Hidden Markov Model State Estimation with Randomly Delayed Observations

Jamie S. Evans and Vikram Krishnamurthy, *Senior Member, IEEE*

***Abstract*—This paper considers state estimation for a discrete-time hidden Markov model (HMM) when the observations are delayed by a random time. The delay process is itself modeled as a finite state Markov chain that allows an augmented state HMM to model the overall system. State estimation algorithms for the resulting HMM are then presented, and their performance is studied in simulations. The motivation for the model stems from the situation when distributed sensors transmit measurements over a connectionless packet switched communications network.**

I. Introduction

IN THIS PAPER, we consider the state estimation (filtering) problem for a noisily observed finite state Markov chain when the observations are delayed by a random time. If the observations were always received in order, the optimal state estimates would be obtained from the standard hidden Markov model (HMM) state filter [1], [2]. However, we allow for the random delays to lead to a possible reordering of the measurements. With the delays modeled as a finite-state Markov chain, we show how the problem can be reformulated as a standard HMM filtering problem and give a recursive update equation for calculating the conditional state probabilities. We also show that a finite-dimensional recursive filter results when the delays are modeled as continuous random variables with density indexed by a finite-state Markov chain. We begin by detailing the motivation for this paper and introducing the HMM paradigm.

Motivation (Communication in Distributed Systems): One motivation for this problem stems from the increasing movement toward distributed sensing and processing systems. We envisage a system with a number of remote sensors with limited processing capability that communicate measurements back to processing units, which carry out the estimation (and control) algorithms. Ideally, dedicated communication resources (bandwidth) would be made available between each sensor/processor pair; however, due to the increasingly large number of sensors in modern systems, this may not be a viable option. The flexible and cost-effective alternative is to use a shared resource communications network to service the communication needs of all the system components. In this paper, we assume that a connectionless, packet-switched network [3] is employed.

Each packet that is sent from the sensor will arrive at the processor after a random delay that is dependent on the current level of congestion in the network and the path that is taken. This means that packets may arrive out of order at the processor. Normally, packets would contain a sequence number that would allow them to be reordered at the destination; however, we consider the situation when no such information is sent with the packets. We assume that each packet contains only one measurement that avoids any extra delay in communicating the observations to the processing unit. We also assume that each measurement is received without error and that the effect of transmitting a finite-precision approximation to the real-valued observations is negligible.

Hidden Markov Models: An HMM consists of a *signal* modeled as a finite state Markov chain and an *observation* model that relates an observed process to the underlying Markov chain. Typically, the observation model consists of observing the state of the Markov chain perturbed by additive white noise. Such models have become increasingly popular over the last decade: application areas including speech processing, target tracking, digital communications, biomedical engineering, and finance (see [1], [2], and references therein).

A major reason for this is the enormous flexibility and generality of the model and the fact that efficient state and parameter estimation algorithms exist and are well understood. In particular, the finite-state property means that finite-dimensional state filters result even when the model is nonlinear. This makes the HMM formulation very attractive for approximating continuous state space nonlinear models for which finite-dimensional filters rarely exist.

In this paper, the HMM observation model includes the possibility of the noisy observations being reordered as a result of the random time delay introduced during communication of the observation from the sensor to the processing unit. We will show that, provided dependency between the delays is governed by another finite-state Markov chain, a finite-dimensional recursion for the conditional state probabilities results.

Related Work: Before proceeding, we note that in [4], Nilsson and Bernhardsson also consider an environment where measurements (and control signals) are sent over a

Manuscript received July 31, 1997; revised January 20, 1999. This work was supported in part by the Centre for Signal, Sensor, and Information Processing, the Australian Telecommunications and Electronics Research Board, the Australian Research Council, and a Telstra Research Laboratories Postgraduate Fellowship. The associate editor coordinating the review of this paper and approving it for publication was Prof. M. H. Er.

J. S. Evans was with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA. He is now with the School of Electrical and Information Engineering, University of Sydney, Sydney, Australia.

V. Krishnamurthy is with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria, Australia (e-mail: vikram@ee.mu.oz.au).

Publisher Item Identifier S 1053-587X(99)05404-5.

communications network and thereby suffer a random delay. In [4], the authors treat an extension of the standard linear quadratic Gaussian (LQG) control problem, which includes the randomly delayed measurement and control signals. The delay is modeled as a continuous random variable with density indexed by a finite-state Markov chain (we employ a similar model in Section V). They assume that the state of this chain is known to the controller and that the delays are such that the order of measurement and control signals is not corrupted.

We note that our basic approach for dealing with delay models that do not satisfy these assumptions can be applied in the linear-Gaussian case as well as for (finite-state) HMM's. In the linear-Gaussian case, however, the resultant filtering algorithms would have a computational cost that grows exponentially in the data length. Practical (suboptimal) algorithms for the linear-Gaussian case is an area for future work. We are also currently looking at the related partially observed stochastic control problem for both the HMM and linear-Gaussian systems.

Summary and Contributions: We are interested in exploring the extent to which the uncertainty in the arrival order of the observations can be overcome by taking the statistics of the various stochastic processes into account. After stating the problem more precisely in Section II, we show in Section III how it can be reformulated as a standard HMM filtering problem. Once this formulation is in place, we are able to derive the recursive filter equations for the conditional state probabilities in Section IV. State estimates such as the minimum mean squared error (MMSE) or the maximum *a posteriori* (MAP) estimates can be directly calculated from the filtered state probabilities. In Section V, we show how the results of Section IV can be extended to more complicated delay models. In particular, we examine the situation where the delay is modeled as a mixture distribution with dependency governed by a finite-state Markov chain. In Section VI, we examine the use of state aggregation to reduce the computational load of the optimal filters. The resultant filters are suboptimal but are shown to perform close to optimal in the simulations of Section VII. Finally, we discuss some interesting extensions and directions for future research in Section VIII.

II. Problem Formulation

In this section, we introduce the basic stochastic model for our problem. This includes the model for the Markov chain whose state we wish to estimate, the sensor observation model, the delay model, and the model for the observations received at the processing unit. In this section, we model the delay as a finite-state Markov chain; however, the important extension to a continuous delay distribution indexed by a Markov chain is treated in Section V. An overview of the various processes involved is given in Fig. 1.

Signal Model: The signal of interest is modeled as an M-state, homogeneous Markov chain $s(k)$ with state space $\{1, 2, \ldots, M\}$. The transition probability matrix is $A = [a_{ij}]_{M \times M}$, where

$$a_{ij} = P(s(k+1) = j \mid s(k) = i), \quad i, j \in \{1, 2, \ldots, M\}$$

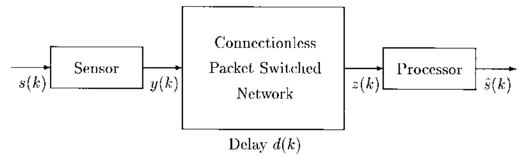


Fig. 1. State estimation with communication over a packet switched network.

with $\sum_{j=1}^{M} a_{ij} = 1$, and the state levels are given by the vector $g = [g_1, g_2, \ldots, g_M]$.

Sensor Observation Model: The states are observed in noise at a sensor leading to the sensor observation process $y(k)$ defined by

$$y(k) = g_{s(k)} + w(k), \quad (k \geq 1) \tag{1}$$

where $w(k)$ is an independent and identically distributed (iid) sequence, each random variable having density $\phi(\cdot)$. The sequence $w(k)$ is assumed independent of $s(k)$. We will denote $Y(k) = \{y(1), \ldots, y(k)\}$.

Delay Model: As already discussed in Section I, the sensor observations are sent over a connectionless, packet-switched communication network to a processing unit that will carry out the desired estimation algorithms. The processor receives observation $y(k)$ after some random delay $d(k)$.

We model the delay $d(k)$ as an N state, homogeneous Markov chain with state space $\{1, 2, \ldots, N\}$ and transition probability matrix $B = [b_{ij}]_{N \times N}$, where

$$b_{ij} = \Pr(d(k+1) = j \mid d(k) = i), \quad i, j \in \{1, 2, \ldots, N\}$$

with $\sum_{j=1}^{N} b_{ij} = 1$. We assume that $d(k)$ is independent of the chain $s(k)$ and the sensor observation noise process $w(k)$. The levels of the chain are given by the vector $h = [h_1, h_2, \ldots, h_N]$, meaning that the delay suffered by $y(k)$ is h_i when $d(k) = i$.

Remark: At first, it may seem unrealistic to model the delay as a finite state process. However, we note that many continuous state processes can be approximated by a finite-state process. Further, the Markov model allows a fairly general dependency structure. Given that the proposed delay model leads to a very nice reformulation of the problem, we believe the assumption is well justified. In Section V, we treat a more involved delay model in which the delay is a continuous random variable with distribution indexed by a finite state Markov chain. Finite-dimensional recursions for the conditional state probabilities are also derived in this case; however, the problem no longer has a clean reformulation as a standard HMM.

Processor Observation Model: We assume the underlying time between sending measurements is Δ units and make the following restriction on the delay values:

$$|h_i - h_j| \neq l\Delta, \quad (l > 0), \quad i, j \in \{1, 2, \ldots, N\}. \tag{2}$$

Remark: This is a minor restriction that results in a simplified reformulation of the problem. It guarantees that two observations do not arrive simultaneously at the processor and thus avoids having to define a random ordering procedure.

The observations may arrive at the processor in a different order than they were sent. We assume that no order information is sent with the measurements. The measurements are read by the processor in the order they arrive, and no further arrival time information is observed by the processor. Let the kth measurement to arrive at the processor be $z(k)$, and denote the first k observations received by $Z(k) = \{z(1), z(2), \ldots, z(k)\}$.

Aim: The objective is to perform state estimation (filtering) for the Markov chain $s(k)$ based on the observations received at the processor. More specifically, we wish to obtain recursive expressions for the conditional state probabilities

$$P(s(k) = i \mid Z(k)), \quad i \in \{1, \ldots, M\}$$

from which the maximum *a posteriori* (MAP) or minimum mean squared error (MMSE) state estimates can be obtained. The problem is made interesting by the fact that the order of observations may be altered as a result of transmission over the network. While no order information is explicitly available, information is present in the measurements received and in the assumed statistics of the Markov chains. In the next section, we reformulate the problem as a standard HMM and then show how the optimal estimation algorithms result.

III. Reformulation as an HMM

In this section, we reformulate the model as a standard HMM problem on an enlarged state space. The trick is to form a new finite-state Markov chain by grouping a number of original states together. The observations at the processor can then be expressed as a function of the new Markov chain plus a white noise process resulting in the standard HMM formulation.

A. Enlarged State Space Markov Chains

To begin, we let

$$H = \max_{i,j} \left\lfloor \left| \frac{h_i - h_j}{\Delta} \right| \right\rfloor \tag{3}$$

which corresponds to the greatest possible change in the position of a measurement as a result of transmission over the network.

Define the new process

$$S(k) = [s(k+H), \ldots, s(k-H)]^T \quad (k \geq 0). \tag{4}$$

Then, $S(k)$ is an $M^{(2H+1)}$ state Markov chain with state space $\{1, 2, \ldots, M\}^{(2H+1)}$. The transition probabilities of $S(k)$ are

$$\begin{aligned} &P(S(k+1) = (j_1, \ldots, j_{2H+1}) \mid S(k) = (i_1, \ldots, i_{2H+1})) \\ &\quad = a_{i_1 j_1} \cdot \delta(j_2 - i_1) \cdots \delta(j_{2H+1} - i_{2H}) \end{aligned} \tag{5}$$

for $j_1, \ldots, j_{2H+1}, i_1, \ldots, i_{2H+1} \in \{1, 2, \ldots, M\}$, where $\delta(\cdot)$ denotes the Kronecker delta function. The initial conditions are given by

$$P(S(0) = (i_1, \ldots, i_{2H+1})) = \sigma_0(i_1, \ldots, i_{2H+1}) \tag{6}$$

for $(i_1, \ldots, i_{2H+1}) \in \{1, 2, \ldots, M\}^{(2H+1)}$.

TABLE I
Determining the Arrival Order of the Packets

packet	$k+H$	$k+H-1$	$\cdots$	$k-H$
relative departure time	$2H\Delta$	$(2H-1)\Delta$	$\cdots$	0
delay	$d(k+H)$	$d(k+H-1)$	$\cdots$	$d(k-H)$
relative arrival time	$d(k+H)+2H\Delta$	$d(k+H-1)+(2H-1)\Delta$	$\cdots$	$d(k-H)$

Similarly, we define

$$D(k) = [d(k+H), \ldots, d(k-H)]^T \quad (k \geq 0) \tag{7}$$

which is a $N^{(2H+1)}$ state Markov chain with state space $\{1, 2, \ldots, N\}^{(2H+1)}$ and transition probabilities

$$\begin{aligned} &P(D(k+1) = (j_1, \ldots, j_{2H+1}) \mid D(k) = (i_1, \ldots, i_{2H+1})) \\ &\quad = b_{i_1 j_1} \cdot \delta(j_2 - i_1) \cdots \delta(j_{2H+1} - i_{2H}) \end{aligned} \tag{8}$$

for $j_1, \ldots, j_{2H+1}, i_1, \ldots, i_{2H+1} \in \{1, 2, \ldots, N\}$. The initial conditions are given by

$$P(D(0) = (i_1, \ldots, i_{2H+1})) = \pi_0(i_1, \ldots, i_{2H+1}) \tag{9}$$

for $(i_1, \ldots, i_{2H+1}) \in \{1, 2, \ldots, N\}^{(2H+1)}$.

Remark: While we find it convenient to work with the two Markov chains defined in (4) and (7), the two chains could readily be combined into a single chain on the state space $\{1, \ldots, M\}^{(2H+1)} \times \{1, \ldots, N\}^{(2H+1)}$. Transition probabilities follow immediately upon observing the independence of the chains.

B. Processor Observation Model

Since the maximum possible position change for any measurement is equal to H places, the kth measurement to arrive at the processor $(z(k))$ corresponds to a sensor measurement from the set $\{y(k+H), \ldots, y(k-H)\}$. Further, $z(k)$ can be mapped to the appropriate measurement given the current augmented delay state $D(k)$. Thus, we have

$$z(k) = f(D(k))[y(k+H), \ldots, y(k-H)]^T \tag{10}$$

where $f(D(k)) = e_i^T$ when $z(k)$ corresponds to $y(k+H+1-i)$, and e_i is the unit column vector in $R^{(2H+1)}$.

To determine the relative arrival order of the measurements $y(k+H), \ldots, y(k), \ldots, y(k-H)$ given the delay values $d(k+H), \ldots, d(k), \ldots, d(k-H)$, we proceed as follows (see Table I).

1) Determine the arrival times referenced to the time $y(k-H)$ was sent

$$d(k+H) + 2H\Delta, \ldots, d(k-H).$$

2) The relative order of arrival is obtained by ordering these arrival times.

3) The measurement corresponding to $z(k)$ is the $(H+1)$th element in the reordered sequence.

We can thus define $f : \{1, 2, \ldots, N\}^{(2H+1)} \rightarrow \{e_1, e_2, \ldots, e_{2H+1}\}$, where for $i \in \{1, 2, \ldots, N\}$ and $j \in \{1, 2, \ldots, 2H+1\}$

$$f((i_1, i_2, \ldots, i_{2H+1})) = e_j^T$$

if

$$\sum_{l=1}^{2H+1} \operatorname{sign}((h_{i_j} - h_{i_l}) - (j-l)\Delta) = 0$$

where

$$\operatorname{sign}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0. \end{cases}$$

From [1] and [10], we thus have

$$\begin{aligned} z(k) &= f(D(k))\big[g_{s(k+H)}, \ldots, g_{s(k-H)}\big]^T \\ &\quad + f(D(k))[w(k+H), \ldots, w(k-H)]^T \end{aligned}$$

which can be expressed in standard HMM form

$$z(k) = \Gamma(S(k), D(k)) + \Sigma(S(k), D(k))W(k)$$

where

$$\begin{aligned} \Gamma(S(k), D(k)) &= f(D(k))\big[g_{s(k+H)}, \ldots, g_{s(k-H)}\big]^T \\ \Sigma(S(k), D(k)) &= f(D(k)) \\ W(k) &= [w(k+H), \ldots, w(k-H)]^T \end{aligned} \quad . \tag{11}$$

Note that $W(k)$ is a sequence of correlated random vectors. Generally, the fact that the observation noise is colored would lead to an exponential growth in the computational requirements of the estimation algorithms. However, in this case, only one component of the vector is selected at each time, and each selected random variable is independent of the others. We can thus rewrite the processor observation model as

$$z(k) = \Gamma(S(k), D(k)) + v(k) \tag{12}$$

where $v(k)$ is a sequence of iid random variables each having density $\phi(\cdot)$ obtained from a reordering of the original $w(k)$ process. Like $w(k)$, the process $v(k)$ is independent of the chains $S(k)$ and $D(k)$.

Summary: In summary, the problem that was formulated in Section II has been reformulated in terms of the augmented Markov chains $S(k)$ and $D(k)$ defined in (4)–(9) and the processor observation model of (12). The model now appears as a standard hidden Markov model with transition probabilities given by (5) and (8) and symbol probabilities given by $P(z(k) \mid S(k), D(k))$.

IV. State Estimation Algorithm

Here, we consider the problem of calculating the conditional state probabilities for $S(k)$ and $D(k)$ given $Z(k) = \{z(1), z(2), \ldots, z(k)\}$. That is, we want to obtain recursive expressions for

$$\sigma_k(i_1, \ldots, i_{2H+1}) = P(S(k) = (i_1, \ldots, i_{2H+1}) \mid Z(k))$$

and

$$\pi_k(j_1, \ldots, j_{2H+1}) = P(D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k)).$$

First, define the joint conditional density

$$\begin{aligned} &\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \\ &\quad = P(S(k) = (i_1, \ldots, i_{2H+1}) \\ &\qquad D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k)). \end{aligned}$$

We have from Bayes' rule

$$\begin{aligned} &P(S(k) = (i_1, \ldots, i_{2H+1}), D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k)) \\ &\quad = CP(z(k) \mid S(k) = (i_1, \ldots, i_{2H+1}) \\ &\qquad D(k) = (j_1, \ldots, j_{2H+1}), Z(k-1)) \\ &\qquad \times P(S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k-1)) \end{aligned}$$

where C is a normalization constant. However, since $S(k)$ and $D(k)$ are Markov chains

$$\begin{aligned} &P(S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k-1)) \\ &\quad = \sum_{m_1} \cdots \sum_{m_{2H+1}} \sum_{n_1} \cdots \sum_{n_{2H+1}} \\ &\qquad \big(a_{m_1 i_1} \cdot \delta(i_2 - m_1) \cdots \delta(i_{2H+1} - m_{2H})\big) \\ &\qquad \times \big(b_{n_1 j_1} \cdot \delta(j_2 - n_1) \cdots \delta(j_{2H+1} - n_{2H})\big) \\ &P(S(k-1) = (m_1, \ldots, m_{2H+1}) \\ &\quad D(k-1) = (n_1, \ldots, n_{2H+1}) \mid Z(k-1)). \end{aligned}$$

Combining the above results, we arrive at the following theorem (also see [1] and [2]).

Theorem 1: The recursion for the filtered state probabilities is given by

$$\begin{aligned} &\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \\ &\quad = C\phi(z(k) - \Gamma((i_1, \ldots, i_{2H+1}), (j_1, \ldots, j_{2H+1}))) \\ &\qquad \times \sum_{m_1} \cdots \sum_{m_{2H+1}} \sum_{n_1} \cdots \sum_{n_{2H+1}} \\ &\qquad \big(a_{m_1 i_1} \cdot \delta(i_2 - m_1) \cdots \delta(i_{2H+1} - m_{2H})\big) \\ &\qquad \times \big(b_{n_1 j_1} \cdot \delta(j_2 - n_1) \cdots \delta(j_{2H+1} - n_{2H})\big) \\ &\qquad \times \alpha_{k-1}(m_1, \ldots, m_{2H+1}, n_1, \ldots, n_{2H+1}) \end{aligned} \tag{13}$$

for $k \geq 1$ and $(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \in \{1, \ldots, M\}^{(2H+1)} \times \{1, \ldots, N\}^{(2H+1)}$, where $\Gamma(\cdot)$ is defined in (11). The initial conditions for the recursion are given by

$$\begin{aligned} &\alpha_0(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \\ &\quad = \sigma_0(i_1, \ldots, i_{2H+1})\pi_0(j_1, \ldots, j_{2H+1}) \end{aligned}$$

for $i_1, \ldots, i_{2H+1} \in \{1, \ldots, M\}$ and $j_1, \ldots, j_{2H+1} \in \{1, \ldots, N\}$.

The marginal densities are then obtained from

$$\begin{aligned} &\sigma_k(i_1, i_2, \ldots, i_{2H+1}) \\ &\quad = \sum_{j_1} \cdots \sum_{j_{2H+1}} \alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \\ &\pi_k(j_1, j_2, \ldots, j_{2H+1}) \\ &\quad = \sum_{i_1} \cdots \sum_{i_{2H+1}} \alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \end{aligned}$$

and similarly, we can obtain the final filtered state probabilities for the original Markov chains as

$$P(s(k) = i_{H+1} \mid Z(k)) = \sum_{i_1} \cdots \sum_{i_H} \sum_{i_{H+2}} \cdots \sum_{i_{2H+1}} \sigma_k(i_1, i_2, \ldots, i_{2H+1}) \quad (14)$$

$$P(d(k) = j_{H+1} \mid Z(k)) = \sum_{j_1} \cdots \sum_{j_H} \sum_{j_{H+2}} \cdots \sum_{j_{2H+1}} \pi_k(j_1, j_2, \ldots, j_{2H+1}). \quad (15)$$

Computation: A naive implementation of (13) would require $O(M^{4H+2}N^{4H+2})$ computations at each time update. However, because of the special structure of the transition probabilities (most are zero), we only require $O(M^{2H+2}N^{2H+2})$ computations at each time. This is clear on rewriting the recursion (13) as

$$\begin{aligned}\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) &= C\phi(z(k) - \Gamma((i_1, \ldots, i_{2H+1}), (j_1, \ldots, j_{2H+1}))) \\ &\times \sum_{m_{2H+1}} \sum_{n_{2H+1}} a_{i_2 i_1} b_{j_2 j_1} \\ &\times \alpha_{k-1}(m_1, \ldots, m_{2H+1}, n_1, \ldots, n_{2H+1}). \quad (16)\end{aligned}$$

In Section VI, we give a suboptimal algorithm based on aggregation that significantly reduces the computational load.

Other Estimation Problems: Once the HMM formulation is in place, it is straightforward to solve numerous estimation problems using standard techniques, e.g., fixed interval smoothing using the forward–backward algorithm, MAP sequence estimation using the Viterbi algorithm, and parameter estimation using the Baum–Welch re-estimation equations. See [1] and the references therein for further details.

V. Extension to Mixture Delay Model

In previous sections, we have modeled the delay as a finite-state Markov chain. This allowed the problem to be reformulated as a standard HMM state estimation problem. In this section, we extend the ideas of the earlier sections to the situation where the delays are continuous random variables indexed by a finite state Markov chain.

Let the delay suffered by the kth packet or observation be $\tau(k)$. We model $\tau(k)$ as a continuous random variable with density belonging to the set $\{\psi_1, \ldots, \psi_N\}$. The particular density applying at time k is determined by the state of the N-state Markov chain $d(k)$ with state space $\{1, \ldots, N\}$ through

$$P(\tau(k) \in A \mid d(k) = i) = \int_A \psi_i(\tau)\, d\tau \quad i \in \{1, \ldots, N\}.$$

The density of $\tau(k)$ [not conditioned on $d(k)$] is thus the mixture density

$$\psi(z; k) = \sum_{i=1}^{N} P(d(k) = i)\psi_i(z).$$

Let $d(k)$ have transition probability matrix $B = [b_{ij}]_{N \times N}$, where

$$b_{ij} = \Pr(d(k+1) = j \mid d(k) = i), \quad i, j \in \{1, 2, \ldots, N\}$$

with $\sum_{j=1}^{N} b_{ij} = 1$ and assume that $d(k)$ is independent of the chain $s(k)$ and the sensor observation noise process $w(k)$.

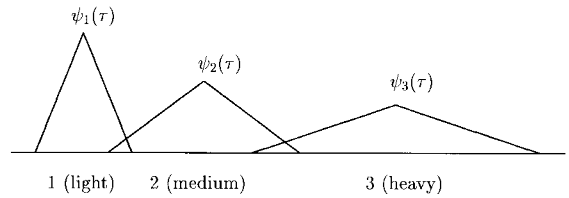


Fig. 2. Example densities for mixture delay model.

Further, assume that $\tau(k)$ is conditionally independent of all other processes given $d(k)$.

Example: Consider the situation shown in Fig. 2 (a similar delay model is introduced in [4]). We have a density corresponding to each of $N = 3$ levels of network congestion: 1 for light load, 2 for medium load, and 3 for heavy load.

We will assume that for each $i \in \{1, \ldots, N\}$, the delay density $\phi_i(\tau)$ is zero outside the interval $[L_i, U_i]$, where $0 \le L_i < U_i < \infty$. Define the integer

$$H = \max_{i,j} \left\lfloor \left| \frac{U_i - L_j}{\Delta} \right| \right\rfloor \quad (17)$$

which is equivalent to (3) for the finite-state delay model.

We continue to use the augmented chains $S(k)$ and $D(k)$ defined in (4) and (7), respectively, and define the new process

$$T(k) = [\tau(k+H), \ldots, \tau(k-H)]^T.$$

While it is no longer possible to write the observation $z(k)$ as a function of $D(k)$ and $S(k)$ perturbed by white noise as in (12), we will show that it is still possible to derive a finite-dimensional recursion of the conditional state probabilities.

We begin by again defining

$$\begin{aligned}\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) &= P(S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k)).\end{aligned}$$

Using Bayes' rule, we then have

$$\begin{aligned}\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) &= Cp(z(k) \mid S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}), Z(k-1)) \\ &\times P(S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}) \mid Z(k-1))\end{aligned}$$

where C is a normalizing constant. Using a model update equation, the second term can be written in the form

$$\begin{aligned}&\sum_{m_1} \cdots \sum_{m_{2H+1}} \sum_{n_1} \cdots \sum_{n_{2H+1}} \\ &\left(a_{m_1 i_1} \cdot \delta(i_2 - m_1) \cdots \delta(i_{2H+1} - m_{2H})\right) \\ &\times \left(b_{n_1 j_1} \cdot \delta(j_2 - n_1) \cdots \delta(j_{2H+1} - n_{2H})\right) \\ &\times \alpha_{k-1}(m_1, \ldots, m_{2H+1}, n_1, \ldots, n_{2H+1})\end{aligned}$$

and it remains for us to express

$$\begin{aligned}&p(z(k) \mid S(k) = (i_1, \ldots, i_{2H+1}) \\ &\quad D(k) = (j_1, \ldots, j_{2H+1}), Z(k-1))\end{aligned}$$

in terms of $z(k)$ only (so that the resultant filter is recursive).

We first write

$$\begin{aligned}&p(z(k) \mid S(k) = (i_1, \ldots, i_{2H+1})\\ &\quad D(k) = (j_1, \ldots, j_{2H+1}), Z(k-1))\\ &\qquad = \int_{R^{(2H+1)}} p(z(k), T(k) \mid S(k) = (i_1, \ldots, i_{2H+1})\\ &\qquad\quad D(k) = (j_1, \ldots, j_{2H+1}), Z(k-1))\, dT(K)\end{aligned}$$

but using the independence properties of the processes, this reduces to

$$\begin{aligned}&\int_{R^{(2H+1)}} p(z(k) \mid T(k), S(k) = (i_1, \ldots, i_{2H+1}))\\ &\quad \times p(T(k) \mid D(k) = (j_1, \ldots, j_{2H+1}))\, dT(k)\\ &\qquad = \int_{R^{(2H+1)}} p(z(k) \mid T(k), S(k) = (i_1, \ldots, i_{2H+1}))\\ &\qquad\quad \times \psi_{j_1}(\tau(k+H)) \cdots \psi_{j_{2h+1}}(\tau(k-H))\, dT(k).\end{aligned}$$

Now, suppose that $T(k) = [\tau(k+H), \ldots, \tau(k-H)]^T = (\tau_1, \ldots, \tau_{2H+1})$. The problem of determining $p(z(k) \mid T(k) = (\tau_1, \ldots, \tau_{2H+1}), S(k) = (i_1, \ldots, i_{2H+1}))$ is exactly the same problem as determining

$$p(z(k) \mid D(k) = (j_1, \ldots, j_{2H+1}), S(k) = (i_1, \ldots, i_{2H+1}))$$

when the delay was modeled as a finite-state Markov chain. On the basis of the relative departure time and the delay values, we can determine which measurement is received as $z(k)$ and then use the corresponding component of $S(k)$ to determine the likelihood of observing $z(k)$.

Mathematically, we define the function

$$f : R^{(2H+1)} \to \{e_1, e_2, \ldots, e_{2H+1}\}$$

where

$$f((\tau_1, \ldots, \tau_{2H+1})) = e_j^T$$

if

$$\sum_{l=1}^{2H+1} \mathrm{sign}((\tau_j - \tau_l) - (j-l)\Delta) = 0.$$

We then have

$$\begin{aligned}&p(z(k) \mid T(k), S(k) = (i_1, \ldots, i_{2H+1}))\\ &\quad = \phi(z(k) - \Gamma(S(k), D(k)))\end{aligned}$$

where

$$\begin{aligned}&\Gamma(S(k), D(k))\\ &\quad = \int_{R^{(2H+1)}} f(T(k)) \left[g_{s(k+H)}, \ldots, g_{s(k-H)}\right]^T\\ &\quad\quad \times \psi_{j_1}(\tau(k+H)) \cdots \psi_{j_{2h+1}}(\tau(k-H))\, dT(k) \qquad (18)\end{aligned}$$

and we remember that ϕ is the density of the observation noise.

Remark: For the finite-state delay model, (2) was required to ensure that the function f is a well-defined mapping. In the continuous state case, no such condition can be given; however, we note that the probability of two or more packets arriving simultaneously is zero so that the set on which f is a not well defined has probability zero. Strictly speaking, f should be redefined over this region; however, we simply note that this can be done and that the change would not affect our development.

In summary, we have the following theorem.

Theorem 2: The recursion for the conditional state probabilities is given by

$$\begin{aligned}&\alpha_k(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1})\\ &\quad = C\phi(z(k) - \Gamma((i_1, \ldots, i_{2H+1}), (j_1, \ldots, j_{2H+1})))\\ &\quad\quad \times \sum_{m_1} \cdots \sum_{m_{2H+1}} \sum_{n_1} \cdots \sum_{n_{2H+1}}\\ &\quad\quad \left(a_{m_1 i_1} \cdot \delta(i_2 - m_1) \cdots \delta(i_{2H+1} - m_{2H})\right)\\ &\quad\quad \times \left(b_{n_1 j_1} \cdot \delta(n_2 - j_1) \cdots \delta(n_{2H+1} - j_{2H})\right)\\ &\quad\quad \times \alpha_{k-1}(m_1, \ldots, m_{2H+1}, n_1, \ldots, n_{2H+1}) \qquad (19)\end{aligned}$$

for $k \geq 1$ and $(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1}) \in \{1, \ldots, M\}^{(2H+1)} \times \{1, \ldots, N\}^{(2H+1)}$, where $\Gamma(\cdot)$ is defined in (18), and C is a normalization constant. The initial conditions for the recursion are given by

$$\begin{aligned}&\alpha_0(i_1, \ldots, i_{2H+1}, j_1, \ldots, j_{2H+1})\\ &\quad = \sigma_0(i_1, \ldots, i_{2H+1})\pi_0(j_1, \ldots, j_{2H+1})\end{aligned}$$

for $i_1, \ldots, i_{2H+1} \in \{1, \ldots, M\}$ and $j_1, \ldots, j_{2H+1} \in \{1, \ldots, N\}$.

Marginal distributions follow by summing over the appropriate indices as in Section IV.

Computation: We note that the only difference between the filters of Theorem 1 (finite state delay) and Theorem 2 (mixture delay) is in the definition of the observation weighting terms (11) and (18), respectively. Since these functions can be calculated off-line, the (on-line) computational requirements of both filters are identical. We also note that the lower computational complexity algorithm of Section VI applies directly to the mixture delay model simply by using the modified weighting term of (18).

Before proceeding, we note that the results for the finite-state delay model can be obtained from the results for the mixture delay model by setting the densities to Dirac delta functions. In particular, we can set

$$\psi_1(\tau) = \delta(\tau - h_1), \ldots, \psi_N(\tau) = \delta(\tau - h_N)$$

to obtain (11) from (18) and (13) from (19).

VI. Reduced Complexity Algorithm via Aggregation

A clear problem with the optimal filtering algorithms of Theorems 1 and 2 is the computational requirement for systems with more than a few states. We have seen in the comments after these theorems that the algorithms require a total of $O(M^{2H+2}N^{2H+2})$ computations at each time. Thus, with H fixed (recall that H measures the maximum delay spread), the computational load grows polynomially in M and N. With M and N fixed, the computational load grows exponentially in H. Thus, we see that the real difficulty occurs when H becomes large—growth in M and N is not nearly as big a problem.

Even for systems with a large augmented state space, all is not lost. Importantly, the optimal filtering algorithms provide a framework for developing efficient implementations of the optimal filters or practical suboptimal algorithms based on the optimal filter structure. Below, we discuss one technique for reducing the computational load of the optimal algorithm by aggregating the HMM to a lower dimension HMM.

Aggregating the HMM [see (7), (9), and (12)] requires aggregation of the Markov chain and output symbol probabilities. We consider the these two steps in the following subsections.

A. State Aggregation of Markov Chain

For convenience, denote the transition matrices of the Markov chains $S(k)$ and $D(k)$ as A^S and A^D, respectively. These are stochastic matrices of dimension $M^{2H+1} \times M^{2H+1}$ and $N^{2H+1} \times N^{2H+1}$, respectively [see (4) and (7)].

We will first show that both $S(k)$ and $D(k)$ enjoy the property of being exactly aggregatable Markov chains. This means that these Markov chains can be exactly aggregated to lower dimensional Markov chains. This result follows because of the way that $S(k)$ and $D(k)$ were constructed from the original Markov chains $s(k)$ and $d(k)$.

We adopt the following definition from [5, Def. 6.3.1, p. 124]: "A Markov chain is exactly aggregatable (in [5] the term 'lumpable' is used instead of strictly aggregatable) with respect to some partition of the state space, if for every *a priori* state probability vector π, the aggregated process is a Markov chain, and the transition probabilities do not depend on the choice of π."

The following theorem is proved in [5, Th. 6.3.2, p. 124]: Let $X(k)$ denote a M state Markov chain with transition probability matrix A^X. Let $\theta < M$ denote a positive integer.

Result 3: A necessary and sufficient condition for $X(k)$ to be exactly aggregatable with respect to a partition $I = (I_1, \ldots, I_\theta)$ of the state space is that for every pair of sets I_i and I_j, $P(X(k+1) \in I_j \mid X(k) = l)$ has the same value for every state l in I_i, $i = 1, \ldots, \theta$ for every $j = 1, \ldots, \theta$. These common values form the transition matrix of the aggregated chain.

Based on the above result, it is straightforward to verify that A^S and A^D defined in (4) and (7) are exactly aggregatable. For example, consider the Markov chain $S(k)$ [which is exactly the same procedure can be applied to $D(k)$ with M replaced by N].

The Markov chain $S(k)$ can be exactly aggregated in several possible ways. Let θ denote an integer such that $0 \leq \theta \leq 2H$. Then, for each such θ, we can aggregate $S(k)$ into the $M^{\theta+1}$ state Markov chain $\bar{S}_k$, where

$$\bar{S}(k) = [s(k+H), \ldots, s(k+H-\theta)]^T \quad 0 \leq \theta \leq 2H.$$

Each aggregated state $\bar{S}(k) = (i_1, \ldots, i_{\theta+1})$ contains $M^{2H-\theta}$ states of $S(k)$, namely

$$(i_1, \ldots, i_{\theta+1}, i_{\theta+2}, \ldots, i_{2H+1})$$

where each index $i_{\theta+2}$, $i_{\theta+3}$, etc., takes on values in $\{1, \ldots, M\}$.

TABLE II
ENUMERATION OF STATES OF $S(k)$ AND $\bar{S}(k)$

$s(k+1)$	$s(k)$	$s(k-1)$	$S(k)$	$\bar{S}_k$ $(\theta=1)$	$\bar{S}_k$ $(\theta=0)$
1	1	1	1	1	1
1	1	2	2		
1	2	1	3	2	
1	2	2	4		
2	1	1	5	3	2
2	1	2	6		
2	2	1	7	4	
2	2	2	8		

Using Result 3, the transition probabilities of the aggregated chain $\bar{S}(k)$ are

$$\begin{aligned} &P(\bar{S}(k+1) = (j_1, \ldots, j_{\theta+1}) \mid \bar{S}(k) = (i_1, \ldots, i_{\theta+1})) \\ &\quad = a_{i_1 j_1} \cdot \delta(j_2 - i_1) \cdots \delta(j_{\theta+1} - i_\theta). \end{aligned}$$

The following example illustrates our notation and the aggregation process when $M = 2$ and $H = 1$. Using the state enumerations given in Table II, we have

$$A^S = \begin{pmatrix} a_{11} & 0 & 0 & 0 & a_{12} & 0 & 0 & 0 \\ a_{11} & 0 & 0 & 0 & a_{12} & 0 & 0 & 0 \\ 0 & a_{11} & 0 & 0 & 0 & a_{12} & 0 & 0 \\ 0 & a_{11} & 0 & 0 & 0 & a_{12} & 0 & 0 \\ 0 & 0 & a_{21} & 0 & 0 & 0 & a_{22} & 0 \\ 0 & 0 & a_{21} & 0 & 0 & 0 & a_{22} & 0 \\ 0 & 0 & 0 & a_{21} & 0 & 0 & 0 & a_{22} \\ 0 & 0 & 0 & a_{21} & 0 & 0 & 0 & a_{22} \end{pmatrix}.$$

Now, with reference to Table II, we see that choosing $\theta = 1$ means $\bar{S}(k) = [s(k+1), s(k)]^T$ and leads to the aggregated transition probability matrix

$$\begin{pmatrix} a_{11} & 0 & a_{12} & 0 \\ a_{11} & 0 & a_{12} & 0 \\ 0 & a_{21} & 0 & a_{22} \\ 0 & a_{21} & 0 & a_{22} \end{pmatrix}.$$

With $\theta = 0$, we have $\bar{S}_k = s(k+1)$ and corresponding aggregated transition probability matrix

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

B. Symbol Probability Aggregation

The aim here is to aggregate the symbol probabilities $P(z(k) \mid S(k), D(k))$ of the HMM, i.e., compute $P(z(k) \mid \bar{S}(k), \bar{D}(k))$.

For notational convenience, let $\bar{M} = M^{\theta+1}$ denote the number of states of the aggregated chain $\bar{S}(k)$. Similarly, let $\bar{N} = N^{\theta+1}$ denote the number of states of the aggregated chain $\bar{D}(k)$. In addition, let $I_1, \ldots, I_{\bar{M}}$ (resp. $J_1, \ldots, J_{\bar{N}}$) denote the partition of the state space of S_k [resp. $D(k)$], which results in the $\bar{M}$ (resp. $\bar{N}$) state aggregated process $\bar{S}(k)$ [resp. $\bar{D}(k)$]. We will use the indices $\bar{m} \in \{1, \ldots, \bar{M}\}$

and $\bar{n} \in \{1, \ldots, \bar{N}\}$ to denote the states of $\bar{S}(k)$ and $\bar{D}(k)$. Note that each subset $I_{\bar{m}}$ consists of $M^{2H-\theta}$ states of $S(k)$.

With $\bar{m} = (i_1, \ldots, i_{\theta+1})$ and $\bar{n} = (j_1, \ldots, j_{\theta+1})$, define the aggregated symbol probabilities $\bar{\phi}_{\bar{m},\bar{n}}$ as

$$\bar{\phi}_{\bar{m},\bar{n}}(z(k)) = P(z(k) \mid \bar{S}(k) = \bar{m}, \bar{D}(k) = \bar{n}).$$

Finally, let $m = (i_1, \ldots, i_{2H+1})$ and $n = (j_1, \ldots, j_{2H+1})$ denote the states of $S(k)$ and $D(k)$, respectively.

We assume that $s(k)$ and $d(k)$ are irreducible Markov chains so that $S(k)$ and $D(k)$ are also irreducible Markov chains. Thus, there are unique steady-state probability distributions π^A and π^D [which are known as the Perron Frobenius (PF) eigenvectors of A^S and A^D, respectively] satisfying

$$(A^S)^T \pi^S = \pi^S, \quad \pi_i^S > 0, \quad \sum_i \pi_i^S = 1$$
$$(A^D)^T \pi^D = \pi^D, \quad \pi_i^D > 0, \quad \sum_i \pi_i^D = 1.$$

Lemma 4: The (approximate) aggregated symbol probabilities are computed as

$$\bar{\phi}_{\bar{m},\bar{n}}(z(k)) = \frac{\sum_{m \in I_{\bar{m}}} \sum_{n \in I_{\bar{n}}} \pi_m^S \pi_n^D \phi(z(k) - \Gamma(m,n))}{\sum_{m \in I_{\bar{m}}} \sum_{n \in I_{\bar{n}}} \pi_m^S \pi_n^D}. \tag{20}$$

Proof:

$$\begin{aligned} P(z(k) \mid \bar{S}(k) = I_{\bar{m}}, \bar{D}(k) = J_{\bar{n}}) &= P(z(k) \mid S(k) \in I_{\bar{m}}, D(k) \in I_{\bar{n}}) \\ &= \frac{P(z(k), S(k) \in I_{\bar{m}}, D(k) \in I_{\bar{n}})}{P(S(k) \in I_{\bar{m}}, D(k) \in I_{\bar{n}}).} \end{aligned}$$

The numerator is equal to

$$\sum_{m \in I_{\bar{m}}} \sum_{n \in I_{\bar{n}}} P(S(k) = m) P(D(k) = n) \times P(z(k) \mid S(k) = m, D(k) = n)$$

and the denominator equals

$$\sum_{m \in I_{\bar{m}}} \sum_{n \in I_{\bar{n}}} P(S(k) = m) P(D(k) = n).$$

The last expression above converges with probability 1 to (20) since $S(k)$ and $D(k)$ are assumed irreducible. □

We note that (20) is exact only when the distributions of $S(k)$ and $D(k)$ have converged to their stationary distributions. The aggregated symbol probabilities could be calculated exactly using model update equations for $S(k)$ and $D(k)$ (instead of the limiting distribution); however, this would require $O(M^{2H+2} N^{2H=2})$ computations at each step, thus eliminating any possible computational savings.

The aggregated symbol probabilities $\bar{\phi}(\cdot)$ of (20) are weighted sums of the unaggregated symbol probabilities $\phi(\cdot)$. The weighting coefficients are the components of the PF eigenvectors π^S and π^D. Since A^S and A^D are known, π^S and π^D can be computed off-line. If A^S and A^D have very large dimensions, then the computation of π^S and π^D can be simplified by using numerical algorithms such as stochastic complementation [6].

C. Reduced-Complexity Filters and Smoothers

Using the aggregation procedure outlined in Sections VI-A and B, the $M^{2H+1} N^{2H+1}$ state HMM can be aggregated into a $\bar{M}\bar{N}$ state HMM, which we will denote as $\bar{\lambda} = (A^{\bar{S}}, A^{\bar{D}}, \bar{\phi})$.

Reduced-Complexity Filters: The filtered density for the states $\bar{S}(k)$ and $\bar{D}(k)$ of this aggregated HMM is defined as

$$\begin{aligned} \bar{\alpha}_k(\bar{m}, \bar{n}) &= P_{\bar{\lambda}}(\bar{S}(k) = \bar{m}, \bar{D}(k) = \bar{n} \mid Z(k)) \\ &= P_{\bar{\lambda}}(\bar{S}(k) = (i_1, \ldots, i_{\theta+1}) \\ &\qquad \bar{D}(k) = (j_1, \ldots, j_{\theta+1}) \mid Z(k)). \end{aligned}$$

The subscript $\bar{\lambda}$ emphasizes that the filtered density $\bar{\alpha}_k$ is with respect to the aggregated model.

$\bar{\alpha}_k(\bar{m}, \bar{n})$ is computed similarly to Theorem 1 as

$$\begin{aligned} &\bar{\alpha}_{k+1}(\bar{p}, \bar{q}) \\ &\quad = \bar{C} \sum_{\bar{m}} \sum_{\bar{n}} \bar{\alpha}_k(\bar{m}, \bar{n}) A^{\bar{S}}(\bar{m}, \bar{p}) A^{\bar{D}}(\bar{n}, \bar{q}) \bar{\phi}_{\bar{p},\bar{q}}(z(k+1)) \end{aligned}$$

where $\bar{C}$ is a normalization constant. Note that we are implicitly making the approximation

$$\begin{aligned} &P(z(k+1) \mid \bar{S}(k+1) = \bar{p}, \bar{D}(k+1) = \bar{q}, Z(k)) \\ &\quad = P(z(k+1) \mid \bar{S}(k+1) = \bar{p}, \bar{D}(k+1) = \bar{q}) \\ &\quad = \bar{\phi}_{\bar{p},\bar{q}}(z(k+1)) \end{aligned}$$

which is the reason the above expression is not exact.

Finally, the filtered state probabilities for the original Markov chains $s(k+H-\theta)$ and $d(k+H-\theta)$ can be computed as

$$P_{\bar{\lambda}}(s(k+H-\theta) = i_{\theta+1} \mid Z(k)) = \sum_{i_1} \cdots \sum_{i_\theta} \sum_{\bar{n}} \bar{\alpha}_k(\bar{m}, \bar{n})$$
$$P_{\bar{\lambda}}(d(k+H-\theta) = j_{\theta+1} \mid Z(k)) = \sum_{\bar{m}} \sum_{j_1} \cdots \sum_{j_\theta} \bar{\alpha}_k(\bar{m}, \bar{n}).$$

The computational complexity of this reduced-complexity filter is $O(\bar{M}^2 \bar{N}^2)$ or $O(M^{2\theta+2} N^{2\theta+2})$ at each time instant.

Reduced-Complexity Smoothers: In order to compute $P_{\bar{\lambda}}(s(k) \mid Z(k))$ and $P_{\bar{\lambda}}(d(k) \mid Z(k))$, it is necessary to use a fixed-lag smoother as follows.

For a fixed lag G (where G is a positive integer), define the fixed-lag smoothed density of the aggregated HMM $\bar{\lambda}$ as

$$\bar{\gamma}_{k|k+G}(\bar{m}, \bar{n}) = P_{\bar{\lambda}}(\bar{S}(k) = \bar{m}, \bar{D}(k) = \bar{n} \mid Z(k+G))$$

where $\bar{\gamma}_{k|k+G}$ is easily computed in terms of the filtered density $\bar{\alpha}_k$ as

$$\bar{\gamma}_{k|k+G}(\bar{m}, \bar{n}) = \frac{\bar{\alpha}_k(\bar{m}, \bar{n}) \bar{\beta}_{k|k+G}(\bar{m}, \bar{n})}{\sum_{\bar{m}} \sum_{\bar{n}} \bar{\alpha}_k(\bar{m}, \bar{n}) \bar{\beta}_{k|k+G}(\bar{m}, \bar{n})}.$$

Here

$$\begin{aligned} \bar{\beta}_{k|k+G}(\bar{m}, \bar{n}) &= P_{\bar{\lambda}}(z(k+G), z(k+G-1), \ldots \\ &\qquad z(k+1) \mid \bar{S}(k) = \bar{m}, \bar{D}(k) = \bar{n}) \end{aligned}$$

is computed via the backward recursion

$$\begin{aligned} \bar{\beta}_{k+t|k+G}(\bar{m}, \bar{n}) &= \sum_{\bar{p}} \sum_{\bar{q}} \bar{\beta}_{k+t+1|k+L}(\bar{p}, \bar{q}) A^{\bar{S}}(\bar{m}, \bar{p}) \\ &\quad \times A^{\bar{D}}(\bar{n}, \bar{q}) \bar{\phi}_{\bar{p},\bar{q}}(z(k+t)) \\ &\qquad t = G-1, G-2, \ldots, 0 \end{aligned}$$

beginning from $\bar{\beta}_{k+G|k+G}(\bar{m}, \bar{n})) = 1$ for all $\bar{m}$ and $\bar{n}$.

Computing $\bar{\beta}_{k|k+G}$ involves $O(GM^{2\theta+2}N^{2\theta+2})$ computations. Hence, the cost involved in computing the fixed lag estimate $\bar{\gamma}_{k|k+G}$ (including computing $\bar{\alpha}_k$ and $\bar{\beta}_k$) at each time instant k is $O(GM^{2\theta+2}N^{2\theta+2})$.

Note that $P(s(k) \mid Z(k))$ is computed using the above fixed lag smoother with a lag $G = H - \theta$. Thus, if we choose $\theta = 0$, say, the overall computational load of the reduced-complexity filter is $O(HM^2N^2)$.

Remark: For details on computationally efficient smoothers for HMM's including saw-tooth lag smoothers, see [7].

VII. Numerical Studies

In this section, we study the performance of the optimal state estimation (filtering) algorithm and the reduced computational complexity algorithm for the finite-state delay model through some numerical examples.

The system we consider has an underlying sampling time of $\Delta = 1$ unit. With reference to the parameters defined in Section II, we have

$$M = 4, \quad A = \begin{bmatrix} 0.8 & 0.1 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.8 & 0.1 \\ 0.05 & 0.05 & 0.1 & 0.8 \end{bmatrix}, \quad g = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \end{bmatrix}$$

for the Markov chain we wish to estimate $s(k)$, and

$$N = 3, \quad B = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.25 & 0.5 & 0.25 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, \quad h = \begin{bmatrix} 0.25 \\ 0.75 \\ 1.5 \end{bmatrix}$$

for the delay model $d(k)$. The three delay states model a simple situation where we have light, medium, and heavy levels of network congestion. While finite-state delay models with more states or the mixture delay model could also readily be treated, the three-state model has the advantage of low computation time as well as allowing the pertinent features of the filtering algorithms to be clearly illustrated. Larger systems are readily handled using the reduced computational complexity algorithms of Section VI.

For this example, we have $H = 1$, and thus,

$$S(k) = [s(k+1), s(k), s(k-1)]^T$$

is a 64-state Markov chain, and

$$D(k) = [d(k+1), d(k), d(k-1)]^T$$

is a 27-state Markov chain.

The transition probability matrices for $S(k)$ and $D(k)$ are formed in terms of the the transition probabilities of $s(k)$ following the procedure of Section VI-A. When looking at the reduced complexity algorithm of Section VI, we use $\theta = 0$ so that $\bar{S}(k) = s(k+1)$ and $\bar{D}(k) = d(k+1)$ (see Section VI-A).

The sensor observation noise $w(k)$ is assumed to be Gaussian with zero mean and variance σ_w^2 so that

$$\phi(w) = (2\pi)^{-1/2}\sigma_w^{-1} \exp\left(-w^2/2\sigma_w^2\right).$$

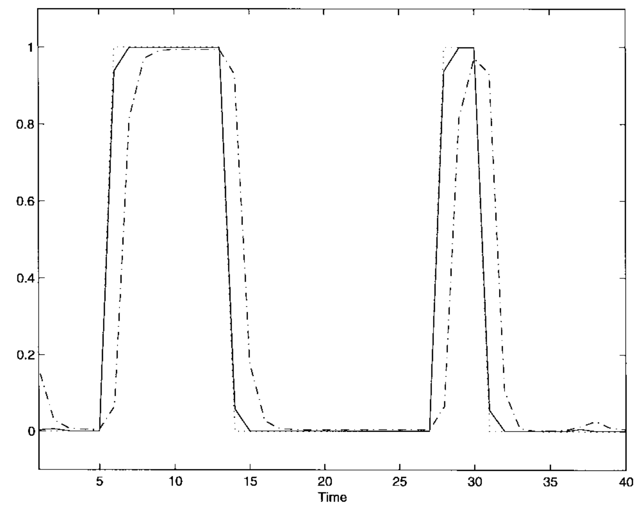


Fig. 3. Sample path of optimal and suboptimal filtered state probabilities. Low noise [optimal (solid), suboptimal (dash-dot), true (dashed)].

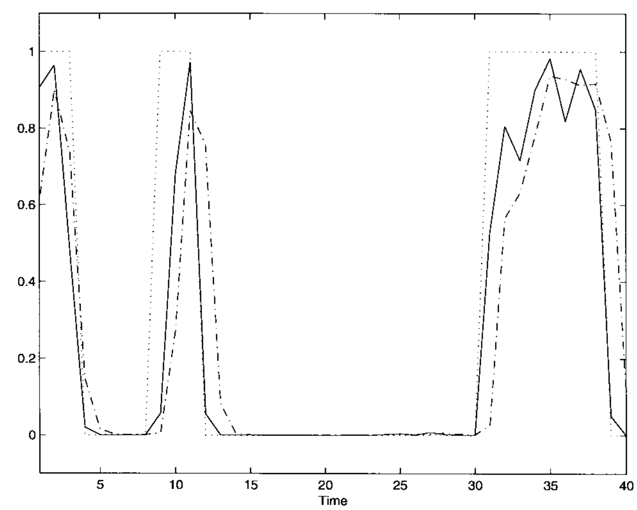


Fig. 4. Sample path of optimal and suboptimal filtered state probabilities. Medium noise [optimal (solid), suboptimal (dash-dot), true (dashed)].

We consider situations with low noise ($\sigma_w = 0.1$), medium noise ($\sigma_w = 0.4$), and high noise ($\sigma_w = 1.0$).

Figs. 3–5 show 40 points of sample paths of the true state and optimal and suboptimal estimates for each of the observation noise levels.

We calculate and plot conditional probability estimates based on the processor observations. The optimal filtered state estimates $P(s(k) = i \mid Z(k))$ are calculated as in Section IV. We also show the state estimate calculated using the reduced-complexity algorithm of Section VI-C with $\theta = 0$.

The plots show the exact or approximate conditional probability that $s(k) = 2$ along with the indicator function for this event (the curve labeled true in the plots). The conditional probabilities of $s(k) = i$, $i \neq 2$ show similar behavior.

We observe that the suboptimal filter performs well for all levels of observation noise. In all cases, the bandwidth of the suboptimal filter is smaller than the optimal filter in the sense that suboptimal filter responds less quickly to the observations.

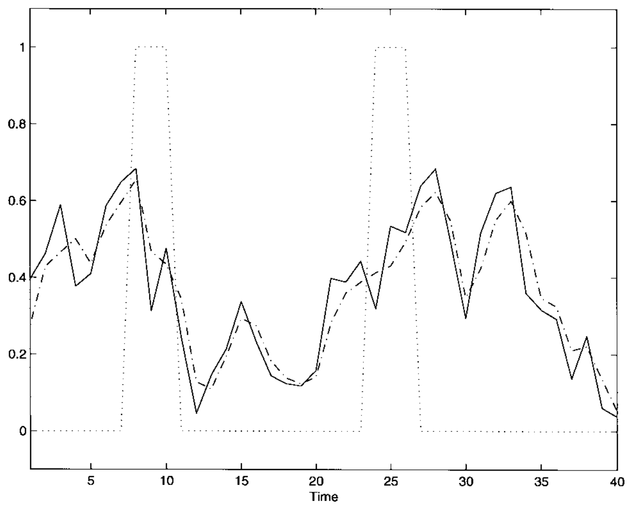


Fig. 5. Sample path of optimal and suboptimal filtered state probabilities. High noise [optimal (solid), suboptimal (dash-dot), true (dashed)].

This gives the filtered probability sample path a smoother appearance than the optimal filter response.

VIII. Conclusion

This paper has addressed the HMM state estimation problem in an environment where state measurements are sent from the sensor to the processing element over a connectionless, packet-switched network. The random delay introduced by the network is modeled as both a finite-state Markov chain and as a continuous-valued process with density indexed by the state of a finite-state Markov chain. In both cases, we derive a finite-dimensional recursion for the conditional (filtered) state probabilities based on a reformulation of the problem in terms of augmented state Markov chains.

Future work will examine the analagous filtering problem for linear-Gaussian systems and the extension of the filtering results to the partially observed stochastic control framework. We remark briefly here that while the linear-Gaussian case can be formulated using the state augmentation techniques used in this paper, the resultant state filter requires a cost in computation and memory that grows exponentially with the length of the data. The resultant filter has a similar structure to those for hybrid Markov systems such as Markov jump linear systems [8], [9]. There is interest is in developing efficient suboptimal filtering algorithms for this case.

References

[1] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[2] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. New York: Springer-Verlag, 1995.

[3] M. Schwartz, *Telecommunications Networks: Protocols, Modeling and Analysis*. Reading, MA: Addison-Wesley, 1987.

[4] J. Nilsson and B. Bernhardsson, “LQG control over a Markov communication network,” in *Proc. IEEE Conf. Decision Contr.*, San Diego, CA, Dec. 1997, pp. 4586–4591.

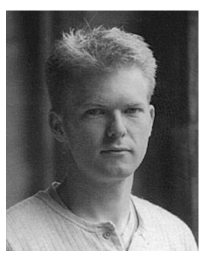
[5] J. Kemeny and J. Snell, *Finite Markov Chains*. New York: Van Nostrand, 1960.

[6] C. Meyer, “Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems,” *SIAM Rev.*, vol. 31, pp. 240–272, 1989.

[7] V. Krishnamurthy and J. Moore, “On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure,” *IEEE Trans. Signal Processing*, vol. 41, pp. 2557–2573, Aug. 1993.

[8] J. Tugnait, “Adaptive estimation and identification for discrete systems with Markov jump parameters,” *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 1054–1065, Oct. 1982.

[9] H. A. P. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Trans. Automat. Contr.*, vol. 33, pp. 780–783, Aug. 1988.



Jamie S. Evans was born in Australia in 1970. He received the B.Sc. degree in physics and the B.E. degree in computer engineering from the University of Newcastle, Callaghan, Australia, in 1992 and 1993, respectively. He then moved to the University of Melbourne, Parkville, Australia, where he received the M.Eng.Sc. degree in 1996 and the Ph.D. degree in 1998.

From March 1998 until June 1999, he worked as a Postdoctoral Fellow with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. He is now with the School of Electrical and Information Engineering, University of Sydney, Sydney, Australia. His interests are in stochastic filtering and control theory and in the design and analysis of wireless communications networks.

Vikram Krishnamurthy (SM’99), for a photograph and biography, please see this issue, p. 2156.