., Univ. of Connecticut, Storrs, CT., Aug. 1985.

	LANSA $\beta = 1$			LANSA $\beta = 0.5$			LANSA $\beta = \alpha$		
α	$\bar{\lambda}$	m*	n*	$\bar{\lambda}$	m*	n*	$\bar{\lambda}$	m*	n*
0.500	0.348	1	4	0.371	1	3	0.371	1	3
0.400	0.390	1	4	0.419	1	4	0.427	1	3
0.300	0.442	1	4	0.480	1	4	0.498	1	3
0.200	0.510	2	4	0.561	1	4	0.597	1	3
0.100	0.629	2	5	0.676	2	4	0.743	1	3
0.050	0.723	3	6	0.771	2	5	0.849	1	3
0.020	0.816	5	9	0.855	4	7	0.931	1	3
0.010	0.867	7	12	0.898	5	9	0.964	1	3
0.005	0.904	10	16	0.928	7	12	0.981	1	3
0.002	0.938	16	24	0.954	11	17	0.992	1	3
0.001	0.956	22	33	0.968	16	24	0.996	1	3

Table 1. The lower bound $\bar{\lambda}$ on the maximum stable throughput of the LANSA, and the parameters m* , n* for representative values of the propagation delay α and the conflict truncation time β.

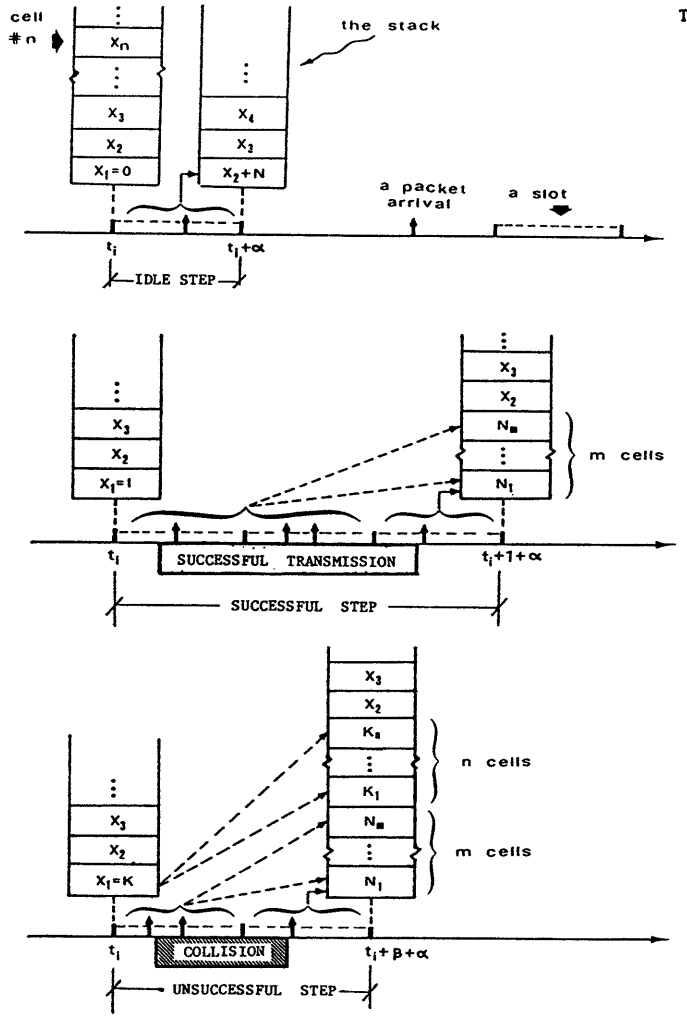


Figure 1.

Illustration of the ith algorithm step using the stack. X_i denotes the number of packets in cell i, at t_i; $N_1 + N_2 + \ldots + N_m = N$, where N is the number of new arrivals, and $K_1 + K_2 + \ldots + K_n = K \geq 2$.

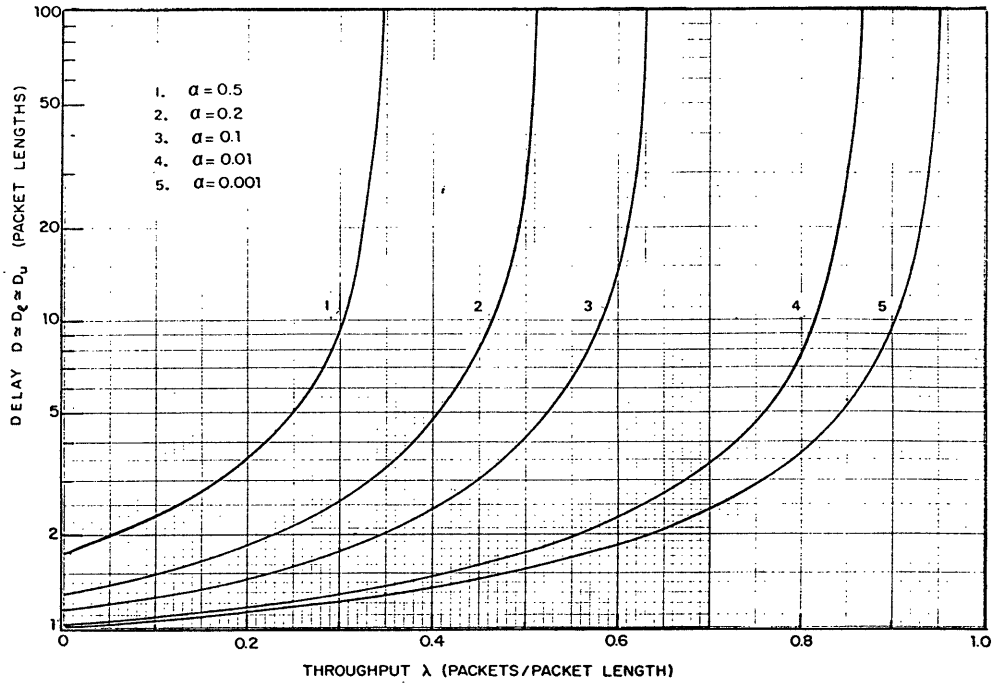


Figure 2. Mean packet delay versus throughput for LANSA ($\beta = 1$).

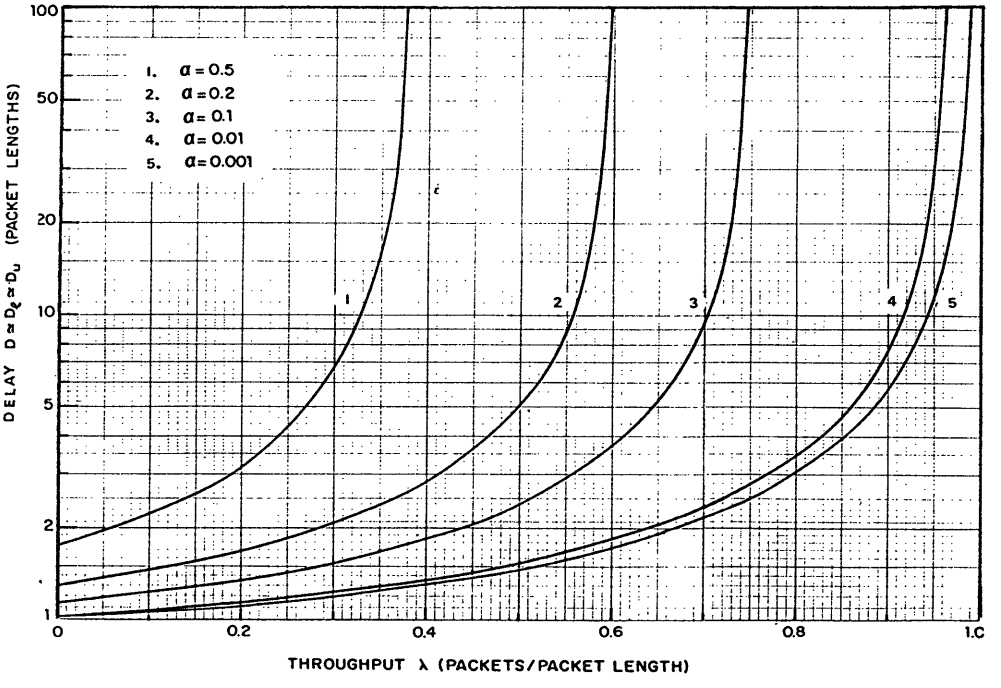


Figure 3. Mean packet delay versus throughput for LANSA ($\beta = \alpha$).

Broadcast Communication Policies for Distributed Aeroacoustic Tracking

J.R. Delaney
M.I.T. Lincoln Laboratory

R.R. Tenney
ALPHATECH,Inc.

Abstract

The behavior of any distributed estimation system is highly dependent on the information flow within the system. Broadcast communications simplify the information flow, even when they have limited range. But different policies for deciding what information to broadcast and when produce different information flows and behaviors. This paper addresses two distinct policies for what information to broadcast and a spectrum of policies for when to broadcast, all in the context of a particular distributed aeroacoustic tracking system [3]. Results are reported for a series of experiments tracking a helicopter.

1. The MIT/LL DSN Testbed

The MIT Lincoln Laboratory (MIT/LL) Distributed Sensor Network (DSN) project investigates concepts and algorithms for distributed systems which employ many geographically dispersed, identical sensor/processor nodes connected by a limited range broadcast communications network for surveillance and target tracking. The focus is the MIT/LL DSN testbed which uses acoustic sensors to detect and track low-flying, subsonic aircraft.

The primary testbed sensors are arrays of relatively inexpensive microphones laid on the ground over an area about 5 meters in diameter; there is one such array per node. By using sophisticated signal processing [1], helicopters and similar aircraft can be detected at ranges of several kilometers and their azimuths measured with standard deviations of a few degrees. Broadcast communications within the testbed are currently simulated using an Ethernet.

2. The Distributed Aeroacoustic Tracking Algorithm

The DSN tracking algorithm is described in detail an earlier paper [3]. It is reviewed in this section for the reader's convenience. Figure 1 illustrates the data flows into, out of, and within the algorithm for a single node. The figure also shows the component algorithms and data bases. Local azimuth measurements and foreign (received from other nodes) azimuth and position tracks are the data flowing in. Local azimuth and position tracks are the data flowing out (broadcasts to other nodes). These same data flow into and out of the component algorithms and data bases.

Local azimuth measurements are used to create and maintain local azimuth tracks. Each such track consists of an estimate of a target's perceived azimuth and azimuth rate, the associated covariance matrix, and a unique identifier. These quantities are computed using a conventional, two-state Kalman Filter. "Perceived" means that the effects of acoustic propagation delay are not removed at this stage. For subsonic targets, actual and perceived azimuths can differ by up to sixty degrees.

A node broadcasts a local azimuth track when that track satisfies certain criteria that are described in a later section. Upon receipt of an azimuth track broadcast, a node tries to combine that foreign azimuth track with a local azimuth track to initiate a new local position track. Position and velocity estimates are calculated by solving four simultaneous nonlinear equations, the coefficients of which are functions of the local and foreign track estimates and of the node locations. The associated covariance matrix is computed from the local an foreign azimuth track covariance matrices by linearizing the measurement equation at the estimated position and velocity. Track initiation fails if there is no solution to the four equations, if the solution is outside either node's nominal coverage, or if the r.m.s position variance is too large. Successful track initiation effectively consumes the azimuth tracks used. A unique identifier is assigned to each position track when it is created.

Local azimuth measurements can also be used to maintain local position tracks. An Extended Kalman Filter (EKF) with minor modifications is used to update the local position tracks. Local azimuth measurements are associated preferentially with local position tracks and used in this way. Only those left over are used in creating and maintaining local azimuth tracks.

A node broadcasts a local position track when the track satisfies certain criteria that are dicussed in a later section. Upon receipt of a foreign position track, the position track combining algorithm checks whether there is a local position track with the same identifier. If not, the foreign position track is copied into the local position track file. Otherwise, the two position tracks must be merged to create a new local position track based on the measurement information inherent in both the local and foreign position track.

Matching local and foreign position tracks are based on overlaping measurement information sets due to past broadcast of position tracks with the same identifier. Redundant information must be removed when combining the information in the local and foreign position tracks or some past measurement information will be overly weighted in the new local position track. Redundant information is represented in each node by a common position track stored with each local position track. The necessary information arithmetic can be done by translating all three tracks

*This work was sponsored by the Defense Advanced Research Projects Agency. The views expressed are those of the authors and do not reflect the official policy or position of the U.S. Government.

into the Fisher information form [2], adding the translated estimates and covariances of the local and foreign position tracks, subtracting those of the common track, and translating the result back into the usual form. The back-translated result is the new local position track.

More explicitly, but with all three steps run together, this method of merging matching local and foreign position tracks is as follows:

$$\hat{X}'_L = \Sigma'_L \{\Sigma^{-1}_L \hat{X}_L + \Sigma^{-1}_F \hat{X}_F - \Sigma^{-1}_C \hat{X}_C\}$$

$$\Sigma'_L = \{\Sigma^{-1}_L + \Sigma^{-1}_F - \Sigma^{-1}_C\}^{-1}$$

where $\hat{X}$ and Σ are estimates and covariances; the subscripts L, F, and C indicate local, foreign, and common position track elements; and "'" indicates a new value.

Computation of common position tracks is simplified by ignoring the influence of acceleration on the measurement information accumulated by different nodes. A common track is then a track based on all measurement information that is implicit in all the local or foreign position tracks with the same identifier ever broadcast or received by that node. When a new foreign position track is received by the node and copied into the local track data base, an identical common position track is created and stored with new local position track. After a local position track is broadcast, the corresponding common position track must set equal to it. And when a foreign position track is received, any new inherent measurement information must be incorporated into the common position track just as it is into the local position track.

3. What to Broadcast

The above description mentions the broadcast of local azimuth and position tracks. In each case, the information sent includes a state estimate, covariance matrix, and unique identifier. The state consists of azimuth and azimuth rate for azimuth tracks and x,y positions and velocities for position tracks. Under certain assumptions, that is enough to support the tracking algorithm. The assumptions are

- Broadcast communications directly link all nodes which can detect a target,
- No broadcast messages are ever lost, and
- Nodes broadcast local position tracks far enough apart in time for all nodes to complete the processing of a target before more information is received for the same target.

The assumptions are required to keep common tracks consistent in all nodes using the position combining algorithm described above [3].

The first assumption is fairly easy to satisfy in practice. The second is not; two limited range broadcast transmissions can interfere with each other at a receiver between the transmitters without the interference being detectable at either transmitter. And in a distributed system, the last assumption is also difficult to satisfy. It can be approximated by artificially losing foreign position tracks which arrive too soon. Such an artifice is feasible because the tracking algorithm is robust enough to tolerate lost messages.

The impact of lost messages was studied [4] using a symbolic simulation of the position track combining algorithm. The simulation kept track of the measurement information sets underlying matching local and common position tracks in three nodes. Loss of a particular local position track broadcast leaves a "hole" in the information set where the message was not received. Depending on the pattern of broadcasts, that hole will be filled in or will migrate around among the matching local and common position tracks in all nodes as they communicate. But the hole never grows, so the worst that happens is a degradation in performance that dies out with time as the missing information becomes less relevant to the present. Of course, if the loss of local position track broadcast is a frequent occurance, the overall degradation of tracking performance can be severe. That will happen if local position tracks are broadcast frequently and artificially lost. The result then is roughly equivalent to placing a limit on the rate of local position track broadcasts.

Experimentation showed that this effective rate limitation resulted in unsatisfactory tracking performance under circumstances that are illustrated in the next section. Broadcasting both local and common position track pairs gets around the effective rate limit. The foreign common position track received with a foreign position track is removed from that specific foreign position track when carrying out the information arithmetic. This extracts measurement information from the foregn track that was not previously broadcast by the foreign node. Simultaneous broadcasts no longer interfere with maintenence of the local and common tracks. Lost messages again cause holes in the measurement information sets. In this case, the holes are never filled in, but they migrate less. And the broadcast communications need only indirectly connect the nodes which can detect a target. That results in common tracks which are only common among directly connected nodes, but this limited commonality has no impact on tracking peformance because of the way the common tracks are used.

4. When to Broadcast

The earlier paper [3] introduced a broadcast policy based on critical events. In order to prevent loss (or poor use) of measurement information, a node must broadcast a local position track (and the matching common track if desired) when:

- A new local position track is created by the position track initiation algorithm,
- A local position track estimates a target has just entered the coverage of a foreign node, or
- A local position track estimates a target has just left the coverage of the local node.

Upon the occurance of the last kind of event, the local and common position track pairs are deleted from the local data base.

As a result of this policy, position track initiation need never be repeated and a position track identifier is effectively a unique target identifier. A target's position tracks follow it through a DSN as nodes pass the tracks along. Figure 2 illustrates the resulting sequence of events the produce this effect.

The definitions of the last two critical events require a node to estimate its own and other nodes' sensor coverages. So far, this has been done by assuming a nominal detection range and compensating for motion during acoustic propagation. Compensation skews the nominally circular coverage alone a target's heading. Experimental results have shown this model to be a poor one in some cases, such as that discussed in the next section, but that the algorithms can operate satsifactorily nevertheless.

After some experimentation, two additional criteria for broadcasting position tracks were introduced. The first requires that the r.m.s. position covariance of a local track be less than some threshold in order for a broadcast to occur. An unusually large variance usually reflects a dearth of recent measurement information in the position track; it is not worth broadcasting. The second requires that a local/common position track pair be broadcast whenever the time since it was last broadcast crosses a threshold. Making the second threshold large causes the criterion to be suppressed; making it small causes broadcasts after each new set of local azimuth measurements is processed. Generally speaking, the more frequent the broadcasts, the more measurement information available at each node and the more accurate each node's local position tracks. This effect can be significant shortly after position track initiation. The point is illustrated in the next section.

Criteria for broadcasting local azimuth tracks were not discussed in the earlier paper [3]. Experimentation lead to the adoption of criteria paralleling the last two position track broadcast criteria. A local azimuth track is broadcast only if its azimuth variance is less than a threshold. And it is broadcast whenever the time since it was broadcast exceeds a threshold. Making the second threshold large cause a local azimuth track to be broadcast just once, when it is first sufficiently accurate. Making the second threshold small causes a local azimuth track to be broadcast then and following every measurement until it is used for position track initiation by the local node.

Experimentation has shown that broadcasting a local azimuth track only when it is first sufficiently accurate is a poor idea. Track initiation fails if the azimuth tracks used do not differ sufficiently in estimated azimuth; the position covariance is too large. As a target approachs a pair of nodes, it is possible for their local azimuth tracks to become quite accurate before there is sufficient difference in the azimuth estimates. Broadcasting only at that time would be useless.

5. Experimental Results

The preceding sections alluded to experimentation with broadcast policies. A considerable number of experiments were carried out with both real and synthetic prerecorded local azimuth measurement data. Many were focused on testing aspects of the tracking algorithm other than the broadcast policy or simply for verifying the correct implementation of the algorithm. Real data was typically used to produce realistic and possibly unanticipated demands on the algorithm and its implementation. Synthetic data was used for carefully controled experimentation.

In the course of the experimentation, one particular set of measurement data turned out to be particularly demanding. It is real data taken of a helicopter flying over Hanscom Airfield and past the MIT/LL complex on an approximately straight path at roughly Mach 0.1. Acoustically, the environment was very noisy. Interference was noted from automobiles, construction equipment, ventilating fans, and other aircraft at a distance. Azimuth measurement data is available from four sensor arrays. Figure 3 shows the nominal flight path and the sensor locations. Milestones are marked along the flight path from west to east, the direction flown. The pilot attempted to fly over these markers as accurately as he could. Spotters called out the helicopter's passage over them. As a result, we have only approximate ground truth data.

Our experiments with this data showed that the simple model used for estimating sensor coverage was quite inadequate. Not only did nodes first detect the helicopter at different ranges, but they detected it significantly further when it was approaching than when it was receding. As a result, many nominal coverage limits resulted in alerting "downstream" nodes late or maintaining position tracks long after any new information was received.

Despite these problems the robustness of the tracking algorithm is such that good tracking performance was still obtainable. Figures 4 and 5 illustrate the point. The first shows the local helicopter position track formed and maintained by the western-most node. The other figure is for the eastern-most. Both figures show the tracks a as sequence of line segments connecting sequential position estimates. Each position estimate is surrounded by an ellipse representing the uncertainty of that estimate. Each is the equivalent of a one standard deviation error bar. For reference, each figure also shows the nominal flight path, the sensor locations, the outlines of the Hanscom runways, the MIT/LL complex, and Rte. I-95. The assumed coverage limit was 2.5 kilometers, the r.m.s. position uncertainty threshold was 0.25 kilometers, and the time threshold for broadcasts was 2 seconds.

The initial local position track uncertainty took a little while to shrink because only the first two nodes were contributing measurement information at that time and the first node was intermittantly detecting the helicopter. Once the target enters the nominal coverage of the last two nodes, they are contributing information steadily and the accuracy becomes quite good. But after the helicopter passed the last sensor, the detections became intermittant. The last detection occured when the helicopter was over Rte. I-95. After that, the position tracking algorithm just "coasts" the local position track out to the nominal coverage limit.

Small increases in the time threshold between broadcasts initially had no noticeable effect. But as the interval became larger, the transient following position track initiation became worse. In contrast, the quality of the local position track maintained by the eastern-most node degraded only slightly. Figures 6 and 7 illustrate this point. The threshold in these cases was effectively infinite. Only critical events triggered position track broadcasts. The western-most node receives no external measurement information until the second node broadcasts to alert the third node to the presence of the helicopter. Its track is rather poor until it incoporates additional information serendipitously received from the second node. But it is still somewhat awry until the second and third nodes both contribute more information as the eastern-most node is alerted. At that point, the track becomes relatively accurate. The local position track maintained by the eastern-most node shows jumps

FIGURE 3

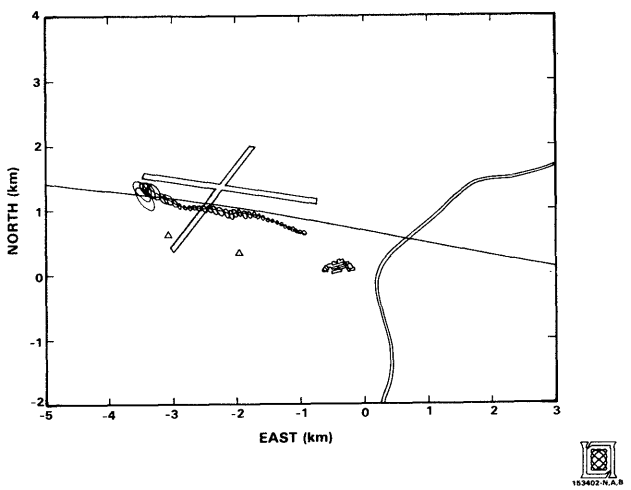


FIGURE 4

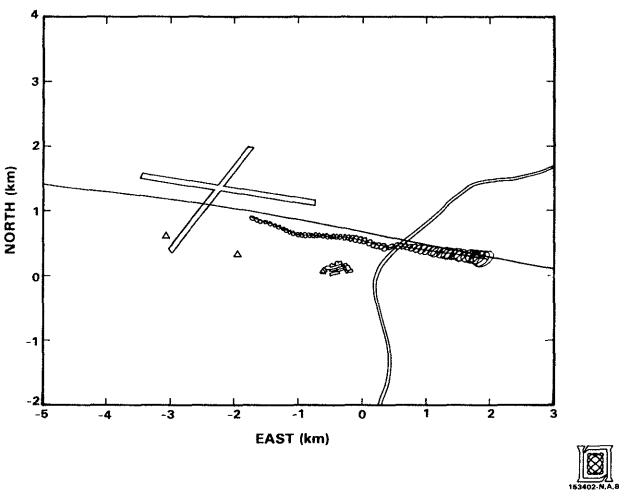


FIGURE 5

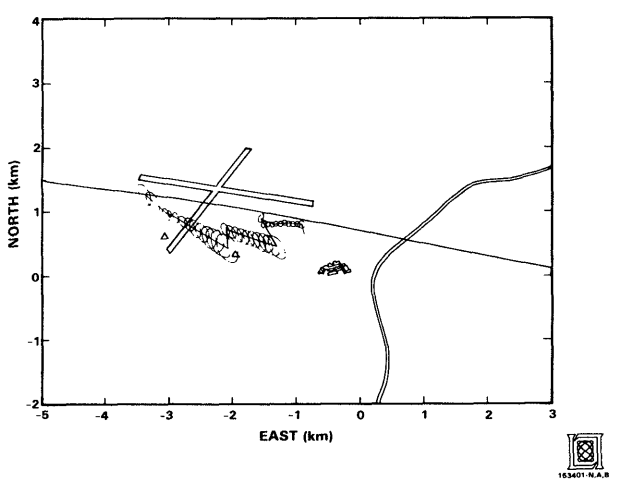


FIGURE 6

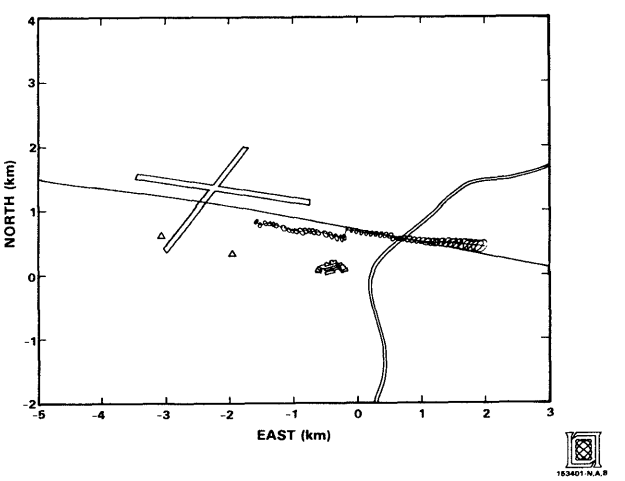


FIGURE 7

occuring when critical events trigger some sharing of information, but the overall degradation is far less.

One significant conclusion can be read from the four figures. If one has a deep DSN and is concerned only with long-term position track accuracy, broadcasting only when critical events occur can be satisfactory. But barrier DSNs, those for which a target will fly through the coverage of very few nodes, require more frequent broadcasts. They also require broadcasting local and common position track pairs to maximize the use of information and minimized information loss.

References

[1] S.H. Nawab, F.U. Dowla, and R.T. Lacoss. A New Method for Wideband Sensor Array Processing. In Proceedings of the 1984 International Conference on Acoustics, Speech and Signal Processing, pages 4.12.1-4.12.4. San Diego, March, 1984.

[2] F.C. Schweppe. Uncertain Dynamic Systems. Prentiss-Hall, Englewood Cliffs, New Jersey, 1973.

[3] R.R. Tenney and J.R. Delaney. A Distributed Aeroacoustic Tracking Algorithm. In Proceedings of the 1984 Automatic Control Conference, pages 1440-1450. San Diego, June, 1984.

[4] Distributed Sensor Networks Semiannual Technical Summary Report, MIT Linclon Laboratory, 30 September 1984.

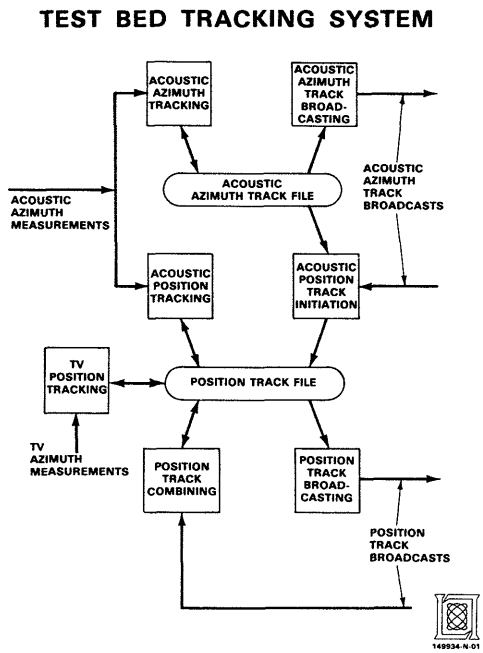


FIGURE 1

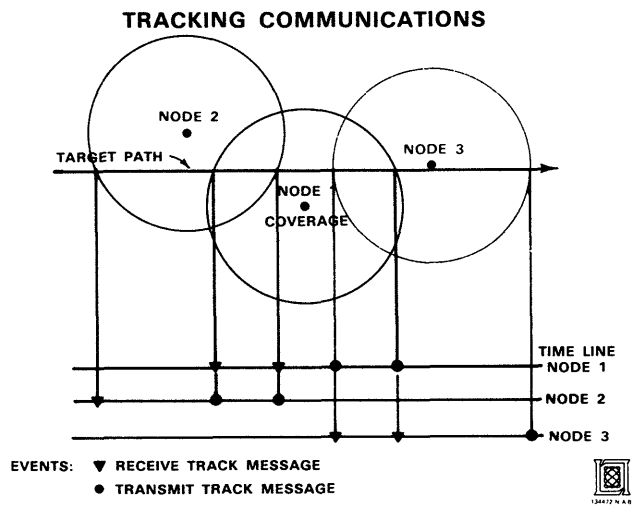


FIGURE 2

List of Attendees of the 8th MIT/ONR Workshop on C^3 Systems

Maurice F. Aburdene
Associate Professor
Dept. of EE&CS
Bucknell University
Lewisburg, PA 17837
Tel: (717) 524-1449

Gerald R. Andersen
Director
US Army Ballistic Research Lab.
ATTN: AMXBR (SECAD)
Aberdeen Proving Ground, MD 21005
Tel: (301) 278-6657

Robert Anthony
Member of Technical Staff
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-6817

Mark S. Asher
Business & Technological Systems
Aerospace Bldg., Suite 440
10210 Greenbelt Road
Seabrook, MD 20706
Tel: (301) 794-8800

Michael Athans
Professor EECS
Massachusetts Institute of Technology
Room 35-406/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-6173

Ghassan J. Bejjani
Trainee
Morgan Stanley & Co., Inc.
55 Water Street
New York, NY 10041
Tel: (212) 248-1244

Barry Belkin
Vice President
Daniel H. Wagner, Associates
Station Square One
Paoli, PA 19301
Tel: (215) 644-3400

J. Douglas Birdwell
Associate Professor
College of Engineering
Dept. of Electrical Engineering
The University of Tennessee
Knoxville, TN 37996-2100
Tel: (615) 974-5468

Kevin L. Boettcher
Graduate Student
Massachusetts Institute of Technology
Room 35-401/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-3449

Christine Bohner
Graduate Student
Massachusetts Institute of Technology
Room 35-415/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-2794

Paul L. Bongiovanni
Electrical Engineer
Naval Underwater Systems Ctr.
Building 1171-3
Newport, RI 02841
Tel: (401) 841-2432

Stuart Brodsky
Director
Electronics Systems
Sperry Corporation
12010 Sunrise Valley Drive
Reston, VA 22091
Tel: (703) 620-7922

Roger Casey
Code 84
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152
Tel: (619) 225-6268

David A. Castanon
Member Technical Staff
ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Samuel Chamberlain
Artillery Control Environment
Director: USA BRL
ATTN: AMXBR-SECAD (CHAMBERLAIN)
Aberdeen Proving Ground, MD 21005
Tel: (301) 278-6660

Gerald A. Clapp
USMC C^3I Prog. Mgr.
Naval Ocean Systems Center
Code 805
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2044

Bernard E. Clark
Research Triangle Institute
P.O. Box 12194
Research Triangle Park, NC 27709
Tel: (919) 541-6461

Marvin S. Cohen
Director
Knowledge-Based and Decision Systems
Decision Science Consortium, Inc.
7700 Leesburg Pike
Suite 421
Falls Church, VA 22043
Tel: (703) 790-0510

Lt. Col. Ronald J. Coulter
Head, Command and Control Branch
C³ Division (D103)
Development Center, MCDEC
Quantico, VA 22134
Tel: (703) 640-3161

Earl J. Craighill
Program Manager
Telecommunications Sciences Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
Tel: (415) 859-4572

John R. Delaney
Member Technical Staff
MIT Lincoln Laboratory
P. O. Box 73
Lexington, MA 02173
Tel: (617) 863-5500

A. P. Dempster
Professor
Department of Statistics
Harvard University
1 Oxford Street
Cambridge, MA 02138
Tel: (617) 495-5498

Mukund Desai
Staff Scientist
C. S. Draper Laboratory
MS 3B
555 Technology Square
Cambridge, MA 02139
Tel: (617) 258-2224

Debra Deutsch
Computer Scientist
BBN Labs
10 Moulton Street
Cambridge, MA 02238
Tel: (617) 497-3115

Marc D. Diamond
Project Engineer
FMC Corporation
Northern Ordnance Division
East Coast Engineering Office
16 Danube Drive
King George, VA 22485

Donald D. Edmonds
Department Staff
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-6808

Joseph Frank
Professor
New Jersey Institute of Technology
Newark, NJ 07102
Tel: (201) 596-3499

Gary E. Frishkorn
Associate Engineer
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707
Tel: (301) 953-5000

Sheldon B. Gardner
Electrical Engineer
Naval Research Laboratory
Code 5709
4555 Overlook Avenue, S. W.
Washington, D.C. 20375-5000
Tel: (202) 767-1167

Ashima Garg
Graduate Student
Massachusetts Institute of Technology
35-417/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-2346

John R. Gersh
Engineer
The Jonhs Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707
Tel: (301) 953-5503

Cdr. Paul Girard
Project Director
Code 252, Room 804
800 N. Quincy Street
Office of Naval Research
Arlington, VA 22217
Tel: (202) 696-4713

I. R. Goodman
Mathematician
Naval Ocean Systems Center
Bldg. 600, Rm 222, Seaside, Code 421
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2014

William E. Gutowski
Cognitive Science Engineer
ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Bruce W. Hamill
Research Psychologist
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707
Tel: (301) 953-6220

Cpt. G. Chesley Harris
Project Officer
Combined Arms Center
ATTN: ATZL-CAC-CI
Ft. Leavenworth, KS 66027-5300
Tel: (913) 684-5125

Marc Q. Jacobs
Program Manager
Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D.C. 20332-6448
Tel: (202) 767-4940

Victoria Jin
Graduate Student
Massachusetts Institute of Technology
Room 35-410/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-2346

Carl Joeckel
Manager, Command Systems
GTE - Strategic Systems Division
1 Research Drive
MS 35
Westboro, MA 01581
Tel: (617) 366-6000

Ivan Kadar
Technical Advisor
Systems Engineering
GRUMMAN Aerospace Corp.
MS B36-35
Bethpage, NY 11714
Tel: (516) 575-5359

Joseph G. Karam
ICF, Inc.
International Square
1850 K Street, N.W.
Washington, D.C. 20006
Tel: (202) 828-2970

Virginia A. Kaste
US Army Ballistic Research Lab.
ATTN: AMXBR-SECAD
Aberdeen Proving Ground, MD 21005-5066
Tel: (301) 278-6884

David L. Kleinman
Professor-Associate Dept. Head
EECS Department
University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-3066

Augustine Kong
Student
Dept. of Statistics
Harvard University
Cambridge, MA 02138
Tel: (617) 495-5496

Michael Kovacich
Scientist
COMPTEK, Inc.
100 Corporate Place
Suite B
Vallejo, CA 94590
Tel: (707) 552-3538

Elliott S. Lamb
Member Technical Staff
AT&T Bell Laboratories
Room 6E 128A
Whippany Road
Whippany, NJ 07981-0903
Tel: (301) 386-5091

Lcdr. Marvin J. Langston
Technical Director
AAW Special Projects
Naval Sea Systems Command
SEA-06Y1
Washington, D.C. 20362-5101
Tel: (202) 692-9012

Joel S. Lawson, Jr.
4773-C Kahala Avenue
Honolulu, Hawaii 96816
Tel: (808) 737-4231

Alexander H. Levis
Senior Research Scientist
Massachusetts Institute of Technology
Room 35-410/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-7262

A. Martin Lidy
Group Leader
The MITRE Corporation
MS-W954
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-6133

Peter B. Luh
Assistant Professor
Department EE&CS
University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-4821

Robert Lyons
Scientific Advisor to the
Chief Engineer
Defense Communication Agency
Code H106
Washington, D.C. 20305-2000
Tel: (703) 883-5539

Gerald S. Malecki
Scientific Officer
Code 442
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217
Tel: (202) 696-4741

Israel Mayk
Electronics Engineer
US Army CECOM-CENSEI
AMSEL-SEI-F (ATTN: MAYK)
Fort Monmouth, NJ 07703
Tel: (201) 544-4996

Dennis C. McCall
Head, C^2 Information Procressing
and Display Technology Branch
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152
Tel: (619) 225-2773

John McClure
Staff
C. S. Drapper Laboratory
MS 3B
555 Technology Square
Cambridge, MA 02138
Tel: (617) 258-2249

Michael Melich
Code 74
Naval Postgraduate School
Monterey, CA 93943
Tel: (202) 769-3959

Lazaros Merakos
Assistant Professor
Dept. of EE&CS
University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-3979

Andrew U. Meyer
Professor
Electrical Engineering Dept.
New Jersey Institute of Technology
323 Dr. Martin Luther King Blvd.
Newark, NJ 07102
Tel: (202) 596-3530

Richard C. Morgan
Business & Technological Systems
Aerospace Bldg.
Suite 440
10210 Greenbelt Road
Seabrook, MD 20706
Tel: (301) 794-8800

David F. Noble
Engineering Research Associates
8618 Westwood Center Drive
Vienna, VA 22180
Tel: (703) 734-8800

Philip L. Noggle
Senior Member Technical Staff
E-Systems, Inc. - CAPA
10530 Rosehaven Street
Suite 200
Fairfax, VA 22030
Tel: (703) 352-0300

Thomas Nolan
Director,
U.S. Army Material Systems Analysis Activity
ATTN: AMXSY-CC
Aberdeen Proving Ground, MD 21005
Tel: (301) 278-6612

Neal J. Plotkin
The MITRE Corporation
MS-W964
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-7167

David S. Porter
Business & Technological Systems
Suite 440
10210 Greenbelt Road
Seabrook, MD 20706
Tel: (301) 794-8800

Edward Preston
Technical Staff
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-7020

Robert J. Ravera
Director,
Information Sciences Research Dept.
Center for Naval Analyses
2000 N. Beauregard Street
Alexandria, VA 22311
Tel: (703) 998-3751

S. Rosenstark
Professor
Electrical Engineering Dept.
New Jersey Institute of Technology
323 Dr. Martin Luther King Blvd.
Newark, NJ 07102
Tel: (201) 596-3525

Nils Sandell, Jr.
President
ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Daniel Serfaty
Graduate Student
Dept. of EE&CS
University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-3261

Robert A. Shade
Senior Research Supervisor
HAZELTINE Corporation
Greenlawn, NY 11740
Tel: (516) 261-7000

Mark J. Shensa
Code 632
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2349

J. Randolph Simpson
Scientific Officer
Code 411MA
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217
Tel: (202) 696-4324

Douglas B. Smith
Professor
Department of EE&CS
University of Michigan
Ann Arbor, MI 48109

James G. Smith
Research Associate
Code Air-340R
Naval Air Systems Center
Washington, D.C. 20361
Tel: (202) 692-7414

Cdr. Joseph Stewart
Naval Postgraduate School
Code 74
Monterey, CA 93943
Tel: (408) 646-2772

Walter R. Stromquist
Senior Associate
Daniel H. Wagner, Associates
Station Square One
Paoli, PA 19301
Tel: (215) 644-3400

Therese Swift
USACAC and Ft. Leavenworth
(ATZL-CAC-A)
Ft. Leavenworth, KS 66027-5300
Tel: (913) 684-4876

Harold Szu
Research Physicist
Naval Research Laboratory
Code 5709
4555 Overlook Avenue, S.W.
Washington, D.C. 20375-5000
Tel: (202) 767-1493

George Tasoulis
S&NWSC 6112 - NC #1
CPG - Room 684
Washington, D.C. 20363
Tel: (202) 692-9207

John Tavantzis
Professor
New Jersey Institute of Technology
Newark, NJ 07102
Tel: (201) 596-3499

Debora M. Teasdale
Scientist
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2083

Charles W. Therrien
Professor
Electrical & Computer Eng.
Code 6eTi
Naval Postgraduate School
Monterey, CA 93943
Tel: (408) 646-2082

Martin A. Tolcott
Decision Science Consortium, Inc.
7700 Leesburg Pike
Falls Church, VA 22043
Tel: (703) 790-0510

Andre M. van Tilborg
Research Scientist
Computer Science Dept.
Carnegie-Mellon University
Pittsburgh, PA 15213
Tel: (412) 578-3801

W. S. Vaughan, Jr.
Acting Division Head
Psychological Sciences
Office of Naval Research
Code 442EP
800 N. Quincy Street
Arlington, VA 22217-5000

Scott Weingaertner
Graduate Student
Massachusetts Institute of Technology
Room 35-415/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-2794

Elbert J. Wells
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2083

Paul Wiley
Graduate Student
Massachusetts Institute of Technology
Room 35-401/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-3449

Leah Wong
Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7196

Ronald Edward Wright
Chief Engineer
New Ventures Dept.
FERRANTI Computer Systems, Ltd.
Western Road
Bracknell
Berkshire RG12 1RA
UNITED KINGDOM
Tel: Code + 0344-483232

Kepi Wu
Program Manager
Naval Space & Warfare Command
2511 Jefferson Davis Hwy.
Crystal City, VA 22150
Tel: (202) 692-9189

8th MIT/ONR WORKSHOP ON C^3 SYSTEMS

JUNE 24 TO JUNE 28, 1985

PROGRAM

(AS OF JUNE 12, 1985)

Sponsored by

LABORATORY FOR INFORMATION AND DECISION SYSTEMS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

and

MATHEMATICS PROGRAM
OFFICE OF NAVAL RESEARCH
UNDER CONTRACT ONR/N00014-77-C-0532

and in cooperation with

IEEE CONTROL SYSTEMS SOCIETY
Technical Committee on C^3

MONDAY MORNING, JUNE 24, 1985

SESSION 1: TOWARDS A C^3 THEORY ROOM: 10-250

Chairman: Michael Athans, LIDS/MIT

8:00 - 3:00 P.M. **REGISTRATION** (Outside Room 10-250)

9:00 - 9:15 A.M. **WELCOME AND INTRODUCTION**

M. Athans, LIDS/MIT
J. R. Simpson, ONR

9:15 - 9:45 A.M. **NEW ALGORITHMS FOR THE SOLUTION OF MARKOV DECISION PROBLEMS AND DYNAMIC TEAM PROBLEMS**

D.A. Castanon, M. Kastner, D. Bertsekas, ALPHATECH, Inc.

9:45 - 10:15 A.M. **EFFECTIVENESS ANALYSIS OF EVOLVING SYSTEMS**

J. G. Karam, ICF, Inc., and A. H. Levis, LIDS/MIT

10:15 - 10:45 A.M. BREAK

10:45 - 11:15 A.M. **DELAYS IN DISTRIBUTED DECISIONMAKING ORGANIZATIONS**

V. Jin and A. H. Levis, LIDS/MIT

11:15 - 11:45 A.M. **PERFORMANCE EVALUATION OF FREE-CHOICE STOCHASTIC TIMED PETRI NETS**

P. Wiley and R. R. Tenney, LIDS/MIT

12:00 - 1:15 P.M. LUNCH - Sala de Puerto Rico, Student Center

MONDAY AFTERNOON, JUNE 24, 1985

SESSION 2: DISTRIBUTED DECISION MAKING I ROOM: 10-250

Chairman: J. R. Simpson, ONR

1:30 - 2:00 P.M. **NORMATIVE SOLUTION TO A TEAM DETECTION PROBLEM**

J. Papastavrou and M. Athans, LIDS/MIT

2:00 - 2:30 P.M. **ON THE ANALYSIS AND DESIGN OF HUMAN INFORMATION PROCESSING ORGANIZATIONS**

K. L. Boettcher and R. R. Tenney, LIDS/MIT

2:30 - 3:00 P.M. **DISTRIBUTED DECISIONMAKING WITH CONSTRAINED DECISIONMAKERS - A CASE STUDY**

K. L. Boettcher and R. R. Tenney, LIDS/MIT

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **AN EXPERIMENTAL PLAN FOR STUDYING DISTRIBUTED TACTICAL DECISIONMAKING**

D. Serfaty and D. L. Kleinman, University of Connecticut

4:00 - 4:30 P.M. **OPTIMAL TASK ALLOCATION OF TWO SERVICE STATIONS WITH MULTIPLE CLASSES OF IMPATIENT TASKS**

Z. J. Wu, P. B. Luh and S. C. Chang, University of Connecticut

4:00 - 5:00 P.M. **ISSUES IN PROVIDING MANUAL BACK-UP FOR AUTOMATED COMMAND AND CONTROL SYSTEMS**

G. A. Clapp, NOSC

TUESDAY MORNING, JUNE 25, 1985

SESSION 3: C^3 ISSUES I ROOM: 10-250

Chairman: A. H. Levis, LIDS/MIT

8:30 - 3:00 P.M. **REGISTRATION**

9:00 - 10:00 A.M. **BATTLE FORCE SYSTEM ENGINEERING**

Lt. Col. M. Langston and R. Gray,
Naval Sea Systems Command

10:00 - 10:30 A.M. BREAK

10:30 - 11:00 A.M. **SURVIVABILITY OPTIONS FOR FUTURE TACTICAL AIR CONTROL SYSTEMS**

P. L. Noggle, E-Systems, Inc.

11:00 - 11:30 A.M. **BROADCAST COMMUNICATION POLICIES FOR DISTRIBUTED AEROACOUSTIC TRACKING**

J. R. Delaney, MIT Lincoln Lab.
R. R. Tenney, ALPHATECH, Inc.

11:30 - 12:00 P.M. **THE EFFECTS OF EMERGING TACTICAL SYSTEMS ON COMMAND AND CONTROL ORGANIZATIONS - A MARINE CORPS PERSPECTIVE**

Lt. Col. R. J. Coulter, USMC

12:00 - 1:15 P.M. LUNCH - Sala de Puerto Rico, Student Center

TUESDAY AFTERNOON, JUNE 25, 1985

SESSION 4: ARMY COMMAND AND CONTROL ROOM: 10-105 (Bush)

Chairman: Israel Mayk, CECOM

1:30 - 2:00 P.M. **DIVISION COMMANDER'S CRITICAL INFORMATION REQUIREMENTS**

Cpt. G. C. Harris, Combined Arms Center

2:00 - 2:30 P.M. **SIMULATING THE COMMAND, CONTROL AND SUBORDINATE SYSTEMS (CCS2)**

L. D. Godfrey and Maj. R. M. Curasi, ATZL-CAC-A

2:30 - 3:00 P.M. **C^3 SYSTEMS MODELLING**

S. Rosenstark and J. Frank, New Jersey Institute of Technology

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **MARKOVIAN MODELING OF CANONICAL C^3 SYSTEMS COMPONENTS**

I. Rubin, IRI Corp., and I. Mayk, CECOM

4:00 - 4:30 P.M. **A MODEL FOR ONE-DIMENSIONAL IDENTIFICATION/COUNTER DYNAMICS**

A. U. Meyer and D. Blackmore, New Jersey Institute of Technology
I. Mayk, CECOM

4:30 - 5:00 P.M. **A STOCHASTIC MODEL OF LANCHESTER'S EQUATION**

J. Tavantzis, S. Rosenstark and J. Frank, New Jersey Institute of Technology

6:00 - 8:00 P.M. **RECEPTION** - Howard Johnson Hotel

TUESDAY AFTERNOON, JUNE 25, 1985

SESSION 5: SURVEILLANCE ROOM: 10-250

Chairman: R. R. Tenney, ALPHATECH, Inc.

1:30 - 2:00 P.M. **MAXIMUM LIKELIHOOD DETECTION/ESTIMATION OF JUMPS IN DYNAMIC SYSTEMS**

R. C. Morgan, M. S. Asher, P. A. Kolmus, D. Porter, J. L. Cantor, W. S. Levine
Business and Technological Systems, Inc.

2:00 - 2:30 P.M. **TWO SENSOR CORRELATION PERFORMANCE MODEL**

M. Kovacich, COMPTEK Research Inc.

2:30 - 3:00 P.M. **CORRELATION AND TRACKING IN OCEAN SURVEILLANCE**

B. Belkin and W. Stromquist, Daniel H. Wagner Associates

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **COMBINATION OF EVIDENCE IN C^3 SYSTEMS**

I. R. Goodman, NOSC

4:00 - 4:30 P.M. **OVERVIEW OF TARGET MOTION ANALYSIS**

M. Desai, C. S. Draper Laboratory

4:30 - 5:00 P.M. **INTEGRATED PLANNING FOR TRACKING MISSIONS**

M. McClure, C. S. Draper Laboratory

6:00 - 8:00 P.M. **RECEPTION** - Howard Johnson Hotel

WEDNESDAY MORNING, JUNE 26, 1985

SESSION 6: C^3 SYSTEM EVALUATION ROOM: 10-250

Chairman: A. H. Levis, LIDS/MIT

8:30 - 3:00 P.M. **REGISTRATION**

9:00 - 9:45 A.M. **ASSESSMENT OF TIMELINESS IN COMMAND AND CONTROL**

P. H. Cothier, Service Technique Des Telecommunications
et Equipments Aeronautiques

A. H. Levis, LIDS/MIT

9:45 - 10:15 A.M. **ARTILLERY CONTROL ENVIRONMENT**

S. C. Chamberlain and V. A Kaste, U.S. Army Ballistic Research Lab.

10:15 - 10:45 A.M. BREAK

10:45 - 11:15 A.M. **ANALYSIS OF THE FIRE SUPPORT TEAM FORCE DEVELOPMENT TESTING AND EXPERIMENTATION II**

V. A. Kaste and S. M. Chamberlain, U.S. Army Ballistic Research Lab.

11:15 - 11:45 A.M. **ASSESSING THE ORGANIZATIONAL RESPONSIBILITY OF HEADQUARTERS UNDER LEVELS OF STRESS**

Cdr. J. F. Stewart and M. Sovereign, Naval Postgraduate School

12:00 - 1:15 P.M. LUNCH - Sala de Puerto Rico, Student Center

WEDNESDAY AFTERNOON, JUNE 26, 1985

SESSION 7: NETWORKS ROOM: 10-105 (Bush)

Chairman:

1:30 - 2:00 P.M. **LIMITED SENSING ALGORITHMS FOR COMMUNICATION NETWORKS USING CARRIER-SENSE MULTIPLE ACCESS CHANNELS**

M. Merakos and P. Papantoni-Kazakos, University of Connecticut

2:00 - 2:30 P.M. **EVALUATING BATTLE-DAMAGE-TOLERANT SHIPBOARD NETWORK DESIGNS**

M. Vineberg, NOSC

2:30 - 3:00 P.M. **FAST HEURISTICS FOR REAL-TIME CLUSTERING, ROUTING AND SCHEDULING**

L. Platzman, Georgia Institute of Technology

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **PROBABILISTIC LOGIC MODELS OF MILITARY CONFLICT AND C^3I DEVELOPMENT**

R. Anthony, The MITRE Corp.

4:00 - 4:30 P.M. **ON AN ELEMENTARY VERSION OF THE MODERN THEORY OF MARTINGALES WITH AN APPLICATION TO POINT PROCESSES AND NETWORKS OF QUEUES**

G. R. Andersen, U.S. Army Ballistic Research Lab.

4:30 - 5:00 P.M. **MULTI MEDIA COMMUNICATION NETWORKS IN COMMAND AND CONTROL**

R. Shade, Hazeltine

WEDNESDAY AFTERNNON, JUNE 26, 1985

SESSION 8: DECISION AIDS IN C^2 ROOM: 10-250

Chairman: J. Malecki, ONR

1:30 - 2:00 P.M. **INFORMATION STORAGE AND ACCESS IN DECISIONMAKING ORGANIZATIONS**

G. J. Bejjani, Morgan Stanley and Co., Inc., and A. H. Levis, LIDS/MIT

2:00 - 2:30 P.M. **ASSOCIATION ALGORITHMS FOR DISTRIBUTED ESM SYSTEMS**

D. A. Castanon, T. G. Allen, and M. P. Merriman, ALPHATECH, Inc.

2:30 - 3:00 P.M. **RESPONSE DEPICTION FOR AUTOMATED COMBAT SYSTEMS**

G. E. Frishkorn and J. G. Gersh, The Johns Hopkins University

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **THE ATTRIBUTES OF EXPERT SYSTEMS IN DECISION AND PLANNING SUPPORT SYSTEMS**

R. E. Wright, FERRANTI Computer Systems

4:00 - 4:30 P.M. **COMPARISON OF A SIMPLE SEARCH TECHNIQUE WITH AN ASSOCIATIVE MATRIX MEMORY SEARCH**

H. Szu, S. Gardner, L. Sweet, Naval Research Lab.

4:30 - 5:00 P.M. **APPLICATION OF PERSONALIZED DECISION AIDING CONCEPTS TO AN ADAPTIVE AID FOR ROUTE PLANNING**

M. S. Cohen, Decision Science Consortium, Inc.

THURSDAY MORNING, JUNE 27, 1985

SESSION 9: C^3 ISSUES II ROOM: 10-250

Chairman: M. Athans, LIDS/MIT

8:30 - 11:30 A.M. **REGISTRATION**

9:00 - 9:45 A.M. **WHY IS INTEGRATED SPACE DEFENSE DIFFERENT?**

J. J. Shaw and M. Athans, ALPHATECH, Inc.

9:45 - 10:30 A.M. **OVERVIEW OF ISSUES IN BATTLE MANAGEMENT C^3I FOR STRATEGIC DEFENSE**

N. Sandell, Jr., ALPHATECH, Inc.

10:30 - 11:00 A.M. BREAK

11:00 - 11:30 A.M. **ON COMBINING UNCERTAIN MESSAGES USING BELIEF FUNCTIONS**

A. P. Dempster and A. Kong, Harvard University

11:30 - 12:00 P.M. **A THEORY OF INFORMATION PRESENTATION FOR COMMAND AND CONTROL**

D. Noble, Engineering Research Associates, Inc.

12:00 - 1:30 P.M. LUNCH - Sala de Puerto Rico, Student Center

THURSDAY AFTERNNON, JUNE 27, 1985

SESSION 10: DISTRIBUTED DECISION MAKING II ROOM: 10-250

Chairman: A. H.Levis, LIDS/MIT

1:30 - 2:00 P.M. **BATTLE STAFF BEHAVIOR UNDER DIFFERENT TEMPOS**

J. L. Lawson, Jr., Naval Postgraduate School

2:00 - 2:30 P.M. **ON NASH AND STACKELBERG GAMES CONCEPTS; ARE THEY USEFUL IN TEAM DECISION PROBLEMS?**

M. Athans, LIDS/MIT

2:30 - 3:00 P.M. **FURTHER RESULTS ON THE CONSENSUS PROBLEM AMONG DISTRIBUTED DECISIONMAKERS**

D. A. Castanon, D. Teneketzis, ALPHATECH, Inc.

3:00 - 3:30 P.M. BREAK

3:30 - 4:30 P.M. **PLENARY SESSION**
Round Table Discussion

FRIDAY's program is being organized by ONR.

These sessions will be classified and held at

The MITRE Corp., Bedford, MA. Details are provided separately.

AUTHOR INDEX

	Page
Anthony, R.	1
Asher, M. S.	173
Bejjani, G. J.	101
Belkin, B.	185
Boettcher, K. L.	69
	75
Carducci, O. M.	153
Castanon, D. A.	95
	145
Chamberlain, S. M.	25
	31
Chang, S. C.	95
Cothier, P. H.	39
Delaney, J. R.	195
Dempster, A. P.	167
Diamond, M. D.	153
Frank, J.	121
	135
Frishkorn, G. E.	111
Gardner, S.	141
Gersh, J. G.	111
Goodman, I. R.	161
Jin, V.	87
Karam, J. G.	53
Kaste, V. A.	25
	31
Kazakos-Papantoni, P.	189
Kastner, M.P.	145
Kleinman, D. L.	65
Kong, A.	167
Kovacich, M.	179
Levine, W. S.	173
Levis, A. H.	39
	53
	87
	101
Luh, P. B.	95
Mayk, I.	15
	127
Merakos, M.	189
Meyer, A. U.	127
Morgan, R. C.	173
Noble, D.	115
Porter, D.	173
Rosenstark, S.	121
	135
Rubin, I.	15
Serfaty, D.	65
Sovereign, M.	49
Stewart, J. F.	49
Stromquist, W.	185
Sweet, L.	141
Szu, H.	141
Tavantzis, J.	135
Tenney, R. R.	69
	75
	81
	195
Wiley, R. P.	81
Wright, R. E.	7
Wu, Z. J.	95