January, 1983 LIDS-P-1269

OPTIMAL SMOOTHING AND ESTIMATION FOR HYBRID STATE PROCESSESS[1]

F. Bruneau[2]

R.R. Tenney[3]

---

1. Support from ONR contract N00014-77-0532C (NR 041-519) is gratefully acknowledged.

2. 106 Rue Charles Lafitte, 92200 Neville, France

3. MIT/LIDS Rm. 35-213, Cambridge, Mass. 02139. Address all inquiries to this author.

ABSTRACT

Consider the estimation and smoothing problem for a hierarchical Markov process. The supremal state evolves autonomously; infemal dynamics and observations may be statistically dependent on the supremal state. This class of processes has more structure than a general Markov process; the implications of this structure are developed here. Of special interest is the case of hybrid systems, where the supremal state is discrete and the infemal dynamics are linear and Gaussian. This structure commonly appears in diverse applications, including failure detection, maneuvering target tracking, and digital communications on analog channels. It is also the structure for which the most useful conclusions can be drawn.

OPTIMAL SMOOTHING AND ESTIMATION FOR HYBRID STATE PROCESSES.

I. INTRODUCTION

The impending availability of very large scale integrated circuits gives us the opportunity to review some of the classical approaches to control and estimation problems, particularly those with a combinatorial structure, and reconsider some of the design tradeoffs inherent therein. This new technology provides the option of efficiently implementing algorithms which are rather profligate in their requirements for multiplies and adds, provided that the algorithm can be decomposed into a number of highly structured loci of computation with relatively loose coupling (in terms of data transfer) between them. Dynamic estimation problems provide a source of such algorithms, particularly when they have a more complex structure than the oft discussed linear-Gaussian case.

The special structure considered here involves a Markov process with state space $\underline{X}$ which can be decomposed into subspaces $\underline{X}_1 \times \underline{X}_2$, and where the dynamics on $\underline{X}_1$ are independent of $\underline{X}_2$, but not vice-versa. The observation space $\underline{Y}$ can be decomposed compatibly. This structure lies at the heart of several important applications, particularly in hybrid systems where $\underline{X}_1$ is discrete (modeling failure modes, maneuver modes, or digital symbols) and $\underline{X}_2$ continuous (modeling system dynamics, target trajectories, or channel dynamics, respectively). These problems are usually dominated by the entire discrete state sequence. Many ad hoc solutions to these types of problems have appeared in the literature [1-3,11], where approximations are required in order to overcome the exponential growth of the set of discrete state trajectories as the time horizon of the problem advances. Now that combinational problems are not necessarily computationally unassailable, it is worthwhile understanding the extent to which these problems can be solved exactly. While the result may still be unimplementable, inclusion of computation-reducing features which do not affect performance (and which do exist) certainly provides a starting point for other modifications.

This paper develops optimal methods for approaching the filtering and smoothing problems for systems with the above structure. The contributions are of two types: specific techniques for reducing the complexity of hybrid system estimation algorithms, and a general structure for approaching this class of problems. The techniques and approach seem quite helpful in designing algorithms for VLSI implementation, but do not entirely solve the problem. As an example will show, the specific techniques developed here may reduce the combinatorial growth of a problem from exponential to linear (in time); this is helpful, but still not practical, and approximations must also be introduced. Thus a prime purpose of this work is to delimit the power of exact techniques, and create a framework for future performance analysis of approximate techniques.

The development begins with a formal problem statement, followed by the derivations of optimal filtering and smoothing techniques in a general setting. These are then specialized to the linear-Gaussian and hybrid linear-Gaussian cases. In the latter the greatest payoff is obtained; an illustration of this concludes the work.

II. PROBLEM STATEMENT

A. Models

Let the state space of a Markov process be $\underline{X} = \underline{X}_1 \times \underline{X}_2$. $\underline{X}_1$ is the state space of the supremal subsystem which evolves autonomously; $\underline{X}_2$ that of the infemal subsystem which is dependent upon the value of the supremal state $x_1(t)$. Formally, we make:[4]

Assumption 1: The state transition probabilities factor as

$$p(x_1(t+1),x_2(t+1)|x_1(t),x_2(t)) = p(x_1(t+1)|x_1(t))p(x_2(t+1)|x_1(t),x_2(t)) \quad \square \qquad (2\text{-}1)$$

The process is observed via the space $\underline{Y} = \underline{Y}_1 \times \underline{Y}_2$, where $\underline{Y}_1$ contains observations of the supremal state only, and $\underline{Y}_2$ of the joint state. Again, make

Assumption 2: The observation probabilities factor as

$$p(y_1(t),\ y_2(t)|x_1(t),x_2(t)) = p(y_1(t)|x_1(t))p(y_2(t)|x_1(t),x_2(t)) \quad \square \qquad (2\text{-}2)$$

Implicit in the above are the usual conditional independence assumptions for a Markov process , so the quantities in (2.1) and (2.2) completely specify the system.

We will be interested in the maximum a posteriori (MAP) estimates of the state (filtering), or entire state trajectory (smoothing), conditioned on an sequence of observations received from the system. Introducing the notation

4. For the general case derivations will be done formally. This is exact when $\underline{X}$ is discrete, and when all invoked distributions exist and are well defined.

$$X_i(t) \in \underline{X}_i^t \qquad\qquad Y_i(t) \in \underline{Y}_i^t \qquad (2\text{-}3)$$

for sequences of states and observations (over a time interval $\tau \in \{1,\ldots,t\}$), the problem is

Assumption 3: Find

a) for the filtering problem, the state $x^*(t)$ which maximizes $p(x(t)|Y(t))$

b) for the smoothing problem, the state trajectory $X^*(t)$ maximizing $p(X(t)|Y(t))$. □

This is the general problem. Two special cases which are of interest are the linear-Gaussian, and the discrete/linear-Gaussian (hybrid) structures. In the former, assume that $\underline{X}_i = \mathbb{R}^{n_i}$, and that the system dynamics are linear with additive white Gaussian driving noise. (2.1) becomes

Assumption 1L: The hierarchical dynamics[5] are:

$$\vec{x}_1(t+1) = \underline{A}_{11}\vec{x}_1(t) + \vec{w}_1(t) \qquad \vec{w}_1(t) \sim N(\vec{0},\underline{Q}_1) \qquad (2\text{-}4)$$

$$\vec{x}_2(t+1) = \underline{A}_{21}\vec{x}_1(t) + \underline{A}_{22}\vec{x}_2(t) + \vec{w}_2(t) \qquad \vec{w}_2(t) \sim N(\vec{0},\underline{Q}_2) \qquad (2\text{-}5)$$

where $\underline{Q}_1$ and $\underline{Q}_2$ are positive definite, and $\vec{w}_1$ and $\vec{w}_2$ are jointly independent and white. □

Similarly, the observations lie in $Y_1 = \mathbb{R}^{m_1}$, and

Assumption 2L: The observation equations are

$$\vec{y}_1(t) = \underline{C}_{11}\vec{x}_1(t) + \vec{v}_1(t) \qquad \vec{v}_1(t) \sim N(\vec{0},\underline{R}_1) \qquad (2\text{-}6)$$

---

5. Extension to time-varying system and noise matrices is straightforward and not considered here for notational clarity. A number of other assumptions may be relaxed; the purpose here is to develop some new structure in the simplest setting possible.

$$\vec{y}_2(t) = \underline{C}_{21}\,\vec{x}_1(t) + \underline{C}_{22}\,\vec{x}_2(t) + \vec{v}_2(t) \qquad \vec{v}_2(t) \sim N(\vec{0},\underline{R}_2) \qquad (2\text{-}7)$$

where $\underline{R}_1$ and $\underline{R}_2$ are positive definite and $\vec{v}_1$ and $\vec{v}_2$ are jointly independent. □

Thus the conditonal distributions for the linear case $p(x_1(t+1)|x_1(t))$, etc. in (2-1) and (2-2) are all multivariable Gaussian densities with means and covariances specified by (2-4) - (2-7).

For hybrid models, a combination of discrete and continuous dynamics exist. The supremal system is discrete, specifying some structural mode, and the infemal is assumed linear - Gaussian, with descriptive matrices dependent upon the value of the supremal state. Thus $\underline{X}_1 = \{x_1^1,\ldots,x_1^{n_1}\}$, $\underline{X}_2 = R^{n_2}$, and

<u>Assumption 1H:</u> The dynamics are specified by

$$p(x_1(t+1)|x_1(t)) \qquad (2.8)$$

$$\vec{x}_2(t+1) = \underline{A}(x_1(t))\,\vec{x}_2(t) + \vec{w}(t) \qquad \vec{w}(t) \sim N(\vec{0},\underline{Q}(x_1(t)))$$

with $\underline{Q}$ positive definite.[6] □ (2.9)

Finally, $Y_1 = \{y_1^1,\ldots,y_1^{m_1}\}$, $Y_2 = \mathbb{R}^{m_2}$, and

<u>Assumption 2H:</u> The observations for a hybrid system are specified by

$$p(y_1(t)|x_1(t)) \qquad (2.10)$$

$$\vec{y}_2(t) = \underline{C}(x_1(t))\,\vec{x}_2(t) + \vec{v}(t) \qquad \vec{v}(t) \sim N(\vec{0},\underline{R}(x_1(t))) \qquad (2.11)$$

with $\underline{R}$ positive definite.[6] □

---

6. Nonzero, x_1- dependent means may be treated by state augmentation and the dependence of $\underline{A}$ or $\underline{C}$ on $x_1(t)$

These are the three classes of models treated in sections III-V, respectively.

B. Applications

It is not appropriate to completely review all applications involving filtering and smoothing of hierarchical or hybrid Markov processes here. However, some sense of the applications to which these techniques may apply serves as useful motivation.

Certain failure detection and identification problems [1] are naturally described by (2.8) - (2.11). $\underline{X}_2$ represents the usual states of a dynamical system operating under feedback control. $\underline{X}_1$ describes various failures which may occur in actuators and internal parts of the system (causing changes in $\underline{A}$ or $\underline{Q}$) or in sensors (appearing in $\underline{C}$ or $\underline{R}$). As long as the causes of failure are unrelated to the dynamic states (e.g. due to stressing operation), the above model applies. Interest typically centers on determining the time and type of failures; estimation of $\vec{x}_2$ itself is often a secondary goal.

Maneuvering and multiobject tracking in clutter [2] are other hybrid estimation problems. $\underline{X}_2$ represents positions and velocities of objects in a region; $\underline{X}_1$ may indicate maneuver modes [4], target identities, detection/ nondetections due to environmental effects [5], or the permutations of sensor returns which are not labeled with the target from which they originated [6] . In the latter case, signature information derived from the sensed waveform would be modeled in $\underline{Y}_1$; position and velocity data by $\underline{Y}_2$.

Finally, certain communication problems exhibit a hybrid structure. $\underline{X}_1$ may represent a digital source (e.g. of a pseudorandom code), and

$\underline{X}_2$ analog channel dynamics. $\underline{X}_1$ might also model the existance of bursty interference which effectively sets $\underline{C}$ to zero, and $\underline{R}$ large, intermittently.

III. THE GENERAL CASE

This section develops the concepts, notation, and basic techniques for optimal filtering and smoothing under assumptions 1-3.

A. Filtering

One might expect that the hierarchical structure of the system would lead to the posterior distribution having some structure such as

$$p(x_1(t),x_2(t)|Y_1(t),Y_2(t)) = p(x_1(t)|Y_1(t))\ p(x_2(t)|x_1(t),Y_2(y)) \tag{3-1}$$

If this were so, then one could design a filter for the supremal system alone, and then one for the infemal system which used the results of the supremal filter in the estimation of x_2.

Unfortunately, this is not the case. One step of the Bayesian estimator is

$$p(x_1(t+1),x_2(t+1)|Y_1(t+1),Y_2(t+1)) = p(y_1(t+1)|x_1(t+1))\ p(y_2(t+1)|x_1(t+1),x_2(t+1)).$$

$$\frac{\sum_{x_1(t)} p(x_1(t+1)|x_1(t)) \sum_{x_2(t)} p(x_2(t+1)|x_1(t),x_2(t))\, p(x_1(t),x_2(t)|Y_1(t)Y_2(t))}{p(y_1(t+1),y_2(t+1)|Y_1(t),Y_2(t))} \tag{3-2}$$

Note that even if the conditional distribution at time t had a separation property such as (3-1), it would be lost both in the propagation of the dynamics of x_2 and in the update with y_2. The intuition behind this becomes clear in an extreme example: set $\underline{X}_1 = \underline{X}_2$ and the infemal dynamics so that $x_2(t) = x_1(t)$. Then not only does an observation y_1 provide direct information on x_1, but also about x_2.

Therefore the filtering solution exhibits no special structure in this case.

B. Smoothing: Compact

We will consider two approaches to the smoothing problem, one the usual optimal algorithm, and the other an expanded version which better permits exploitation of the hierarchical structure at a cost of increased computation. This section treats the former; section C the latter.

The suitability of the hierarchical structure to the smoothing problem is suggested by the fact that

$$\max_{X_1,X_2} p(X_1,X_2|Y_1,Y_2) = \frac{1}{p(Y_1,Y_2)} \max_{X_1} \{p(Y_1|X_1)p(X_1) \cdot \max_{X_2} \{p(Y_2|X_1,X_2)\ p(X_2|X_1)\}\} \tag{3-3}$$

This is a direct result of assumptions 1 and 2, since they imply :

$$p(X_1,X_2) = p(X_2|X_1)\ \ p(X_1) \tag{3-4}$$

$$p(Y_1,Y_2|X_1,X_2) = p(Y_1|X_1)\ p(Y_2|X_1,X_2) \tag{3-5}$$

Note that it is not necessary to compute $p(Y_1,Y_2)$ at all: (3-3) suggests an algorithm by which the best X_2, is found for each X_1; and then the best X_1 is found. Unfortunately, $X_1(t)$ and $X_2(t)$ are elements of rather large sets.

However, this has not yet considered the Markov structure of the problem, which is essential to recursive smoothing techniques. Consider the smoothing solution on $\underline{X}_1 \times \underline{X}_2$; we will be interested in determining if (3-3) affects its structure.

<u>Definition</u>: A <u>survivor function</u> [7] $s(x(t)|Y(t))$ is defined by

$$s(x(t)|Y(t)) = \max_{X(t-1)} p(Y(t)|x(t),X(t-1))p(x(t),X(t-1)) \quad (3\text{-}6)$$

□

Technically, s is a function on $\underline{X} \times \underline{Y}^t$; since we will only be interested in evaluating it along a particular realization of the output process $Y(t)$, it is convenient to view it as a function of $x(t)$. It indicates the unnormalized probability of the most likely state trajectory $X(t)$ which terminates in $x(t)$, conditioned on the observation sequence $Y(t)$. Note that the maximizing $X(t-1)$ in (3-6) may not be unique, but one of them may be selected and stored for each $x(t)$. This permits reconstruction of the entire MAP state sequence by finding $x(t)$ which maximizes $s(x(t)|Y(t))$, and then determining the $X(t-1)$ thus associated with it.

The implications of Markov structure are that s is recursively computable.

<u>Lemma 1.</u> $s(x(t)|Y(t))$ may be computed via

$$s(x(t+1)|Y(t)) = \max_{x(t)} p(x(t+1)|x(t))\, s(x(t)|Y(t)) \quad (3\text{-}7)$$

$$s(x(t+1)|Y(t+1)) = p(y(t+1)|x(t+1))\, s(x(t+1)|Y(t)) \quad (3\text{-}8)$$

<u>Proof:</u> Bayes' theorem, interchange of max operations with functions not of the same variable, and the Markov assumptions:

$$p(Y(t)|X(t)) = \prod_{s=1}^{t} p(y(s)|x(s)) \quad (3\text{-}9)$$

$$p(X(t)) = \left[\prod_{s=1}^{t-1} p(x(s+1)|x(s)) \right] p(x(1)) \qquad \square \tag{3-10}$$

If s is replaced with $-\ell n(s)$, a monotonic operation, and the resulting function is minimized, the Viterbi algorithm [7] emerges.

Computationally, the Viterbi algorithm is relatively simple, requiring only $O(N_1N_2)$ operations per time step (for discrete $\underline{X}$). Memory for storing the preceeding trajectory associated with each $x(t)$ is the dominant factor in its implementation. As in the filtering problem, however, the hierarchical structure of the Markov process does nothing, in general, to simplify the algorithm further. Again, the case where $x_1(t) = x_2(t)$ for all t generates an $s(x|Y)$ which is diagonal on $\underline{X}_1 \times \underline{X}_2$, and demonstrates the lack of decomposition.

C. Smoothing: Expanded

An algorithm for the smoothing problem can be constructed which does exploit the hierarchical structure, but at a great increase in computational complexity. As such, it is not useful for general problems of the class considered here, but it will be the key to the structure of the hybrid smoothing problem.

Definition: A conditional survivor function $s(x_2|X_1,Y_2)$ is defined as

$$s(x_2(t)|X_1(t),Y_2(t)) = \max_{X_2(t-1)} p(Y_2(t)|X_1(t),X_2(t)) \; p(X_2(t)|X_1(t)) \qquad \square \tag{3-11}$$

This function is an intermediary in the solution of the smoothing problem, as the second maximization in (3-3) can be rewritten as

$$\max_{X_2(t)} \{p(Y_2(t)|X_1(t),X_2(t))p(X_2(t)|X_1(t))\} = \max_{x_2(t)} s(x_2(t)|X_1(t),Y_2(t)) \tag{3-12}$$

These equations suggest an algorithm whereby $s(x_2|X_1,Y_2)$ is computed, based only on Y_2, for each X_1. The result may be summarized in the function

$$r(X_1(t)|Y_2(t)) \triangleq \max_{x_2(t)} s(x_2(t)|X_1(t),Y_2(t)) \tag{3-13}$$

The outer maximization in (3-3) is then over the product of $p(X_1|Y_1)\ p(X_1)$, which is computable just from the structure of the sumpremal system, and $r(X_1|Y_2)$, derived from the infemal structure only.

This algorithm does capitalize on the hierarchical structure, but leaves two questions to be answered. First, can the $s(x_2|X_1,Y_2)$ be computed recursively? Second, is there some recursive structure which can be exploited in the outer maximization, over X_1, without reducing the solution to a Viterbi algorithm on $\underline{X}_1 \times \underline{X}_2$? The answer to the latter is particularly critical, as the size of $\underline{X}_1^t$ grows exponentially with time.

The answer to both questions is yes. Consider the computation of $s(x_2|X_1,Y_2)$ first.

<u>Lemma 2</u>: $s(x_2|X_1,Y_2)$ may be computed as

a) predict:

$$s(x_2(t+1)|X_1(t),Y_2(t)) = \max_{x_2(t)} p(x_2(t+1)|x_1(t),x_2(t))\ s(x_2(t)|X_1(t),Y_2(t)) \tag{3-14}$$

b) update:

$$s(x_2(t+1)|X_1(t+1),Y_2(t+1)) = p(y_2(t+1)|x_1(t+1),x_2(t+1))\ s(x_2(t+1)|X_1(t),Y_2(t)) \tag{3-15}$$

Proof: Identical to Lemma 1, with conditioning on X_1. □

The structure of these computations is straightforward. For each supremal trajectory $X_1(t)$, (3-14) and (3-15) implement a Vitebri calculation for the survivor function on x_2. Note that the explicit conditioning on X_1 removes the coupling between the statistics of x_2 and the supremal observations Y_1; X_1 provides a more complete statistical specification of the evolution of x_2 than does Y_1;

Note that $s(x_2|X_1,Y_2)$ and $p(x_2|X_1,Y_2)$ convey very different things. The most likely single trajectory to x_2, consistant with X_1 and Y_2, is captured by s; p gives the aggregate probability of being in x_2. If one state in x_2 can be reached by a number of individually low probability paths, while another can by a single high probability path, the s and p will generally be of quite different character.

The best visualization of this structure is to view X_1^t as a tree, rooted at the time zero at a single point, and with each node representing a state sequence over a period of time. From each node branches lead to all states which may be reached as the system extends that state sequence by one time step. Computation of $s(x_2|X_1,Y_2)$ involves running a Vitebri algorithm along each branch of that tree.

Clearly this becomes cumbersome as t grows large. One can then consider the outer optimization in (3-3) as a means for pruning the tree of X_1 sequences. However, this must be done carefully; it is not appropriate to merely eliminate sequences in X_1 merely because they have low probability at time t. (For example, there may be only one sequence X_1, albeit of low probability, which enables $x_2(t)$ to enter some state x_2^*; if a subsequent value of y_2 indicates that $x_2(t) = x_2^*$ with probability 1, it would be helpful if that X_1 were still under consideration).

In fact, the tree of X_1 sequences may indeed be pruned, in a way which guarantees that the MAP trajectory will never be eliminated yet preserves the hierarchical structure. This first requires:

Lemma 3: $p(Y_1(t)|X_1(t))\ p(X_1(t))$ may be computed recursively:

$$[p(Y_1(t+1)|X_1(t+1))\ p(X_1(t+1)] = \tag{3-16}$$

$$= p(y_1(t+1)|x_1(t+1))p(x_1(t+1)|x_1(t))[p(Y_1(t)|X_1(t))\ p(X_1(t))]$$

Proof: Elementary manipulations and the Markov properties. □

Lemma 3 provides for the computation of the term other than $r(X_1|Y_2)$ - the term which captures the supremal dynamics through $p(x_1(t+1)|x_1(t))$, and the supremal observation through $p(y_1(t+1)|x_1(t+1))$.

One more notion is needed.

Definition: The sources of a state $x_1(t)$ are all trajectories in X_1^t terminating in x_1.

Theorem 1: Let $(X_1^*(t), X_2^*(t))$ be the MAP trajectory for observations $Y_1(t), Y_2(t)$. Let τ be any time preceeding t. If, at time τ

$$p(Y_1(\tau)|\tilde{X}_1(\tau))\ p(\tilde{X}_1(\tau))\ s(x_2(\tau)|\tilde{X}_1(\tau),Y_2(\tau)) \le \max_{\{X_1(\tau)\}} p(Y_1(\tau)|X_1(\tau))\ p(X_1(\tau))\ s(x_2(\tau)|X_1(\tau),Y_2(\tau)) \tag{3-17}$$

for each $x_2(\tau)$, where $\{X_1(\tau)\}$ contains all sources of $\tilde{x}_1(\tau)$ **except** $\tilde{X}_1(\tau)$ itself, then $\tilde{X}_1(\tau)$ will not be a subsequence of $X_1^*(t)$.

Proof: Consider the (compact) smoothing algorithm, and the $s(x_1(\tau),\ x_2(\tau)\,|\,Y_1(\tau),\ Y_2(\tau))$ computed by it. Each $(x_1(\tau),\ x_2(\tau))$ has a sequence $(X_1(\tau),\ X_2(\tau))$ associated with it which constitutes an optimal trajectory estimate through $(x_1(\tau),\ x_2(\tau))$. If $\tilde{X}_1(\tau)$ never appears as the first component of one of these associated sequences, it will not appear as a subsequence of any longer trajectory. $\tilde{X}_1(\tau)$ can only appear in association with states of the form $(\tilde{x}_1(\tau),\ x_2(\tau))$. (3-17) assures that there is no $x_2(\tau)$ for which $\tilde{X}_1(\tau)$ is the most likely source of $\tilde{x}_1(\tau)$, hence $\tilde{X}_1(\tau)$ may be eliminated. $\square$

Theorem 1 establishes a looser requirement for eliminating trajectories than the compact smoothing algorithm. The Viterbi algorithm will eliminate trajectories at each point $(x_1(\tau),\ x_2(\tau))$, leaving only one candidate terminating there. (3-17) suggests eliminating $\tilde{X}_1(\tau)$ only if there is no $x_2(\tau)$ at all with which $\tilde{x}_1(\tau)$ may be paired and which preserves $\tilde{X}_1(\tau)$ as a candidate. An even looser criterion is given by

Corollary 1a: $\tilde{X}_1(\tau)$ will not be a subsequence of the optimal estimate if there exists some $X_1(\tau) \neq \tilde{X}_1(\tau)$, both sources of $x_1(\tau)$, where

$$p(X_1(\tau)\,|\,Y_1(\tau))\ s(x_2\,|\,\tilde{X}_1(\tau),\ Y_2(\tau)) \quad \leq \qquad (3\text{-}18)$$
$$p(x_1(\tau)\,|\,Y_1(\tau)) = s(x_2\,|\,X_1(\tau)\ Y_2(\tau))$$

for every x_2.

Proof: (3-18) implies (3-17). The converse is not true as the maximizing $X_1(\tau)$ in (3-17) may vary with x_2. $\square$

Thus we have established two pruning rules for the x_1 trajectories. Both require functional dominance between two scaled versions of $s(x_2\,|\,X_1, Y_2)$

to hold for a trajectory X_1 to be eliminated. Both are weaker than the optimal pruning rules on $\underline{X}_1 \times \underline{X}_2$ implied by the Viterbi algorithm, as the latter are pointwise dominance relations. Thus the strength of the pruning technique has been sacrificed; this can only be advantageous if either $p(X_1|Y_1)$ or $s(x_2|X_1,Y_2)$ has a particularly convenient form compared to $s(x_1,x_2|Y_1,Y_2)$. We will see that this is the case for hybrid state models.

IV. LINEAR - GAUSSIAN CASE

Before moving to the hybrid case, the relation between linear filtering and smoothing algorithms and the quantities introduced above need to be established. While the linear case exhibits no special solution structures as a result of assumptions 1L and 2L, the development here is necessary for section V.

A. Filtering

The solution to the joint filtering problem in $\underline{X}$ is well known for the linear -Gaussian case: the Kalman filter [8]. The statistics

$$\hat{\vec{x}}(t) = E\{\vec{x}(t)|Y(t)\} \qquad \underline{p}(t) = \text{cov}\{\vec{x}(t)\ Y(t)\} \tag{4-1}$$

may be recursively computed as

$$\hat{\vec{x}}(t) = \underline{A}\,\hat{\vec{x}}(t-1) + \underline{K}(t)(\vec{y}(t) - \underline{C}\,\underline{A}\,\hat{\vec{x}}(t-1)) \tag{4-2}$$

$$\underline{P}(t) = [I - K(t)\ \underline{C}][\underline{A}\ P(t-1)\ \underline{A}^T + \underline{Q}][\underline{I} - \underline{K}(t)\underline{C}]^T + \underline{K}(t)\ \underline{R}\ \underline{K}^T(t) \tag{4-3}$$

$$\underline{K}(t) = \underline{P}(t-1)\ \underline{C}^T\ [\underline{C}\ P(t-1)\ \underline{C}^T + R]^{-1} \tag{4-4}$$

While assumptions 1L and 2L imply a block triangular or diagonal form in $\underline{A}$, $\underline{C}$, $\underline{R}$, and $\underline{Q}$, this is not reflected in the propagation of $\underline{P}(t)$, and hence in the structure of the algorithm. The reason for a lack of separation is (4-4); the update gains are not block triangular as both $y_1(t)$ and $y_2(t)$ convey information about both components of the state, just as in section IIIA. Thus the linear - Gaussian assumption does not allow extra structure to become apparent.

B. Smoothing

We will consider only the compact smoothing problem here, as the set $\underline{X}_1^t$ is an entire $N_1 t$ dimensional vector space which cannot be profitably dealt with on a pointwise basis. Thus we will specialize Lemma 1 to this case.

Theorem 2: Under assumptions 1L and 2L, and with $\underline{A}$, $\underline{C}$, $\underline{Q}$ and $\underline{R}$ the matrices which can be partitioned to provide A_{11}, A_{21}, etc. $s(\vec{x}(t)|Y(t))$ is of the general form

$$s_o(t)\, e^{\frac{1}{2}(\vec{x}(t) - \hat{\vec{x}}(t))^T \underline{P}^{-1}(t)(\vec{x}(t) - \hat{\vec{x}}(t))}$$

with the parameters $\hat{\vec{x}}$ and $\underline{P}$ computable via (4-2) - (4-4) and with $s_o(t)$ given by:

$$s_o(t) = (2\pi)^{-N/2} (2\pi)^{-M/2} \det(\underline{Q})^{-1/2} (\underline{R})^{-1/2} s_o(t-1) \qquad (4\text{-}5)$$

$$e^{\frac{-1}{2}(\vec{y}(t) - \underline{C}\,\underline{A}\,\hat{\vec{x}}(t-1))^T \underline{S}^{-1}(t-1)(\vec{Y}(t) - \underline{C}\,\underline{A}\,\hat{\vec{x}}(t-1))}$$

$$\underline{S}(t) = \underline{C}[\underline{A}\,\underline{P}(t-1)\,\underline{A}^T + Q]\,\underline{C}^T + \underline{R} \qquad (4\text{-}6)$$

Proof: See Appendix A.

Four points are important about those equalities. Foremost is the fact that the survivor function is an exponential quadratic, with mode $\hat{\vec{x}}$ and quadratic coefficients identical to those of the conditional state density computed by the Kalman filter. This gives a convenient double interpretation to $\hat{\vec{x}}$ and $\underline{P}$; they are parameters of the filter solution, or parameters of the survivor function. This coincidence is quite special to the linear case. Secondly, the quadratic shape $\underline{P}$ is data independent; $\hat{\vec{x}}$ depends on the observation trajectory so s_o is also data dependent, unlike the filtering case. Since its behavior is dominated by an exponential quadratic form of the residuals, s_o provides a quantification of the goodness of fit of the trajectory to the actual observations (bigger is better). Finally, this is only half of the smoothing solution; reconstruction of X^* from the mode of $s(x|Y)$ can be done in the usual way [9].

Thus the survivor function for the linear-Gaussian smoothing problem can be parametrized by exactly the same quantities as those computed by the Kalman filter, plus a goodness of fit measure closely related to $p(Y|X^*)$. However, since the Kalman filter does not lead to a separation along the lines of the hierarchical structure of the problem, neither does $s(x|Y)$.

V. HYBRID CASE

Now we turn to the hybrid system case, given by assumptions 1H and 2H. The set of supremal trajectories X is discrete, so they can be viewed as being arranged in a tree as in section III. The conditional survivor function $S(x_2|X_1,Y_2)$ will be that of a particular linear-Gaussian system with time-varying dynamics specified by X_1, so the results of theorem 2 translate to it. Thus the smoothing solution takes the form of a bank of

Kalman filters, one for each X_1, with some supremal logic which prunes elements of X_1 using the tests of theorem 1. While this scheme is dominated by the combinatorial size of X_1, we will see that this same structure dominates both the filtering solution and the straightforward Viterbi algorithm for hybrid systems. Only the expanded smoothing approach of section IIIC allows any practical reduction in the size of X_1 on-line.

A. Filtering

The filtering problem for a hybrid system was first addressed many years ago [10]. The exact solution is found from a decomposition much like (3-3).

$$p(x_1(t),x_2(t)|Y_1(t),Y_2(t)) = \frac{1}{p(Y_1(t),Y_2(t))}\sum_{X_1(t-1)} p(Y_1(t)|X_1(t))p(X_1(t))\cdot p(Y_2(t)|X_1(t))\; p(x_2(t)|X_1(t),Y_2(t)) \quad (5\text{-}1)$$

This expression has two parts. The conditional distribution $p(x_2(t)|X_1(t),\, Y_2(t))$ is Gaussian, since $X_1(t)$ specifies completely the linear-Gaussian dynamics of x_2, decoupling it from Y_1. The remaining terms form a set of weights, so that the resulting conditional distribution is, for each $x_1(t)$, a weighted sum of Gaussian distributions on $x_2(t)$. In general, there are $N_1^{(t-1)}$ components in each weighted sum, each corresponding to one element of $X_1(t-1)$. The only time this size is reduced is if two components have exactly the same conditional mean and covariance, an event that does not happen at all in general.

Unlike the general case, and the linear case, the structure of the optimal state estimator forces one to consider expansions over $X_1(t)$.

This is because the conditional distribution $p(x_2(t)| X_1(t), Y_2(t))$ is conveniently parameterized by its mean and covariance, but sums of such distributions can only be expressed in terms of the parameters of the components. However, the opportunities for reducing the complexity of this expansion are almost nonexistant, and this is the point at which engineering approximations for the sake of implementation are usually made. These approximations generally fall into two categories: pruning , where a term in the expansion is dropped completely because its weight is small relative to others, and merging, where two or more terms in the expansion are replaced by a single "equivalent" term, where "equivalent" is often taken to mean "of equal conditional mean and variance". Criteria for determining candidates for pruning or merging are legion. However, all have some detrimental impact on estimation performance.

B. Smoothing

The smoothing problem has a structure wherein pruning is a natural operation. While the ideal smoother requires a survivor function which has many components to it, each being a weighted Gaussian shape, the combination of components is by a max operator, rather than a sum. Thus some components may in fact be completely dominated by others, and dropped without affecting the selection of the trajectory estimate. This is the idea behind optimal pruning of X_1 trajectories.

Consider (3-3) and (3-12):

$$\max_{X_1,X_2} p(X_1,X_2|Y_1,Y_2 = \frac{1}{p(Y_1,Y_2)} \max_{X_1} \{p(Y_1|X_1)p(X_1) \cdot \max_{x_2} \{s(x_2|X_1,T_2)\}\} \qquad (5-2)$$

From Lemma 2 and theorem 2, $s(x_2(t)|X_1(t),Y_2(t))$ will have a weighted Gaussian shape; hence the outer maximization is over a set of Gaussian shapes weighted by both supremal and infemal components. It is conceivable that some terms in this set may be eliminated by the criterion stated in Theorem 1.

First establish:

<u>Leamma 4</u>: In the hybrid case, $s(x_2|X_1,Y_2)$ may be computed by

a) predict:

$$\hat{\vec{x}}(t+1|t) = \underline{A}(x_1(t))\ \hat{\vec{x}}(t|t) \tag{5-3}$$

$$\underline{P}(t+1|t) = \underline{A}(x_1(t))\ \underline{P}(t|t)\ \underline{A}^T(x_1(t)) + \underline{Q}(x_1(t)) \tag{5-4}$$

$$s_o(t+1|t) = (2\pi)^{-N_2/2}\ \det\ (\underline{Q}(x_1(t)))^{-1/2}\ s_o(t|t) \tag{5-5}$$

b) update

$$\hat{\vec{x}}(t+1|t+1) = \hat{\vec{x}}(t+1|t) + \underline{K}(t+1)[\vec{y}(t+1) - \underline{C}(x_1(t+1))\hat{\vec{x}}(t+1|t)] \tag{5-6}$$

$$\underline{P}(t+1|t+1) = [\underline{I} - \underline{K}(t+1)\underline{C}(x_1(t+1))]\underline{P}(t+1|t)[\underline{I} - \underline{K}(t+1)\underline{C}(x_1(t+1))] + \underline{K}(t+1)\ \underline{R}(x_1(t+1))\quad (t+1)\underline{K}^T(t+1) \tag{5-7}$$

$$s_o(t+1|t+1) = (2\pi)^{-M_2/2}\ \det\ \underline{R}(x_1(t+1))^{-1/2}\ s_o(t+1|t)\ e^{-\frac{1}{2}[\vec{y}(t+1) - \underline{C}(x_1(t+1))\hat{\vec{x}}(t+1|t)]^T\ \underline{S}^{-1}(t+1)\ [\vec{y}(t+1) - \underline{C}(x_1(t+1))\hat{\vec{x}}(t+1|t)]} \tag{5-8}$$

$$\underline{S}(t+1) = \underline{C}(x_1(t+1))\ \underline{P}(t+1|t)\ \underline{C}^T(x_1(t+1)) + \underline{R}(x_1(t+1)) \tag{5-9}$$

where

$$\hat{\vec{x}}(t|\tau) = \hat{\vec{x}}(t|X_1(\tau),\ Y_2(\tau)) \tag{5-10}$$

etc.

<u>Proof:</u> Apply Theorem 2 to the recursion of Lemma 2, conditioning on X_1. □

This structure of $s(x_2|Y_1,Y_2)$ indicates that a strict Viterbi algorithm on X necessarily involves a parametrization which is based on trajectories X_1. Thus the compact smoothing algorithm of section IIIB is no simpler than the expanded structure of IIIC in this hybrid state case.

<u>Definition:</u> A <u>quality function</u> $q(x_2|X_1,Y_1,Y_2)$ is given by

$$q(x_2,X_1|Y_1,Y_2) = p(Y_1|X_1)\ p(X_1)\ s(x_2|X_1,Y_2) \tag{5-11}$$

□

In the hybrid case, $q(x_2X_1|Y_1,Y_2)$ is a scaled Gaussian with mode and quadratic weights given by Lemma 4, and with scale factor

$$q_o(X_1|Y_1,Y_2) = s_o(X_1,Y_2)\ p(Y_1|X_1)\ p(X_1) \tag{5-12}$$

where the latter two terms may be recursively computed via Lemma 2.

The crux of the expanded smoothing algorithm in the hybrid case is:

<u>Theorem 3:</u> A supremal trajectory $\tilde{X}_1(\tau)$ will never be a subsequence of an optimal trajectory estimate $(X_1^*(t),\ X_2^*(t))$, $t \geq \tau$, if there exists another $X_1(\tau) \neq \tilde{X}_1(\tau)$ which is a source of $x_1(\tau)$ and for which

$$q(\vec{x}_2(\tau),X_1(\tau)|Y_1(\tau),Y_2(\tau)) \leq q(\vec{x}_2(\tau),X_1(\tau)|Y_1(\tau),Y_2(\tau)) \tag{5-13}$$

for all values of $\vec{x}_2(\tau)$. This inequality holds iff

$$\tilde{P}^{-1}(\tau|\tau) - P^{-1}(\tau|\tau) \geq 0 \tag{5-14}$$

$$\frac{1}{2}(\tilde{\vec{x}}_2(\tau|\tau) - \vec{x}_2(\tau|\tau))^T [\tilde{P}(\tau|\tau) - P(\tau|\tau)]^{-1} (\tilde{\vec{x}}_2(\tau|\tau) - \vec{x}_2(\tau|\tau)) \geq$$

$$\ell n\ q_o(\tilde{X}_1(\tau)|Y_1(\tau),Y_2(\tau)) - \ell n\ q_o(X_1(\tau)|Y_1(\tau),Y_2(\tau)) \tag{5-15}$$

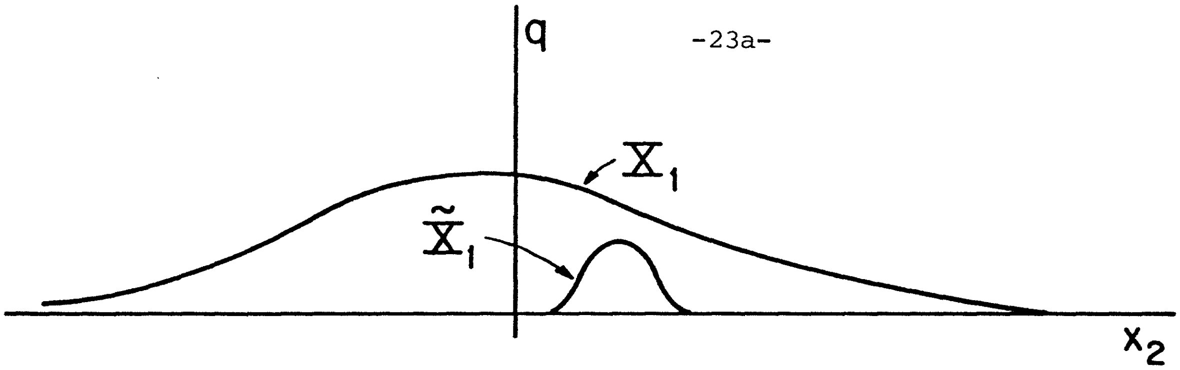
Proof: (5-13) is a restatement of corollary 1a. The equivalence of (5-13) to (5-14) - (5-15) is shown in Appendix B.

□

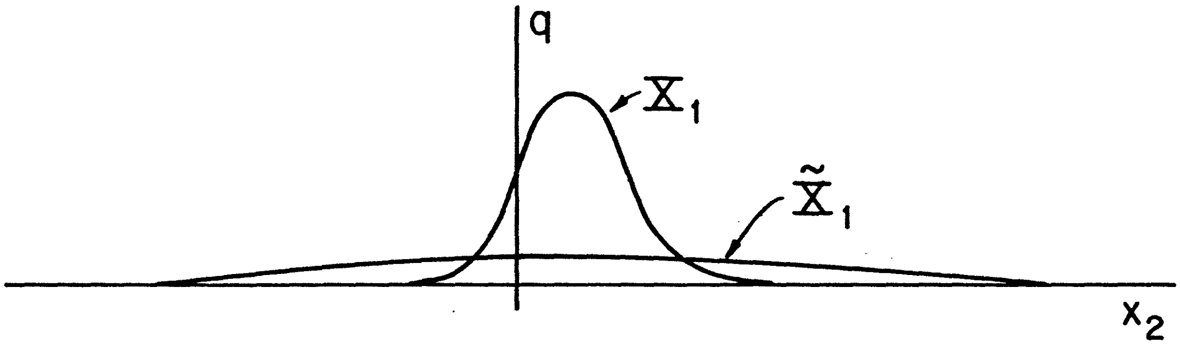
The interpretation of these conditions is interesting. Figure 1a illustrates a case where the $\tilde{q}$ associated with $\tilde{X}_1$ allows it to be eliminated in favor of X_1. (5-14) requires that the conditional Fisher information matrix of a pruned trajectory be greater than that of the one that dominates it; Figure 1b shows that violation of this inequality will lead to $\tilde{q}$ dominating q on the tails of the distributions. Thus trajectories with good conditional information may be eliminated in favor of those with poorer information, but not vice-versa; this imparts a natural conservatism to the pruning. For cases which satisfy (5-14), and for a given $\vec{x}_2$, (5-15) determines an ellipsoidal region wherein $\tilde{\vec{x}}_2$ may lead to pruning $\tilde{X}_2$. Note that (5-14) ensures that the left hand side of (5-15) will always be nonpositive, hence if

$$\frac{q_o(\tilde{X}_1|Y_1,Y_2)}{q_o(X_1|Y_1,Y_2)} > 1 \tag{5-16}$$

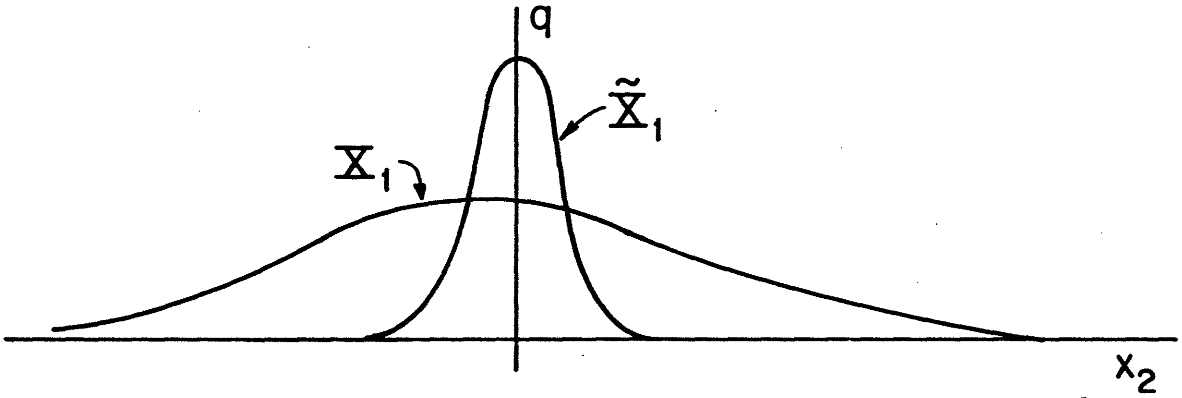
then this ellipsoid will be empty. (Figure 1c). (Note that (5-16) can be interpreted as a likelihood ratio test on the hypotheses that $\tilde{X}_1$ or X_1 is the true trajectory). Even if (5-16) is satisfied, if the offset



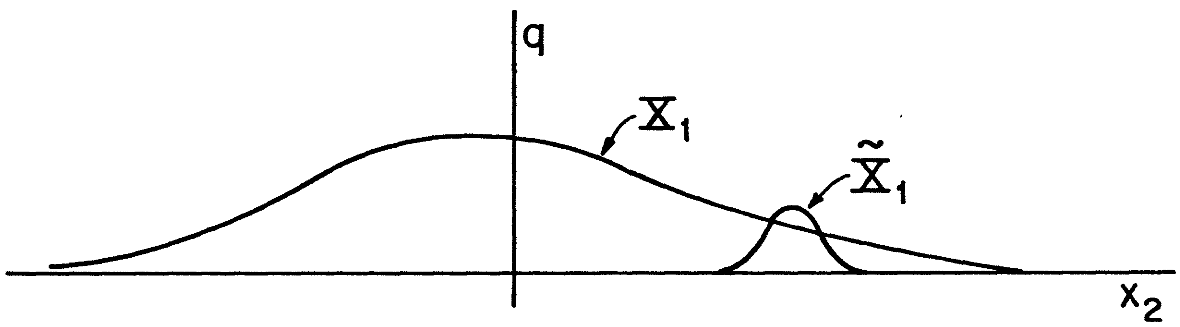
(a) $\tilde{X}_1$ pruned



(b) (5-14) violated



(c) (5-16) violated



(d) (5-15) violated

Figure 1. Comparison of Conditional Qality Functions

between the conditional means is too large, no elimination can take place (Figure 1d).

Since theorem 3 is based on corollary 1a, it is not as complete as possible. There may be cases where X_1 is dominated by neither X_1' nor X_1'' alone, but is dominated by the max of their respective q functions (Figure 2) (provided $x_1(\tau) = x_1'(\tau) = x_1''(\tau)$). While the general inequality of theorem 1 may be applied: prune X_1 if

$$\forall x_2 \quad q(x_2|X_1,Y_1,Y_2) \leq \max_{X_1 \in \text{sources}(x_1)} q(x_2|X_1,Y_1,Y_2) \tag{5-17}$$

the reduction of this test to simple algebraic tests such as (5-14) - (5-15) is rather cumbersome.

Thus the hierarchical structure of the hybrid state dynamics, coupled with the simple parameterization of the conditional survivor function, leads to a hierarchically structured algorithm for the smoothing problem. The infemal level consists of a Kalman filter computing the mode and quadratic spread of the survivor function, and a scale factor calculation based on the Kalman filter residuals and applicable noise covariances. The supremal logic computes conditional probabilities on X_1 based on Y_1, and then prunes away some possibilities based on a Viterbi-like criterion posed in terms of functional, rather than pointwise, dominance.

The smoothing problem is more tractable than the filtering problem because terms in a max-of-weighted- Gaussian functions may be dropped completely, whereas all terms in a sum-of-weighted-Gaussian must be obtained. Thus the smoothing problem admits a simplification in the combinatorial aspects of the hybrid problem which is unavailable in a

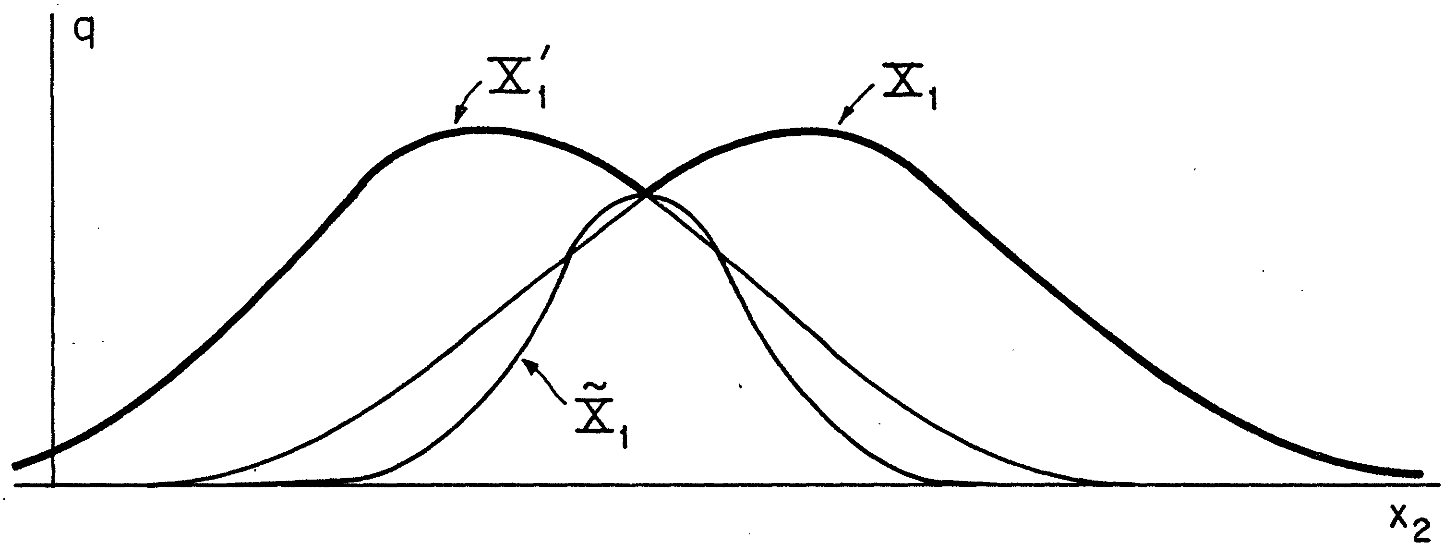


FIGURE 2: Many-on-one Dominance

filtering approach. However, due to the coincidence of the shape of the conditional survivor function $s(x_2|X_1,Y_2)$ and the conditional distribution $p(x_2|X_1,Y_2)$, the Kalman filter statistics computed by the infemal algorithm can also be interpreted as the conditional mean and covariance of x_2 for the trajectory X_1. With this view, the output of the algorithm would be X_1^*, the MAP discrete trajectory, and $p(x_2|X_1^*,Y_2)$, the corresponding continuous state distribution. This type of output may be quite suitable for maneuvering target tracking and communications problems.

Finally, it is important to note that since (5-15) involves means and scale factors, which are data dependent quantities, there is no predetermined order in which the X_1 trajectories are eliminated. Algorithms for which the pruning logic is data dependent are typically quite difficult to analyze; the most important contribution of theorem 3 is the guarantee that the pruning logic stemming from it will never increase the probability of error in the determination of (X_1^*, X_2^*); it is optimal. However, that pruning logic is generally insufficiently powerful to reduce the search for X_1^* to manageable sizes; other techniques are required for an actual implementation.

VI. EXAMPLE

A. Problem

Consider a simple scalar hybrid system, where the plant dynamics are fixed

$$x_2(t+1) = .99\, x_2(t) + w(t) \qquad (6\text{-}1)$$

with $Q = .035$ so that x_2 is normally distributed around zero, with

	x_1	$\hat{\vec{x}}_2$	$\underline{p}$	q
a	0	0.0	1.1	-17.80
b	1	-.428	.204	-16.60
c	0	-5.961	.264	-38.16
d	1	-3.169	.128	-65.59
e	0	-.308	.378	-15.48
f	1	-.439	.15	-14.22
g	0	-.267	.277	-13.09
h	1	-.403	.131	-11.86
i	1	-.343	.096	-9.52

Table 1: Description of Survivors

variance 1.75 in the steady state. The discrete state models a change in the sensor structure; if $x_1 = 1$ (normal)

$$y_2(t) = x_2(t) + v(t) \qquad R(1) = .25 \tag{6-2}$$

and if $x_1 = 0$ (abnormal):

$$y_2(t) = v(t) \qquad R(0) = 18 \tag{6-3}$$

Note that the standard deviation of the prior distribution on y_2 in state 0, $3\sqrt{2}$, is thrice that of the prior on y_2 when $x_1 = 1$ (in the steady state) .

This model may apply in cases where x_2 is a plant and x_1 models a sensor failure; where x_2 is an object and x_1 a random detection process; or where x_2 is a signal and x_1 the presence of interference. In all cases, if the dynamics of x_1 are as shown in Figure 3, and since $p \neq .1$, the failure / inference process is "bursty": it has memory (with an expected holding time of 10/3 time steps in state 0) . This requires an algorithm for smoothing or estimation which exploits the dynamics of x_1 in order to perform well.

Figure 4 shows a typical sample path of the hybrid process described above. Figure 5 shows the corresponding results of using Theorem 3 to prune the tree of possible X_1. Each "x" indicates the time step at which its corresponding trajectory was eliminated. Note that when all descendents of a node are pruned, that node itself is eliminated; the heavy lines indicate the trajectories which are still candidates at time 7. While there are trajectories passing through both $x_1 = 0$ and $x_1 = 1$ at times 1 and 2, all trajectories pass through $x_1 = 0$ at times 3,4, and 5. Refering to figure 4, this indicates that the obvious outliers at t=3 and t=5 have been confirmed as arising from state 0. Note that $y_2(4)$

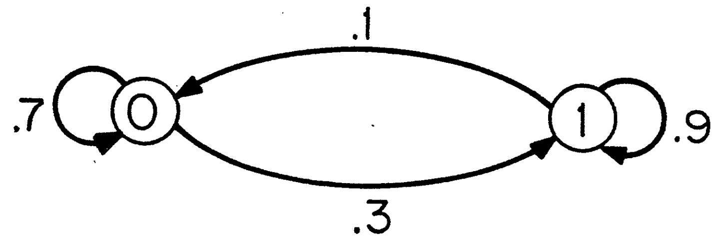


FIGURE 3: x_1 Dynamics

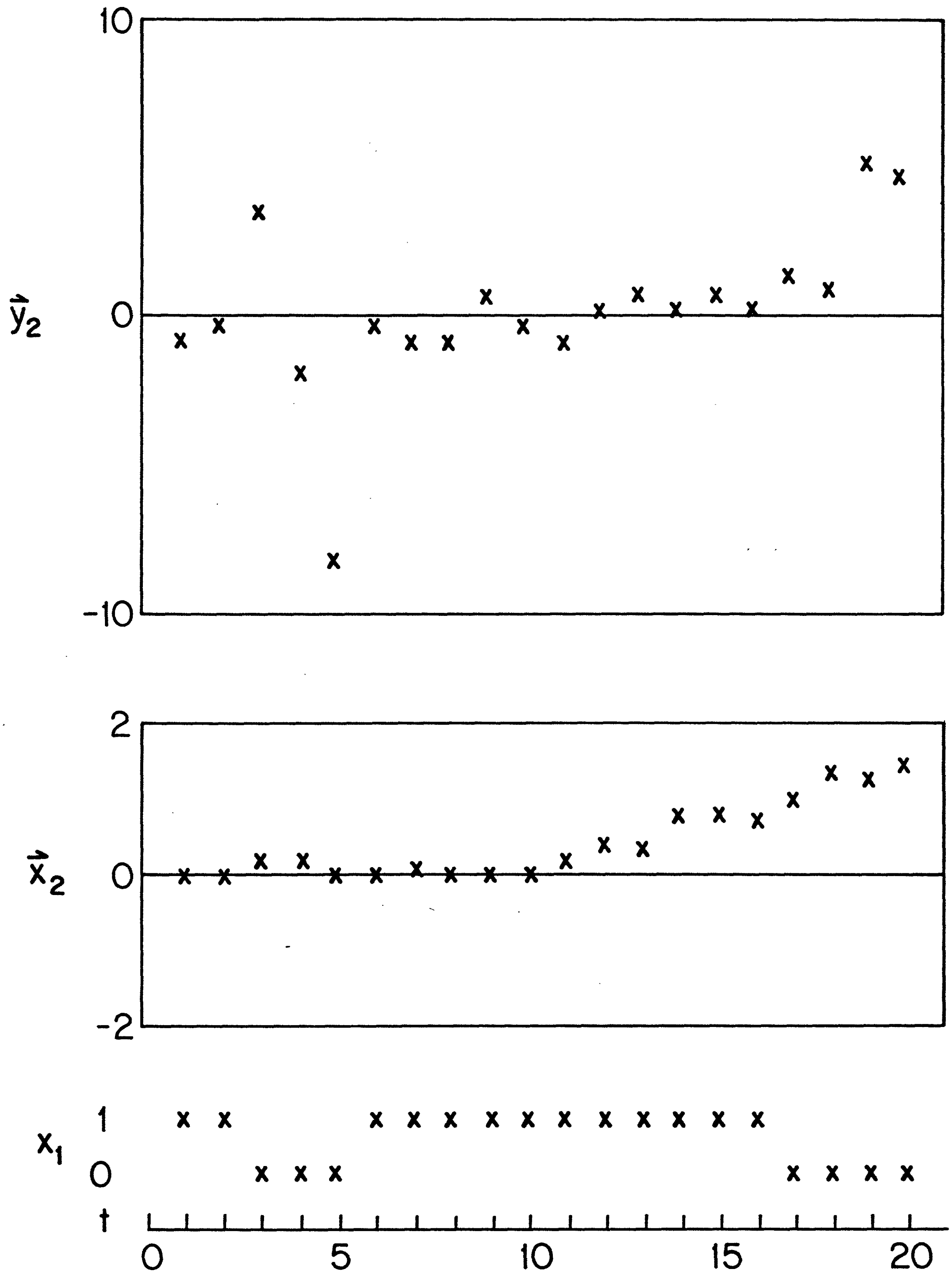


Figure 4. Sample Path of Model

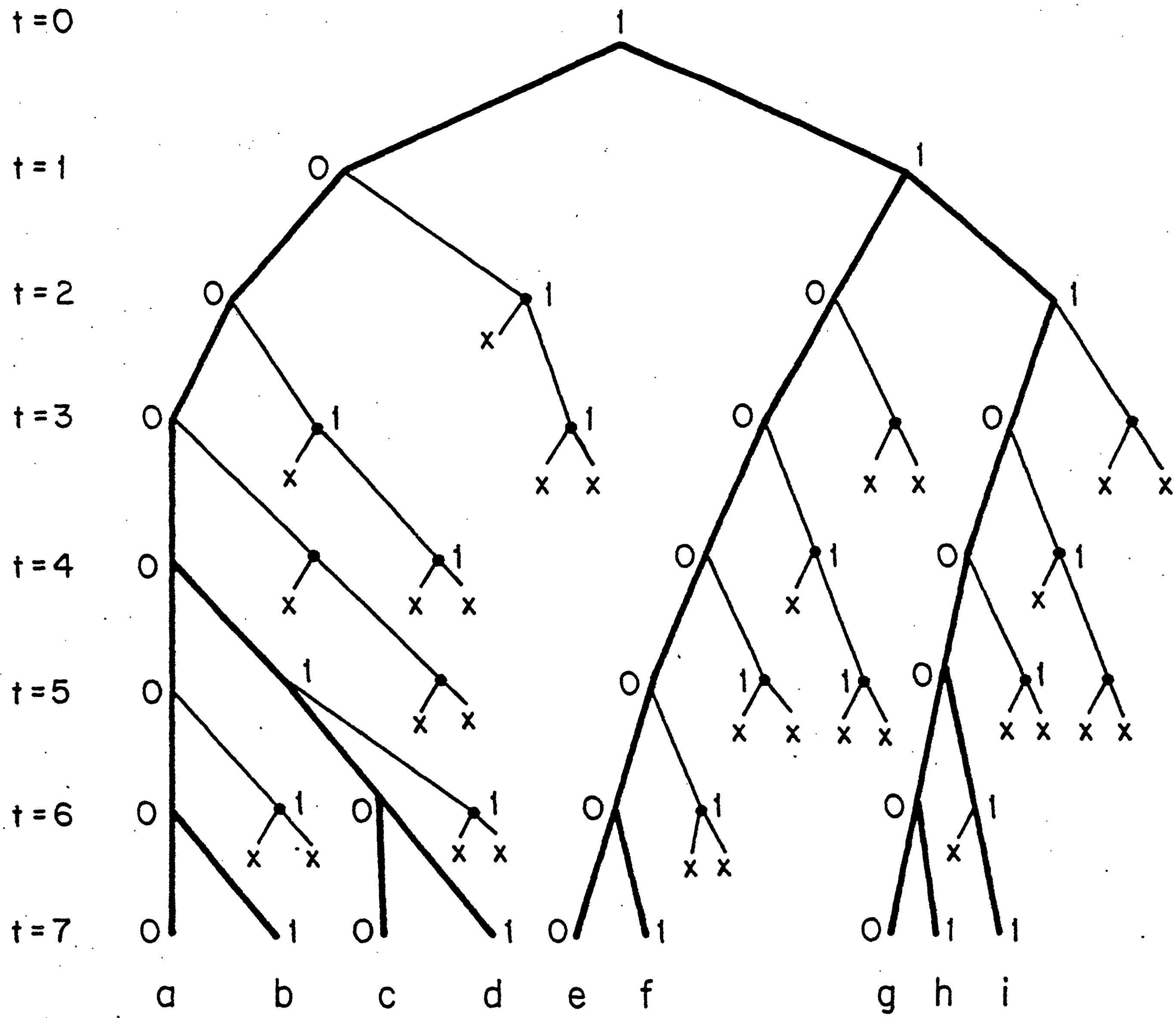


FIGURE 5: Example: Surviving X_1 at time 7

has been (correctly) determined to have arisen from $x_1 = 0$ through the memory of the supremal process.

There is a reason for the apparent dominance of $\underline{X}_1'$s terminating in a run of $x_1(t) = 0$. Consider the model. If $x(t) = 0$ then the Kalman filter estimate of $\vec{x}_2$ is not updated, and the associated conditional covariance $\underline{P}$ is larger than it would be on an identical path except with $x_1(t) = 1$, where $\vec{x}_2$ would be updated. Thus the covariances along trajectories with many $x_1(t) = 0$ will be larger than those with several $x_1(t) = 1$; the condition (5-14) gives preference to the former. In fact, the trajectory $x_1(t) \equiv 0$ will never be eliminated, for this reason; thus events $x_1(t) = 1$ will never be confirmed. Thus the optimal algorithm can only confirm events where $x_1(t) = 0$. (Intuitively, this is in anticipation of the possible, albeit unlikely, event that a future sequence of observations will fit the dynamics of x_2 perfectly, but for an initial state far from zero. If these were observed, the data thus far would be confirmed as all having come from the interference.)

Table 1 shows the parameters describing each surviving trajectory in figure 5. Most represent components of $s(x_1,\vec{x}_2|Y_1,Y_2)$ which are clustered near $\vec{x}_2 = .35$, and it is possible that the general mechanism of theorem 1 might eliminate one or more of these which were missed by that of theorem 3. Coincidentally, the survivor corresponding to the true trajectory, i, has the highest quality factor q.

Finally, figures 6 and 7 show the effectiveness of the optimal pruning mechanism over a longer period of time. Figure 6 compares, on a log scale, the actual number of survivors against the total size of $\underline{X}_1^t$. It is clear that, for this example at least, the exponential

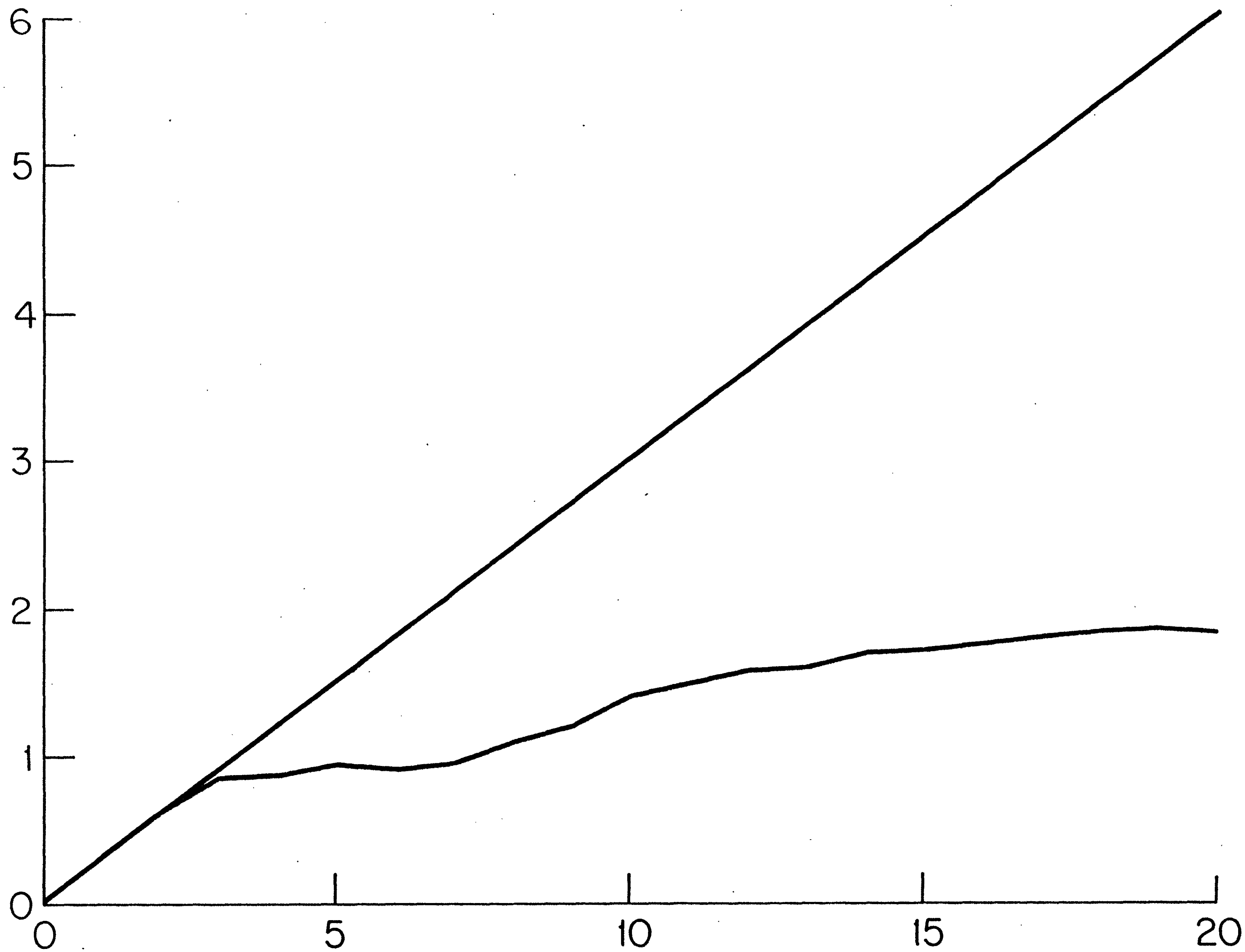


FIGURE 6: Logarithm of Survivors

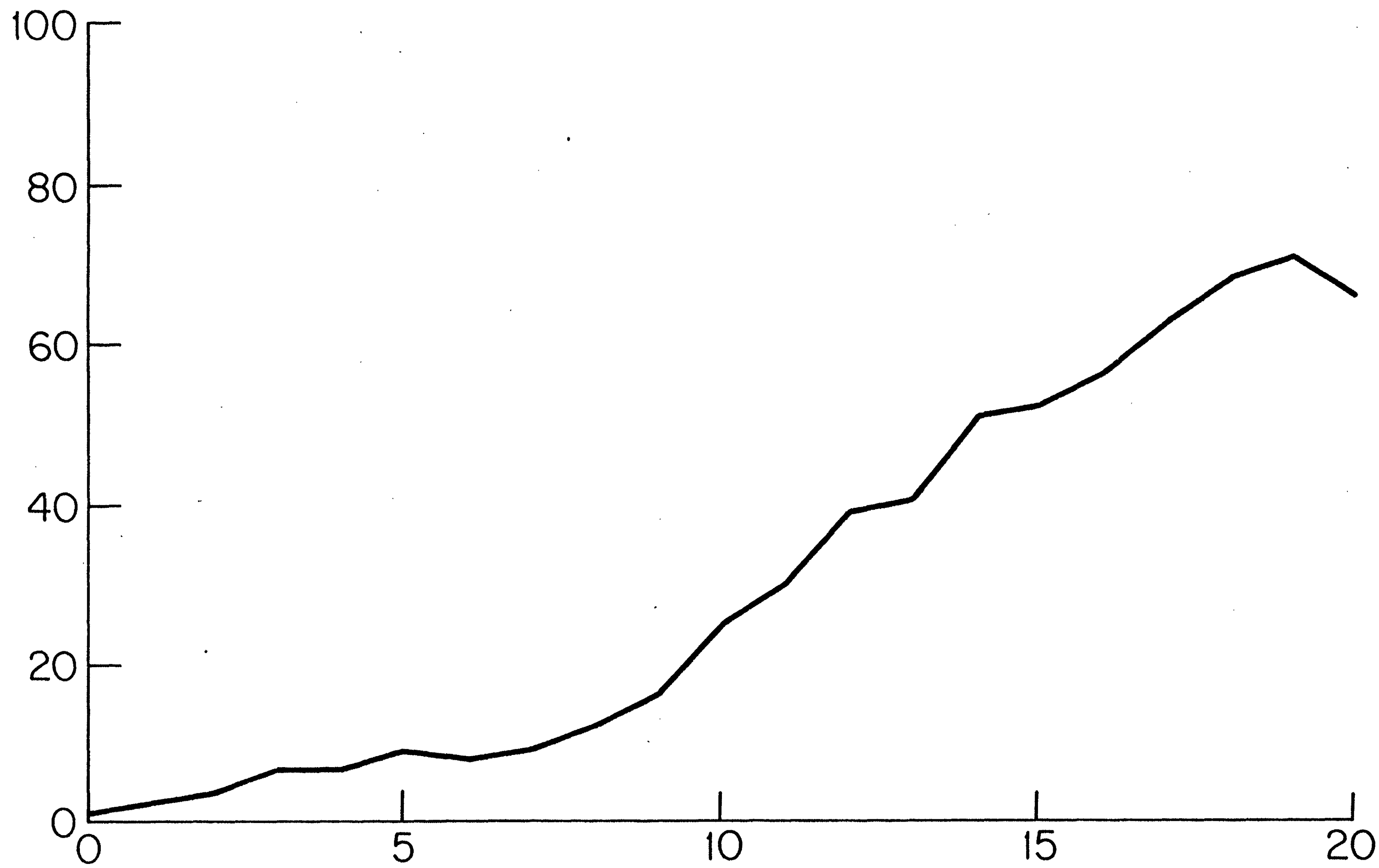


FIGURE 7: Actual Survivors

growth of candidate X_1's has been averted. Figure 7 shows the same count on a linear scale; the number of survivors seems to stay roughly constant when $X_1 = 0$ (as many trajectories with $x_1 = 1$ can be pruned) and to grow roughly linearly when $x_1 = 1$. In particular, the jump in y_2 at $t = 19$ causes a net reduction in the surviving X_1 at $t = 20$.

Thus the optimal pruning mechanism, while not complete, is still capable of significantly reducing the combinatorial aspect of the hybrid smoothing problem, at least for this example. A general categorization of its effectiveness is yet to be determined.

VII. CONCLUSIONS

In conclusion, this work has presented a new perspective on filtering and smoothing for hierarchical Markov processes, particularly hybrid state systems. The results fall into two categories. The negative results are that the hierarchical structure does not contribute to simplification of the solution to the state estimation problem, nor to the trajectory estimation problem for discrete state, or linear - Gaussian, problems. The positive results are related to the hybrid case, where both state and trajectory estimation are dominated by a structure involving combinations of weighted Gaussian terms. While both can then be realized by separate computations of the weights and parameters of the Gaussian shapes, only the smoothing problem affords us the opportunity to eliminate some of the components entirely. This simplification of the combinatorial aspect of the problem suggests adoption of the trajectory estimation approach to hybrid systems, particularly in light of the relationship between the parameters of the Gaussian components in the two cases;

they are computed by the same Kalman filters.

The results of the adoption of the trajectory estimation viewpoint is a pruning rule which is optimal in a well defined sense: the elimination of a trajectory is guaranteed to never increase the probability of error in estimating the discrete state trajectory. An example showed that this rule alone can be effective, but that some other selection mechanism is required in order to bound the number of survivors at a finite level.

Computationally, the structure of the algorithm described in Section V is ideal for VLSI implementation. The infemal calculations, involving Kalman filters and residual computations, are completely separate from one another and would benefit from parallel execution. The interconnection between them is provided by the (simple) supremal computation involving the discrete observation, and the pruning mechanism. The latter involves simple exchange and tests of the results of the separate infemal calculations, and thus is a relatively loosely coupled mechanism.

Thus this work presents a new approach to hybrid state tracking problems. While it does not completely specify an implementable algorithm, an approach which can reduce a set of 1,048,576 candidates to only 66 without increasing the probability of error is a useful first step.

APPENDIX A

Proof of Theorem 2

We seek to specialize Lemma 1 to the linear case, showing that

$$s(\vec{x},Y) = s_o\, e^{-\frac{1}{2}(\vec{x}-\hat{\vec{x}})^T \underline{P}^{-1}(\vec{x}-\hat{\vec{x}})} \qquad \text{(A-1)}$$

is the general form of the survivor function, and deriving recursive equations for s_o, $\hat{\vec{x}}$, and $\underline{P}$. The proof is inductive; assume that (A-1) holds after an update at time t (the basis of the induction is established at time 0 when $\vec{x}(0) \sim N(\hat{\vec{x}}(0),\ P(0))$ and $s(\vec{x}) = p(\vec{x})$).

Use (3-7) to predict to time t+1 :

$$s(\vec{x}^+|Y(t)) = \max_{\vec{x}} (2\pi)^{-N/2} (\det \underline{Q})^{-1/2}\, e^{-\frac{1}{2}(\vec{x}^+ - \underline{A}\vec{x})^T \underline{Q}^{-1}(\vec{x}^+ - \underline{A}\vec{x})}\; s_o(t)\, e^{-\frac{1}{2}(\vec{x}-\hat{\vec{x}})\underline{P}^{-1}(\vec{x}-\hat{\vec{x}})} \qquad \text{(A-2)}$$

The $\vec{x}$ which maximizes (A-2) is given by

$$\vec{x} = [\underline{P}^{-1} + \underline{A}^T \underline{Q}^{-1} \underline{A}]^{-1} [\underline{P}^{-1} x + \underline{A}^T \underline{Q}^{-1} x] \qquad \text{(A-3)}$$

The quadratic form in the exponent becomes

$$-\frac{1}{2}(\vec{x} - \hat{\vec{x}})^T \underline{P}^{-1} \underline{A}^{-1} [\underline{Q} + \underline{A}\,\underline{P}\,\underline{A}^T]\, \underline{A}^{-T} \underline{P}^{-1}(\vec{x} - \hat{\vec{x}}) \qquad \text{(A-4)}$$

which using (A-3) and

$$[\underline{P}^{-1} + \underline{A}^T \underline{Q}^{-1} \underline{A}]^{-1} = \underline{P} - \underline{P}\,\underline{A}^T [\underline{Q} + \underline{A}\,\underline{P}\,\underline{A}^T]^{-1} \underline{A}\,\underline{P} \qquad \text{(A-5)}$$

reduces to

$$-\frac{1}{2}(\vec{x}^+ - \underline{A}\hat{\vec{x}})^T [\underline{Q} + \underline{A}\,\underline{P}\,\underline{A}^T]^{-1} (\vec{x}^+ - \underline{A}\,\hat{\vec{x}}) \qquad \text{(A-6)}$$

This gives the prediction equations:

$$\hat{\vec{x}}(t+1 \mid Y(t)) = \underline{A}\ \hat{\vec{x}}(t \mid Y(t)) \qquad \text{(A-7)}$$

$$\underline{P}(t+1 \mid Y(t)) = \underline{A}\ \underline{P}(t) \mid Y(t))\ \underline{A}^T + \underline{Q} \qquad \text{(A-8)}$$

$$s_o(t+1 \mid Y(t)) = (2\pi)^{-N/2}\ (\det \underline{Q})^{-\frac{1}{2}}\ s_o(t \mid Y(t)) \qquad \text{(A-9)}$$

Now (3-8) updates these with y(t+1):

$$\vec{x}^+ \mid Y(t+1)) = (2\pi)^{-M/2}\ \det(\underline{R})^{-\frac{1}{2}(\vec{y}^+ - \underline{C}\vec{x}^+)^T \underline{R}^{-1} (\vec{y}^+ - \underline{C}\vec{x}^+)}$$

$$s_o(t+1 \mid Y(t))\ e^{-\frac{1}{2}(\vec{x}^+ - \hat{\vec{x}})^T \underline{P}^{-1} (\vec{x}^+ - \hat{\vec{x}})} \qquad \text{(A-10)}$$

where now

$$\vec{x}^+ = \vec{x}(t+1) \qquad \vec{y}^+ = \vec{y}(t+1) \qquad \text{(A-11)}$$

$$\hat{\vec{x}} = x(t+1) \mid Y(t))$$

Combining and completing the squares in (A-10) gives

$$s(\vec{x}^+ \mid Y(t+1)) = s_o(t+1) \mid Y(t))\ (2\pi)^{-M/2}\ \det(\underline{R})^{-1/2}\ e^{-\frac{1}{2}(\vec{y}^+ - \underline{C}\hat{\vec{x}})^T (\underline{C}\ \underline{P}\ \underline{C}^T + \underline{R})^{-1} (\vec{y}^+ - \underline{C}\hat{\vec{x}})}$$

$$e^{-\frac{1}{2}(\vec{x}^+ - \hat{\vec{x}}^+)(CPC^T + R)(\vec{x}^+ - \hat{\vec{x}}^+)} \qquad \text{(A-12)}$$

where

$$\hat{\vec{x}}^+ = \hat{\vec{x}} + \underline{K}(t+1)\ (\vec{y}^+ - \underline{C}\ \hat{\vec{x}}) \qquad \text{(A-13)}$$

This gives the update expressions:

$$\hat{\vec{x}}(t+1 \mid Y(t+1) = \hat{\vec{x}}(t+1) \mid Y(t)) + \underline{K}(t+1)(\vec{y}(t+1) - \underline{C}\ \hat{\vec{x}}(t+1 \mid Y(t))) \qquad \text{(A-14)}$$

$$\underline{K}(t+1) = \underline{P}(t+1 \mid Y(t))\ \underline{C}^T [\underline{C}\ \underline{P}(t+1 \mid Y(t))\ \underline{C}^T + \underline{R}]^{-1} \qquad \text{(A-15)}$$

$$\underline{P}(t+1)|Y(t+1) = [\underline{I} - \underline{K}(t+1)\underline{C}]\ \underline{P}(t+1|Y(t+1))[\underline{I} - \underline{K}(t+1)\underline{C}]^T + \underline{K}(t+1)\ \underline{R}\ \underline{K}^T(t+1) \quad \text{(A-16)}$$

$$s_o(t+1)|Y(t+1)) = s_o(t+1)|Y(t))(2\pi)^{-M/2} \det(\underline{R})^{-1/2} \quad \text{(A-17)}$$

$$e^{-\frac{1}{2}(\vec{y}(t+1)-\underline{C}\ \hat{\vec{x}}(t+1)|Y(t+1))^T \underline{S}^{-1}(\vec{y}(t+1) - \underline{C}\ \hat{\vec{x}}(t+1)|Y(t)))} \quad \text{(A-18)}$$

$$\underline{S} = \underline{C}\ \underline{P}(t+1|Y(t))\ \underline{C}^T + \underline{R}$$

Note that (A-14) - (A-16) are the usual Kalman filter equations; (A-17) - (A-18) accumulate the effect of the residuals

$$\vec{y}(t+1) - \underline{C}\ \hat{\vec{x}}(t+1|Y(t)) \quad \text{(A-19)}$$

weighted by their inverse covariance $\underline{S}$.

Combining (A-17) - (A-9) with (A-14) - (A-18) yields (4-2) - (4-6).

□

APPENDIX B

Proof of Theorem 3

We seek conditions equivalent to the statement

$$\tilde{q}_o \, e^{-\frac{1}{2}(\vec{x} - \tilde{\vec{x}}_2)^T \tilde{\underline{P}}^{-1} (\vec{x} - \tilde{\vec{x}}_2)} \leq q_o \, e^{-\frac{1}{2}(\vec{x} - \vec{x}_2) \, \underline{P}^{-1} \, (\vec{x} - \vec{x}_2)} \tag{B-1}$$

for all $\vec{x}$.

Taking logarithms and rearranging terms, this is equivalent to

$$\frac{1}{2} \vec{x}^T [\tilde{\underline{P}}_o^{-1} - \underline{P}^{-1}] \, \vec{x} - \frac{1}{2} \vec{x}^T [\tilde{\underline{P}}^{-1} \tilde{\vec{x}}_2 - \underline{P}^{-1} \vec{x}_2] - \frac{1}{2} [\tilde{\underline{P}}^{-1} \tilde{\vec{x}}_2 - \underline{P}^{-1} \vec{x}_2]^T \vec{x}$$

$$+ \frac{1}{2} \tilde{\vec{x}}^T \tilde{\underline{P}}^{-1} \tilde{\vec{x}} - \frac{1}{2} \vec{x}_2^T \, P^{-1} \, \vec{x}_2 \geq \ell n \, \tilde{q}_o - \ell n \, q_o \tag{B-2}$$

On the left is a quadratic function of $\vec{x}$. It will be bounded below by a finite constant only if

$$\tilde{\underline{P}}^{-1} - \underline{P}^{-1} \geq \underline{0} \tag{B-3}$$

This gives (5-14). The inequality will hold iff the minimum of the quadratic function is greater than the right hand side. That minimum is achieved[1] at

$$\vec{x} = [\tilde{\underline{P}}^{-1} - \underline{P}^{-1}]^{-1} \; [\tilde{\underline{P}}^{-1} \tilde{\vec{x}}_1 - \tilde{\underline{P}}^{-1} \vec{x}_2] \tag{B-4}$$

Using the fact

$$[\tilde{\underline{P}}^{-1} - \underline{P}^{-1}]^{-1} = \tilde{\underline{P}} - \tilde{\underline{P}}[\tilde{\underline{P}} - \underline{P}]^{-1} \, \tilde{P} \tag{B-5}$$

and substituting into (B-2), the inequality holds iff

$$\frac{1}{2}(\tilde{\vec{x}}_2 - \vec{x}_2)^T [\tilde{\underline{P}} - \underline{P}]^{-1} \; (\tilde{\vec{x}}_2 - \vec{x}_2) \geq \ell n \, \tilde{q}_o - \ell n \, q_o \tag{B-6}$$

This is (5-15).

---

[1] This assumes (B-3) is strict. If not, reformulate this entire development in the largest subspace of $\underline{X}_2$ on which (B-3) holds strictly.

References

1. A.S. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems", Automatica, Vol. 12, pp. 6-1-611, Pergammon Press, 1976.

2. Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment", IEEE Trans. AC, Aug. 1978, pp. 618-626.

3. D.G. Laniotis, "Joint Detection, Estimation, and System Identification", Information and Control, Vol. 19, pp.75-92, Aug. 1971

4. R.R. Tenney, R.S. Hebbert, and N.R. Sandell, Jr., "A Tracking Filter for Maneuvering Sources", IEEE Trans. A.C., April 1977, pp.246-251.

5. H.N. Psaraftis, A.N. Perakis, and P.N. Mikhalevsky, "New Models on the Ocean Acoustic Detection Process", J. Acoust. Soc. Am., Vol. 69, No. 6, June 1981.

6. F.B. Bruneau, "State Estimation of a Hybrid Markov Process with Application to Multitarget Tracking", LIDS-TH-1172, M.I.T., Cambridge, MA. Jan. 1982.

7. G.D. Forney, "The Viterbi Algorithm", IEEE Proceedings, March 1973.

8. A. Gelb, Ed., Applied Optimal Estimation, M.I.T. Press, Cambridge, MA. 1974.

9. B.C. Levy, D.A. Castanon, G.C. Verghese, and A.S. Willsky, "A Scattering Framework for Decentralized Estimation Problems", Proc. 21st CDC, San Diego, Ca., Dec. 1981

10. P.J. Buxbaum and R.A. Haddad, "Recursive Optimal Estimation for a Class of NonGaussian Processes", Proc. Symp. on Comp. Proc. in Comm., Polytechnic Institute of Brooklyn, 1969.

11. J.K. Tugnait, "Detection and Estimation for Abruptly Changing Systems", Automatica, Vol. 18, No. 5, Aug. 1982.