June 1983 LIDS-P-1300

INFORMATION THEORETIC MODELS OF MEMORY IN HUMAN DECISIONMAKING MODELS

Susan A. Hall
RCA Astroelectronics, Princeton, NJ

Alexander H. Levis
Massachusetts Institute of Technology, Cambridge, MA

ABSTRACT

Models of memory and information storage useful in the modeling and analysis of decisionmaking with bounded rationality are discussed. An information theoretic model of permanent memory is presented for describing the accessing of stored information by the algorithms within the human decisionmaker model. It is then applied to the study of the performance - workload characteristic of a decisionmaker performing a dual task.

Submitted to: **Colloquium #34** on Man-Machine Systems

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support by the U. S. Air Force Office of Scientific Research under contract no. AFOSR-80-0029.

Please address correspondence to: Dr. A. H. Levis
35-410/LIDS
M.I.T.
Cambridge, MA 02139
USA

1. INTRODUCTION

Information theory was first developed as an application in communication theory [1]. But, as Khinchin [2] showed, it is also a valid mathematical theory in its own right, and it is useful for applications in many disciplines, including the modeling of simple human decisionmaking processes [3] and the analysis of information-processing systems. Laming [4] observed, however, that the human decisionmaker does not act like a memoryless communications channel, and, in fact, the purpose of most decisionmaking systems is quite other than to reproduce faithfully at the output what was given to the system as input. In accordance with this observation, a two-stage information theoretic model of the decisionmaking process has been developed [5], [6], [7] which includes internal variables and algorithms between the input and the output. However, the model is memoryless; that is, it is unable to recognize any statistical dependence that might exist in the input or access internal or external data bases. This is a simplifying but very limiting assumption: certainly many organizations receive a variety of inputs related to the same situation, and many of these are statistically dependent on one another. Sen and Drenick [8] recognized the need for adding memory to models of decisionmaking systems. They modeled the human decisionmaker as an adaptive channel, i.e., a channel whose input may depend on present and past inputs. With this addition of memory, they achieved results which, in some experimental situations, reflect observed behavior. However, they have made no attempt to model explicitly the various types of memory that may be found in a decisionmaking system.

Several models of memory have been developed [9]. **Buffer storage** allows the decisionmaking system to process sequential statistically dependent inputs simultaneously. **Permanent memory** provides decisionmaking systems with information which is not updated as a result of internal processing, while **temporary memory** allows for the updating of the stored information. All three have been analyzed [9]; however, emphasis is placed in this paper on a model of permanent memory and its use in the analysis of a model of a human decisionmaker faced with a dual task.

In complex situations when a limited amount of time is available for the decisionmaking process, the decisionmaker may be better modeled as being boundedly rational, i.e., constrained in his abilities to formulate actions and foresee consequences. Rather than always being able to make the optimal decision, a decisionmaker with bounded rationality may satisfice, that is, may seek to satisfy some set of minimal criteria in making a decision [10].

The model of a decisionmaker with bounded rationality [5], [6], [7], shown in Figure 1, consists of two stages: the situation assessment (SA) and the response selection (RS) ones.

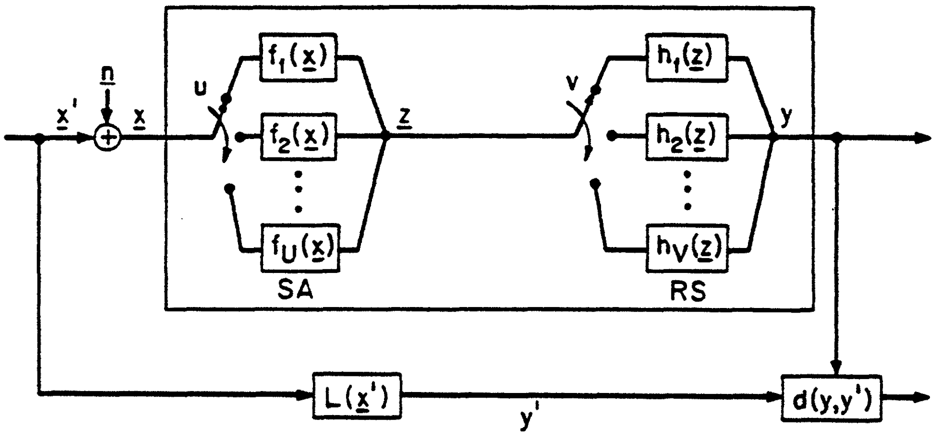

Figure 1. Model of decisionmaking process with performance evaluation mechanism

In the SA stage, one of U algorithms is selected via the variable u to evaluate the input and "hypothesize about its origin." The output of the SA stage, z, could be an estimate of the actual signal given the observed input, or some other statistic of the input, or even the entire input itself. The variable z is then given to the response selection stage (RS), and one of V algorithms is chosen via the variable v, to process the evaluated input into an appropriate response. Both sets of algorithms are assumed to be deterministic, so that, given an input x, and the values of u and v, the output y may be exactly determined. Bounded rationality is modeled by requiring that the total rate of activity of the system, where total rate of activity is a well-defined information theoretic quantity, be less than some maximum value, which is specific to a given decisionmaking system.

The performance of the decisionmaker is evaluated as shown in Figure 1. The actual input $\underline{x}'$ is corrupted by noise, $\underline{n}$, so that the system receives $\underline{x} = \underline{x}' + \underline{n}$, a noisy version of the input. This noise could range from representing actual interference with a message sent to the decisionmaker along standard communications channels, to representing the decisionmaker's inability to observe perfectly, or obtain perfect information pertaining to his environment. The mapping $L(x')$ yields y', which is defined as the ideal response to the actual input $\underline{x}'$; then y' is compared to the output of the system, y. The performance measure of the system is J, the expectation of $d(y,y')$, where the latter is the cost of deciding y when y' is the desired response. In the context of this model, then, a satisficing decisionmaker must choose a decision strategy, i.e., two probability distributions on u and v, that result in $J \leq \bar{J}$, where $\bar{J}$ is the maximum cost that can be tolerated.

The modeling is developed in the analytic context of N-dimensional information theory. There are two quantities of primary interest. The first of these is **entropy**: given a variable x, which is an element of the alphabet X, and occurs with probability p(x), the entropy of x, H(x), is defined to be

$$H(x) \equiv - \sum_{x} p(x) \quad \log p(x) \tag{1.1}$$

and is measured in bits when the base of the logarithm is two. Entropy is also known as the average information or uncertainty in x, where information does not refer to the content of the variable x, but rather to the average amount by which knowledge of x reduces the uncertainty about it. The other quantity of interest is **average mutual information** or **transmission**: given two variables x and y, elements of the alphabets X and Y, and given p(x), p(y), and p(x|y) (the conditional probability of x, given the value of y), the transmission between x and y, T(x:y) is defined to be

$$T(x:y) \equiv H(x) - H_y(x) \tag{1.2}$$

where

$$H_y(x) \equiv - \sum_y p(y) \sum_x p(x|y) \log p(x|y) \tag{1.3}$$

is the conditional uncertainty in the variable x, given full knowledge of the value of the variable y.

McGill [11] extended this basic two-variable input-output theory to N dimensions by extending Eq. (1.2):

$$T(x_1:x_2:\ldots:x_N) \equiv \sum_{i=1}^{N} H(x_i) - H(x_1,x_2,\ldots x_N) \tag{1.4}$$

For the modeling of memory and of sequential inputs which are dependent on each other, the use of the entropy rate, $\bar{H}(x)$, which describes the average entropy of x per unit time, is appropriate:

$$\bar{H}(x) \equiv \lim_{m \to \infty} \frac{1}{m} H[x(t), x(t+1),\ldots,x(t+m-1)] \tag{1.5}$$

Transmission rates, $\bar{T}(x:y)$, are defined exactly like transmission, but using entropy rates in the definition rather than entropies.

Conant's Partition Law of Information Rates (PLIR) [12] is defined for a system with N - 1 internal variables, w_1 through w_{N-1}, and an output variable, y, also called w_N. The PLIR states

$$\sum_{i=1}^{N} \bar{H}(w_i) = \bar{T}(x:y) + \bar{T}_y(x:w_1,w_2,\ldots,w_{N-1}) + \bar{T}(w_1:w_2:\ldots:w_{N-1}:y)$$

$$+ \bar{H}_x(w_1,w_2,\ldots,w_{N-1},y) \qquad (1.6)$$

and is easily derived using information theoretic identities. The left-hand side of Eq. (1.6) refers to the total rate of activity of the system, also designated $\bar{G}$. Each of the quantities on the right-hand side has its own interpretation. The first term, $\bar{T}(x:y)$, is called the **throughput rate** of the system and is designated $\bar{G}_t$. It measures the amount by which the output of the system is related to the input.

$$\bar{T}_y(x:w_1,w_2,\ldots,w_{N-1}) = \bar{T}(x:w_1,w_2,\ldots,w_{N-1},y) - \bar{T}(x:y) \qquad (1.7)$$

is called the **blockage rate** of the system and designated $\bar{G}_b$. Blockage may be thought of as the amount of information in the input to the system that is not included in the output. The third term, $\bar{T}(w_1:w_2:\ldots:w_{N-1}:y)$, is called the **coordination rate** of the system and designated $\bar{G}_c$. It is the N-dimensional transmission of the system; i.e., the amount by which all of the internal variables in the system constrain each other. The last term, $\bar{H}_x(w_1,w_2,\ldots,w_{N-1},y)$ designated $\bar{G}_n$ represents the uncertainty that remains in the system variables when the input is completely known. This noise should not be construed to be necessarily undesirable as it is in communications theory: it may also be thought of as **internally-generated information**, information supplied by the system to supplement the input and facilitate the decisionmaking process. The PLIR may be abbreviated:

$$\bar{G} = \bar{G}_t + \bar{G}_b + \bar{G}_c + \bar{G}_n \qquad (1.8)$$

The bounded rationality constraint is expressed by postulating the existence of a maximum rate of information-processing, or a maximum rate of total activity, $\bar{G}_{max}$, at which a given decisionmaking system can operate without overload. Note that the addition of memory to the decisionmaking model increases the total number of variables in the system and may, therefore, restrict the strategies that may be used to those with lower activity or workload. However, executing a task with memory may result in a better performance than that achievable in a system without memory.

In the next section, a model of memory is presented. In the third section, the model is used to study the performance - workload characteristics of a decisionmaker assigned with the execution of two concurrent tasks — the dual task problem.

2.0 PERMANENT AND TEMPORARY MEMORY

2.1 Introduction

Memory is assumed to consist of both permanent and temporary stores of information which may be drawn upon by the algorithms in the situation assessment and the response selection stages during the decisionmaking process. Permanent memory is defined here to contain values which are constant; that is, they may not be revised or appended by the algorithms that access them. Temporary storage contains values which may be revised by the algorithms; for example, a discrete Kalman filter algorithm would include temporary storage of the best estimate of the present state of the process, to be used in the next iteration of the algorithm. Temporary memory has the effect of adding memory to the algorithms themselves; with temporary memory available, the algorithms can remember values from one iteration to the text. The division of memory into permanent and temporary bears a strong resemblance to the division of memory that is made in the cognitive sciences, into long-term and short-term memory [13].

A third type of memory, called sensory memory, is also hypothesized by psychologists. Information from the environment is stored in sensory memory before it undergoes any processing; sensory memory might therefore be compared to a buffer storage model. The latter allows the simultaneous processing of sequential statistically dependent inputs. Several different models have been developed [9] that depend on the class of inputs that the system receives. Shift register buffers provide the storage rule necessary to process input from a general Markov source. Fixed-length string buffers are a suitable model for the type of storage found in machines. Variable-length string buffers are appropriate models for some types of human sensory memory. Shift register buffers are simple, but add a great deal of activity to the system and result in redundant processing. Fixed-length buffers do not suffer from these deficiencies, but introduce a substantial delay which is proportional to the length of the string. Variable-length string buffers have smaller average delay than fixed-length ones, but increase the overall activity because of their relative complexity.

The model of permanent memory presented in this paper is similar to long-term memory, in that information is stored indefinitely and is accessible by information processing mechanisms. It is different in that new information is being added continuously to long-term memory; the permanent memory model in this paper provides no mechanism for this addition. Second, information may be lost from long-term memory; this permanent memory model does not have a forgetting mechanism. These differences are noted to indicate that, although similarities exist between the model of memory presented here and that found in the cognitive sciences, permanent memory is not intended to be a model of long-term memory per se.

Permanent memory may be accessed by both the situation assessment and the response selection stages. However, in this paper, consideration will be limited to the situation assessment stage. The relationships derived are the same as they would be for the response selection stage, since the two halves of the decisionmaking process are structurally identical.

It is quite possible that a decisionmaking system may contain both buffer storage and permanent and temporary units. However, in order to simplify the presentation, the assumption is made that the decisionmaking system contains no buffer storage. This situation is relaxed easily [9].

The general memory unit applicable to both permanent and temporary memory, is shown in Figure 2. It consist of M variables, d_1 through d_M, as well as an input M-vector, $\underline{D}_I$, and an output M-vector $\underline{D}_0$. Note that because permanent memory may not be revised, its model will not contain the input vector $\underline{D}_I$.

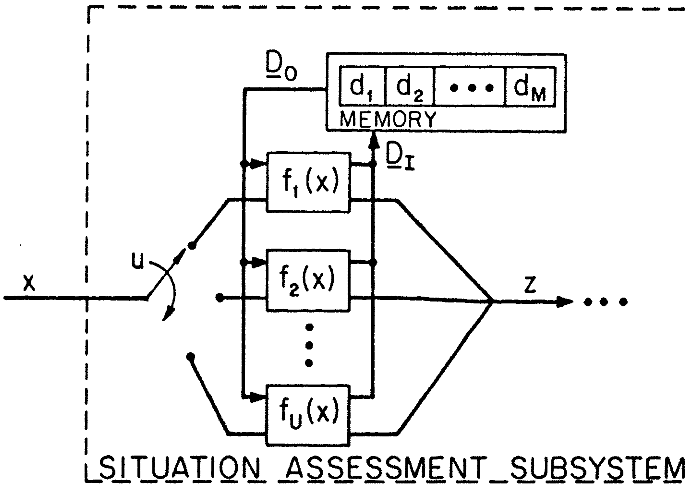

Figure 2. Model of SA subsystem with memory

2.2 Permanent Memory

It might seem at first that the addition of permanent memory to a decisionmaking system might have not effect at all on the total information theoretic rate of activity of the system; if the values of d_k for k=1,2,...,M do not change over time, then

$$\bar{H}(d_k) = 0 \qquad\qquad k = 1,2,\ldots,M \qquad\qquad (2.1)$$

Since total activity is just the sum of the entropies of the individual variables in the system, it appears that the addition of M deterministic variables to a system should have no effect on its total activity. However, the problem is actually more complex. In order to demonstrate the types of changes that occur when permanent memory is added to the model, a particularly simple example will be analyzed.

Let the permanent memory unit consist of one variable, d_1, which may be accessed by one SA algorithm, f_1, as shown in Figure 3.

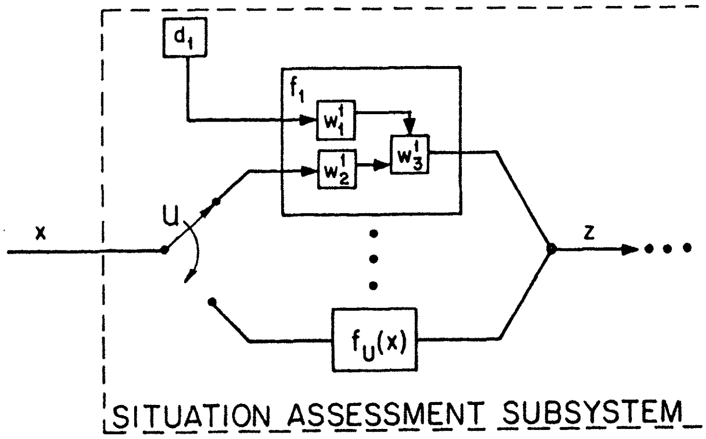

Figure 3. Example of SA subsystem with permanent memory

Algorithm f_1 provides the average value of the two components of a vector input. Whenever a specific algorithm is accessed by the decisionmaking system, the variables of that algorithm are defined to be active, and take values according to some probability distribution which is a function of the input. When that algorithm is not accessed, its variables are defined to be inactive; i.e, they assume some fixed value, say ϕ, which they may not assume when they are active.

Now consider algorithm f_1, which provides the mean value of the two components of a vector input. There are two similar ways of implementing this algorithm. The first does not access permanent memory, although there is some implicit memory in the algorithm itself:

$$w_1 = 2 \; ; \qquad w_2 = x_1 + x_2$$

$$w_3 = w_2/w_1 \; ; \quad z = w_3 \tag{2.2}$$

The second does access permanent memory:

$$d_1 = 2 \; ; \qquad \text{defined outside the algorithm } f_1$$

$$w_1 = d_1 \;; \qquad w_2 = x_1 + x_2$$

$$w_3 = w_2/w_1 \;; \quad z = w_3 \tag{2.3}$$

In the second example, the variables w_1, w_2, w_3 are inactive when the algorithm is not accessed. In the first example, however, w_1 must retain the value of 2 throughout, since no means have been provided to reinitialize its value each time the algorithm is accessed. It is now possible to compare the levels of activity of the system with the permanent memory unit and that without. First consider realization A. The throughput, blockage, and noise rates of the SA subsystem are given by [9]:

$$\begin{aligned} \bar{G}_t &= \bar{H}(z) - \bar{H}_{\underline{x}}(z) \\ \bar{G}_b &= \bar{T}_z(\underline{x}: w_1, \ldots, w_{\alpha U}) \\ \bar{G}_n &= \bar{H}(u) \end{aligned} \tag{2.4}$$

With inputs arriving once every second, the coordination rate is found as follows:

$$\bar{G}_c = \sum_{i=1}^{U} \sum_{j=1}^{\alpha_i} \bar{H}(w_j^i) + \bar{H}(u) + \bar{H}(z) - \bar{H}(u,W,z) \tag{2.5}$$

Here, w_j^i represents the j-th variable of algorithm i; and W represents the entire set of w_j^i in the SA subsystem. Finally, α_i is the number of internal variables of algorithm i which are active or inactive according to the value of u. Equation (2.5) reduces to:

$$\bar{G}_c = \sum_{i=1}^{U} \alpha_i \, \mathbf{H}[p(u)=i)] + \sum_{i=1}^{U} \sum_{j=1}^{\alpha_i} \bar{H}_u(w_j^i) - \bar{H}_u(W) + \bar{H}(z) \tag{2.6}$$

The symbol **H** denotes the binary entropy of its argument, given by:

$$H(p) = - p \log_2 p - (1-p)\log_2(1-p), \; 0 \leq p \leq 1 \qquad (2.7)$$

The same quantities may be calculated for realization B. The rates of throughput, blockage and noise are not effected by the small structural difference in algorithm f_{1B}; the rate of coordination does change. Consider Eq. (2.6): α_{1B} is now equal to 3, because w_1 is now active when u=1 and inactive otherwise. Therefore, the first term of Eq. (2.6) is increased by the amount $H[p(u=1)]$. The second and third terms remain the same, even though there is now some uncertainty associated with the value of w_1. Knowledge of the value of u resolves that uncertainty, so that

$$\bar{H}_u(w_1) = 0 \qquad (2.8)$$

Similarly, $\bar{H}_u(W)$ is unchanged. Only the structure of the algorithm has been changed, so the output remains the same, and the last term of Eq. (2.6) is unchanged. Therefore, the addition of one unit of permanent memory to the SA subsystem provides a total increase in activity of $H[p(u=1)]$. In general, if algorithm i directly accesses β_i values from permanent memory, and no other changes are made in the algorithms, then the incremental activity of the system, $\Delta\bar{G}$, is given by

$$\Delta\bar{G} = \sum_{i=1}^{U} \beta_i \, H[p(u=i)] \qquad (2.9)$$

3.0 THE DUAL-TASK PROBLEM

3.1 Introduction

It has been observed that if a person must execute two tasks by switching between them, his level of performance may be different than when he is allowed to confine himself to one task [13], [14], even if the arrival rate for individual tasks is the same for both cases. If there is sone synergy between the two tasks--that is, if the two tasks are related and executing one actually helps the execution of the other--then performance may improve. If, on the other hand, the two tasks are dissimilar or simply do not reinforce each other, performance may decline from what it was in the single-task case. It is this latter phenomenon that will be explored in this section.

There are numerous possible ways in which the dual-task problem might be modeled. For example, if the two tasks to be performed are assumed to be so different from each other that they demand different sets of algorithms, then a pre-processor may be required for the system. The pre-processor determines which type of task each input represents and then allows access to a set of decisionmaking algorithms appropriate to that task. Of course, the activity of the pre-processor increases overall system activity and may, therefore, lower performance. On the other hand, if the two tasks to be performed are assumed to be similar but non-synergistic, they may be able to use the same basic sets of algorithms, as long as these algorithms are adaptable to each task through two different sets of parameter values stored in permanent memory. Notice that there is an implicit need for a pre-processor in this problem, since the algorithms must have some way of knowing which type of input has been received in order to determine which set of values stored in memory to access. Overall activity is increased in this formulation as well by the necessity of switching between sets of information. An example of this second problem is a hotel switchboard operator who has to process both incoming and outgoing calls; although the tasks require the same basic action, they differ with respect to the information required to execute the tasks. The second problem is addressed in this paper; the first one will be presented at a later time.

In order to simplify this problem, several assumptions will be made. First, to circumvent the need for a pre-processor, it is assumed that there are two separate inputs to the system, x_A and x_B, which are members of disjoint alphabets, X_A and X_B. Only one of these inputs is active at any given time: if x_A is active, task A must be performed, and if x_B is active, task B must be performed. Inputs arrive at the system once every second, and there is a known probability TD (representing the task division) that x_A will be active at any given time.

If inputs are not synergistic, then they are assumed to be statistically independent. If x is generated independently every τ seconds, then

$$\bar{H}(x) = \frac{1}{\tau} H(x) \tag{3.1}$$

Therefore, in the results which follow, entropies rather than entropy rates will be used. Activities are denoted by G in place of $\bar{G}$ for activity rates. The units for G are bits per symbol (as opposed to bits per second for $\bar{G}$). Note that for the problem with synergy between tasks, the assumption of dependence between sequential inputs would be appropriate; a buffer storage model would be added and activity rates would be used in the analysis.

The basic model for the problem is shown in Figure 4. The variable u, which acts independently of the input x, controls which of two situation assessment (SA) algorithms, f_1 and f_2, will be accessed. The decision strategy for a system such as this may then be defined by the probability δ that u is equal to 1. For simplicity, it is assumed that the purpose of both tasks is merely to assess the situation, so that z is the output (no RS stage). The variables s_1 and s_2 are represented as switches external to the algorithms only so that their function may be highlighted. Figure 4 does not explicitly depict the mechanism by which s_1 and s_2 take their values, but only that they are dependent on the value of u and on the values of x_A and x_B.

Specifically, they take values as follows:

$$s_i = \begin{cases} A & \text{if } u = i, \quad x_A \neq \phi, \quad x_B = \phi \\ B & \text{if } u = i, \quad x_A = \phi, \quad x_B \neq \phi \\ \phi & \text{if } u \neq i \end{cases} \qquad i = 1,2 \tag{3.2}$$

In addition to s_1 and s_2, algorithms f_1 and f_2 contain α_1 and α_2 internal variables. Finally, D_A and D_B are the two sets of information or data needed by the algorithms to process input from X_A and X_B, respectively. It is assumed that both algorithms use all of the information in D_A when performing task A and all of D_B for task B.

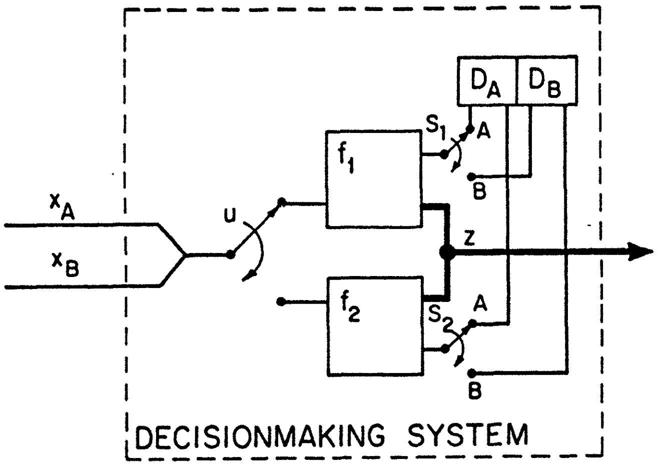

Figure 4. Model of decisionmaker with permanent memory for Dual-Task problem.

3.2 Information Theoretic Analysis

In order to measure if a change in performance level occurs between the single-task and dual-task situations, a performance index is required. The index J that is used is the probability of error. In terms of the quantities defined in section 1 and depicted in Figure 1 (with the output of the system now equal to z),

$$d(z,z') = \begin{cases} 1 & \text{if } z \neq z' \\ 0 & \text{if } z = z' \end{cases} \tag{3.3}$$

and therefore J, the expectation of d(z,z') is

$$J = \sum_{z} p(z)d(z,z') = \text{prob}(z \neq z') \tag{3.4}$$

Because two distinct tasks are being performed, J_A is defined as the probability of error in executing a type A task; and J_B as the probability of error in a type B task. More precisely,

$$J_i = \text{prob}(z=z'|x \in X_i) \qquad i = A,B \tag{3.5}$$

Note that these quantities are independent of the task division TD, the probability that $x \in X_A$, but will be dependent in general on the decision strategy δ. In fact, if it is known how the system performs when pure strategies are employed (either u is 1 with probability 1, and algorithm f_1 is always used, or u is 2 and algorithm f_2 is always used), the performance of the system under the mixed strategy δ (algorithm f_1 is used with probability δ) is simply a convex combination of the performances using pure strategies [5], i.e.,

$$J_i(\delta) = \delta(J_i|u\equiv 1) + (1-\delta)(J_i|u\equiv 2) \qquad i = A,B; \quad 0 \lesssim \delta \lesssim 1 \tag{3.6}$$

With this definition of task performance, it is also possible to define an overall performance index for the system:

$$J(\delta) = (TD)J_A(\delta) + (1-TD)J_B(\delta) \tag{3.7}$$

If errors on one task are more detrimential than errors on the other, then weighting coefficients may be introduced on the right-hand side of Eq. (3.7).

The activity of the system will change both as a function of the decision strategy δ, and as a function of the task division TD. In fact, G, the total activity of the system, is convex both in δ (with a fixed TD) and in TD (with δ fixed). The convexity of G in δ has already been shown [5]; the convexity of G in TD will be demonstrated.

Assume that only task A is being performed. Note that under this assumption, the need for variables s_1 and s_2 disappears; the algorithms may be directly connected to data base D_A. In this case, the levels of activity for a decisionmaker with two SA algorithms f_1 and f_2, containing α_1 and α_2 internal variables, respectively, and a decision strategy δ, are given by:

$$G_t + G_b = H(x)$$
$$G_n + H(u) = H(\delta)$$
$$G_c = (\alpha_1 + \alpha_2)H(\delta) + H(z) + \sum_{i=1}^{2} p(u=i)g_c^i \qquad (3.8)$$

The quantity H is the entropy of a binary variable; $g_c{}^1$ and $g_c{}^2$ are defined to be the internal coordinations of algorithms f_1 and f_2, respectively, where internal coordination is defined as

$$g_c^i = \frac{1}{\text{prob}(u=i)} \quad [\sum_{j=1}^{\alpha_i} H(w_j^i|u=i) - H(W^i|u=i)] \qquad (3.9)$$

and W^i represents the set of all of the variables of algorithm i. The total activity of the system is then the sum of the quantities given in Eq. (3.8):

$$G = H(x) + H(z) + (\alpha_1 + \alpha_2 + 1)H(\delta) + \sum_{i=1}^{2} p(u=i)g_c^i \qquad (3.10)$$

Note that all of the above quantities are conditional on task A being performed: e.g., $g_c{}^i$ could be written $(g_c{}^i|x\varepsilon X_A)$. It has been shown [5] that G is convex in the decision strategy, i.e.:

$$G(\delta) \geq (\delta)(G|u=1) + (1-\delta)(G|u=2) \qquad (3.11)$$

Therefore, using Eqs. (3.6) and (3.11), G may be found parametrically as a function of J for the single-task problem, as shown in Figure 5.

The dual-task problem requires the variables s_1 and s_2 to be included in the model. It is still the case that

$$\begin{aligned} G_t + G_b &= H(x) \\ G_n &= H(u) = H(\delta) \end{aligned} \tag{3.12}$$

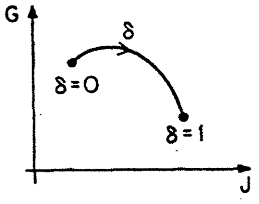

Figure 5. Representation of G vs. J for binary variation of pure strategies

Let d_k represent a single variable of the permanent memory unit, let D represent all the permanent memory (both D_A and D_B), and let W represent all of the internal variables of both algorithms f_1 and f_2; then the coordination of the system performing two tasks is given by:

$$\begin{aligned} G_c = &\sum_{i=1}^{2} \sum_{j=1}^{\alpha_i} H(w_j^i) + H(s_1) + H(s_2) + H(u) + H(z) \\ &+ \sum_k H(d_k) - H(W, s_1, s_2, u, z, D) \end{aligned} \tag{3.13}$$

After much manipulation, Eq. (3.13) may be reduced to

$$\begin{aligned} G_c = &(\alpha_1 + \alpha_2 + 2)H(\delta) + \sum_{i=1}^{2} p(u=i)[(TD)(g_c^i | x\varepsilon X_A) + (1-TD)(g_c^i | x\varepsilon X_B] \\ &+ \sum_{i=1}^{2} \sum_{j=1}^{\alpha_i} T(w_j^i : s_i | u=i) + H(z) \end{aligned} \tag{3.14}$$

and the total activity for the system performing two tasks is given by:

$$\begin{aligned} G = {} & H(x) + H(z) + (\alpha_1+\alpha_2+3)H(\delta) \\ & + \sum_{i=1}^{2} p(u=i)\ [(TD)(g_c^i|x\varepsilon X_A) + (1-TD)\ (g_c^i|x\varepsilon X_B)] \\ & + \sum_{i=1}^{2}\ \sum_{j=1}^{\alpha_i} T(w_j^i : s_i|u=i) \end{aligned} \tag{3.15}$$

There are now two additional $H(\delta)$ terms; these are due to the presence of the two additional system variables, s_1 and s_2. The internal coordination term is now a convex combination of the internal coordinations found when only task A or B is performed. Finally, the last term of Eq. (3.15), which does not even appear in Eq. (3.10):

$$T(w_j^i : s_i|u=i) = H(s_i|u=i) - H_{w_j^i}(s_i|u=i) \tag{3.16}$$

This may be interpreted as the amount of information transmitted between s_i and w_j^i, given that algorithm i is being used; i.e., it is the extent to which variable w_j^i reflects which task is being performed. Since

$$H(s_i|u-i) = p(u=i)H(TD) \tag{3.17}$$

then

$$0 \lesssim T(w_j^i : s_i|u=i) \lesssim p(u=i)H(TD) \tag{3.18}$$

It will now be shown that for a fixed value of δ, $0 \lesssim \delta \lesssim 1$, G is convex in the task division, i.e.,

$$G(TD) \gtrsim (TD)(G|x\varepsilon X_A) + (1-TD)(G|x\varepsilon X_B) \qquad 0 \lesssim TD \lesssim 1 \tag{3.19}$$

The right-hand side (RHS) of (3.19) may be found using (3.10):

$$\mathrm{RHS} = (\mathrm{TD})[H^{A}(x) + H^{A}(z) + \alpha_1+\alpha_2+1)H(\delta) + \sum_{i=1}^{2} p(u=i)(g_c^i|x\varepsilon X_A)]$$

$$+ (1-\mathrm{TD})[H^{B}(x) + H^{B}(z) + \alpha_1+\alpha_2+1)H(\delta) + \sum_{i=1}^{2} p(u=i)(g_c^i|x\varepsilon X_B)] \tag{3.20}$$

Here, $H^j(x)$ and $H^j(z)$ are the entropies of x and z which occur when only a single task is executed. The probability distributions for x and z in the dual-task case are a convex combination of those for the single-task cases,

$$\begin{aligned} p(x) &= (\mathrm{TD})p(x|x\varepsilon X_A) + (1-\mathrm{TD})p(x|x\varepsilon X_B) \\ p(z) &= (\mathrm{TD})p(z|z\varepsilon Z_A) + (1-\mathrm{TD})p(z|z\varepsilon Z_B) \end{aligned} \qquad 0 \le \mathrm{TD} \le 1 \tag{3.21}$$

When a probability distribution is the convex combination of two others, as in Eq. (3.21), then [16]:

$$\begin{aligned} H(x) &\ge (\mathrm{TD})H^{A}(x) + (1-\mathrm{TD})H^{B}(x) \\ H(z) &\ge (\mathrm{TD})H^{A}(z) + (1-\mathrm{TD})H^{B}(z) \end{aligned} \qquad 0 \le \mathrm{TD} \le 1 \tag{3.22}$$

and it follows that Eq. (3.20) can be written as:

$$\mathrm{RHS} = [(\mathrm{TD})H^{A}(x) + (1-\mathrm{TD})H^{B}(x)] + [(\mathrm{TD})H^{A}(z) + (1-\mathrm{TD})H^{B}(z)]$$

$$+ [(\alpha_1+\alpha_2+1)H(\delta)] + \sum_{i=1}^{2} p(u=i)[\mathrm{TD}(g_c^i|x\varepsilon X_A) + (1-\mathrm{TD})(g_c^i|x\varepsilon X_B)] \tag{3.23}$$

Now compare Eq. (3.23) to Eq. (3.15), using the results of Eq. (3.22), the fact that $H(\delta) \ge 0$, and the fact that transmissions must also be non-negative. It follows that Eq. (3.19) does indeed hold, and H is convex in the task division. In fact, if a mixed strategy δ is being used ($0 < \delta < 1$), or if any of the internal variables of an algorithm in use reflects which task is being performed, i.e.,

$$T(w_j^i : s_i | u=i) > 0 \tag{3.24}$$

then the inequality of Eq. (3.19) will be strict.

3.3 Effect of Task Division on Performance

To see the effects that this result has on performance, consider a particularly simple example. It is assumed that the single-task activity or workload versus peformance curves are identical for task A and task B (this implies that J_A and J_B are the same functions of δ: see Figure 6a). Now consider the evolution of the G versus J_A curve as TD changes from 0 to 1. It is meaningless to define J_A for the single-task case in which task B is always performed (TD = 0), but for very small values of TD, J_A is defined as in Eq. (3.5). To find the G versus J_A curve for TD ≈ 0, consider Eq. (3.15). Since H(TD) ≈ 0 for TD ≈ 0, its last term is small (see Eq. (3.18)). The rest of Eq. (3.15) reduces to Eq. (3.25):

$$G(TD \approx 0) \approx (G | x \varepsilon X_B) + 2H(\delta) \tag{3.25}$$

In other words, the G versus J_A curve will be the same as either single-task curve, with the quantity $2H(\delta)$ added on due to the presence of variables s_1 and s_2 (see Figure 6b). As TD increases, G will continue to increase up to some point (because of its convexity in TD), dependent on the value of the last term of Eq. (3.15) and the values of H(x) and H(z). For TD equal to 0.5, the G versus J_A curve will have the general shape shown in Figure 6c. Finally, G will decrease until TD ≈ 1 and G versus J_A is again as shown in Figure 6b. For a fixed value of δ then, say δ=0.2, the workload versus task division curve will be similar to that shown in Figure 6d. The maximum activity need not occur at TD = 0.5.

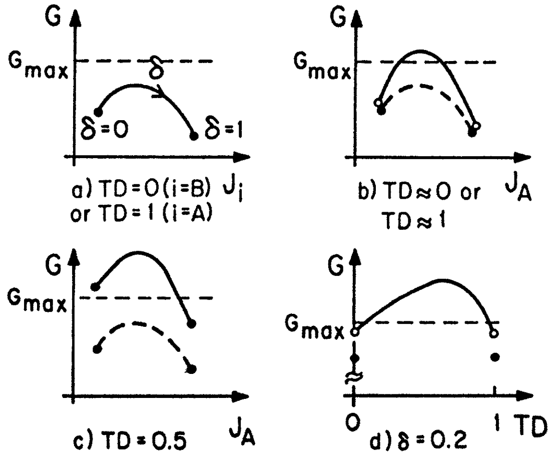

Figure 6. Performance vs. Workload and Task Division

Now consider what happens to performance if the maximum total workload constraint is given by the value marked G_{max} in Figure 6, i.e., the system is required to perform at an activity level $G \leq G_{max}$. In the two single-task cases, the system is unconstrained and may use any strategy $0 \leq \delta \leq 1$. However, for this example, when both tasks arrive with equal probability (TD = 0.5), the set of feasible strategies is greatly reduced, and performance is limited to being very poor. Also, the particular strategy of $\delta = 0.2$ may only be used for task divisions close to 0 or 1 (see Figure 6d).

This simple example illustrates some rather general results. The convexity of G in the task division implies that the rate of activity of the system will be greater in the dual-task case than in at least one of the single tasks. If the workload of the two tasks is very disparate, then the opportunity to switch between a very activity-intensive (high workload) task and a very easy one may actually reduce the workload from what it is in the case that only the difficult task is being performed. When the activity levels for the two single-task cases are comparable, though, as in the preceding example, the workload for the dual-task case is greater than that for either of the single-task cases. This increase in workload arises from three basic sources, which may be seen by an examination of Eq. (3.15).

First, in the dual-task case, the variable x and in most cases also the variable z will have a larger uncertainty associated with them because of their larger alphabets. Second, the dual-task problem requires that the system have some means of switching between the sets of data stored in permanent memory: the variables s_1 and s_2 provide the mechanism but also increase the uncertainty of the system. Third, the rest of the internal variables may, because of access to different values stored in memory, take on a wider range of values when both tasks must be performed than when only one is performed. If the system performing either task alone is operating near its maximum allowable rate, then requiring the system to switch between the two tasks has the effect of both eliminating the more active decision strategies from the feasible set, and, in the case that the more active strategies also result in better performance, lowering the performance of the system.

4.0 CONCLUSION

In order to obtain more realistic models of humans carrying out information processing and decisionmaking tasks, it is necessary that memory, whether internal to the decision process, or external in the form of data bases, be modeled. Three classes of models are described: buffer storage, permanent memory and temporary memory. The modeling of permanent memory has been prsented and illustrated through its use to the analysis of the performance-workload characteristic of a human decisionmaker executing a dual task.

In order to test experimentally these predictions the model for the dual-task problem as defined here, several criteria must be met. First, the two tasks must be similar enough that the same set of algorithms may be used for both tasks; however, they should be independent enough so that execution of one task does not aid in the execution of the other. Second, it should be necessary to switch between tasks, i.e., two different tasks may not be performed simultaneously. Third, individual tasks should arrive at the same rate in the dual-task test as in the single-task test. Finally, this rate of presentation should be near to the bounded rationality constraint of the decisionmaker, since it is hypothesized that it is this constraint that leads to pefrormance degradation.

5.0 REFERENCES

[1] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, University of Illinois, Urbana, 1949.

[2] A. I. Khinchin, Mathematical Foundations of Information Theory, Dover Publications, Inc., New York, 1957.

[3] T. B. Sheridan and W. R. Ferrell, Man-Machine Systems, MIT Press, Cambridge, MA, 1974.

[4] D. R. J. Laming, Information Theory of Choice-Reaction Times, Academic Press, London and New York, 1968.

[5] K. L. Boettcher and A. H. Levis, "Modeling the Interacting Decisionmaker with Bounded Rationality," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No. 3, May/June 1982.

[6] A. H. Levis and K. L. Boettcher, "Decisionmaking Organizations with Acyclical Information Structures," IEEE Transactions on Systems, Man, and Cybernetics, SMC-13, May/June 1983.

[7] A. H. Levis and K. L. Boettcher, "On Modeling Teams of Interacting Decisionmakers with Bounded Rationality," Proc. IFAC/IFIC/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man-Machine Systems, VDI/VDE, Duesseldorf, FRG, September 1982; to appear in Automatica, Vol. 19, No. 6, November 1983.

[8] P. Sen and R. F. Drenick, "Information Processing in Man-Machine Systems," Proc. of the Fourth MIT/ONR Workshop on C^3 Systems, LIDS-R-1159, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, October 1981.

[9] S. A. Hall, "Information Theoretic Models of Storage and Memory," S. M. Thesis, LIDS-TH-1232, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August 1982.

[10] J. G. March and H. A. Simon, Organizations, John Wiley and Sons, New York, 1958.

[11] W. J. McGill, "Multivariate Information Transmission," Psychometrika, Vol. 19, No. 2, June 1954.

[12] R. C. Conant, "Laws of Information Which Govern Systems," IEEE

Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 4, April 1976.

[13] R. L. Solso, Cognitive Psychology, Harcourt, Brace, and Jovanovich, Inc., New York, 1979.

[14] R. T. Kelly, F. L. Greitzer, and R. L. Hershman, "Air Defense: A Computer Game for Research in Human Performance," NPRDC TR 81-15, Navy Personnel Research and Development Center, San Diego, CA, July 1981.

[15] R. T. Kelly, and F. L. Greitzer, "Effects of Track Load on Decision Performance in Simulated Command and Control Operations," NPRDC TR 82-21, Navy Personnel Research and Development Center, San Diego, CA, January 1982.

[16] R. G. Gallager, Information Theory and Reliable Communication, John Wiley and Sons, New York, 1968.