Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors

Vikram Krishnamurthy, *Senior Member, IEEE*

***Abstract*—Consider a Hidden Markov model (HMM) where a single Markov chain is observed by a number of noisy sensors. Due to computational or communication constraints, at each time instant, one can select only one of the noisy sensors. The sensor scheduling problem involves designing algorithms for choosing dynamically at each time instant which sensor to select to provide the next measurement. Each measurement has an associated measurement cost. The problem is to select an optimal measurement scheduling policy to minimize a cost function of estimation errors and measurement costs. The optimal measurement policy is solved via stochastic dynamic programming. Sensor management issues and suboptimal scheduling algorithms are also presented. A numerical example that deals with the aircraft identification problem is presented.**

***Index Terms*—Hidden Markov models, partially observed Markov decision processes, sensor scheduling, stochastic dynamic programming.**

I. INTRODUCTION

IN many signal processing applications, several types of sensors are available for measuring a given process. However, physical and computational constraints often impose the requirement that at each time instant, one is able to use only one out of a possible total of L sensors. In such cases, one has to make the following decision: *Which sensor (or mode of operation) should be chosen at each time instant to provide the next measurement*? Typically associated with each type of measurement is a per unit-of-time measurement cost, reflecting the fact that some measurements are more costly or difficult to make than others, although they may contain more useful or reliable information. The problem of optimally choosing which one of the L sensor observations to pick at each time instant is called the *sensor scheduling problem*. The resulting time sequence that, at each instant, specifies the best sensor to choose is termed the *sensor schedule sequence*.

Several papers have studied the sensor scheduling problem for systems with linear Gaussian dynamics, where linear measurements in Gaussian noise are available at a number of sensors (see [2] for the continuous-time problem and [16] for the discrete-time problem). For such linear Gaussian systems, if the cost function to be minimized is the state error covariance (or some other quadratic function of the state), then the solution has a nice form: The optimal sensor schedule sequence can be determined *a priori* and is *independent* of the measurement data (see [2], [16] for details). This is not surprising since the Kalman filter state covariance estimate (given by the Riccati equation) is independent of the observation sequence and only depends on the model parameters.

In this paper, we study the discrete-time sensor scheduling problem for hidden Markov model (HMM) sensors. We assume that the underlying process is a finite state Markov chain. At each time instant, observations of the Markov chain in white noise are made at L different sensors. However, only one sensor observation can be chosen at each time instant. The aim is to devise an algorithm that optimally picks which single sensor to use at each time instant, in order to minimize a given cost function. The cost function is comprised of the sensor usage costs, together with sensor estimation errors.

There are numerous applications of the optimal scheduling of sensors. Some applications include [5], [9], [16]

1) finding the optimum channel allocation among various components of a measurement vector when they must be transmitted over a time-shared communication channel of limited bandwidth;
2) finding the optimum timing of measurements when the number of possible measurements is limited because of energy constraints;
3) finding the optimum tradeoff between measurement of range and range rate in radar systems with a given ambiguity function.

There is also growing interest in flexible sensors such as multimode radar, which can be configured to operate in one of many modes for each measurement. Unlike the linear Gaussian case, HMM sensor scheduling is more interesting because the optimal sensor schedule in the HMM case is data dependent. This means that past observations, together with past choices of sensors, influence which sensor to choose at present.

In our recent work [9], we formulated the HMM sensor scheduling problem and presented the dynamic programming functional recursion for determining the optimal sensor schedule. However, the dynamic programming equations (Bellman's equation) in [9] do not directly translate into practical solution methodologies. The fundamental problem is that at each time instant of the dynamic programming recursion, one needs to compute the dynamic programming cost over an

Manuscript received June 11, 2001; revised February 22, 2002. This work was supported in part by the Australian Research Council and the Center of Expertise in Networked Decision Systems and the Department of Signals Sensors and Systems, Royal Institute of Technology, Sweden. Parts of this work were presented at ICASSP 2001. The associate editor coordinating the review of this paper and approving it for publication was Prof. Xiaodong Wang.

The author 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(02)04384-2.

uncountably infinite set. An approximate heuristic algorithm was presented in [9], which was based on discretizing the dynamic programming recursion to a finite grid.

The main contribution of this paper are as follows.

1) By exploiting and extending results and algorithms in partially observed Markov decision processes (POMDPs), near optimal *finite dimensional* algorithms for HMM sensor scheduling are presented. Unlike the grid-based approximate solution presented in [9], we determine a closed-form finite dimensional solution to the dynamic programming recursion. The finite-dimensional scheduling algorithms presented in this paper are similar to those recently used in the operations research (see [14] for a tutorial survey) and in robot navigation systems [5] for the optimal control of the POMDP. However, our problem has the added complexity that the cost function is, in general, a nonlinear function (e.g., quadratic function) of the information state, whereas in all the existing literature on POMDPs, only costs that are linear in the information state are considered.
2) We present optimal dynamic programming-based HMM sensor scheduling algorithms that deal with constraints. These include sensor estimation error constraints and sensor management issues that require sensor usage constraints. In addition, steady-state HMM scheduling algorithms are presented.
3) For medium to large size problems, the computational complexity of the optimal dynamic programming-based algorithms are prohibitive. We present two suboptimal HMM scheduling algorithms: Lovejoy's algorithm and a one-step look-ahead algorithm.
4) In Section II-B, we briefly describe an application of the HMM sensor scheduling problem in the tradeoff between prediction and filtering. In Section VI, we illustrate the performance of the optimal HMM sensor scheduling algorithms proposed in this paper to an aircraft identification problem. The scenario involves an incoming aircraft, where using various forms of sensors available at a base station, the task is to determine if the aircraft is a threat or not [5]. The choice of deciding between various sensors arises because the better sensors tend to make the location of the base station more easily identifiable or visible to the aircraft, whereas the more stealthy sensors tend to be more inaccurate.

Limitations: Although the optimal algorithms we present for computing the solution to the HMM dynamic programming recursion are offline (i.e., independent of the observations), a major limitation of such algorithms is that they are PSPACE hard; they have exponential computational complexity [14, p. 55]. While these algorithms were deemed too expensive in the 1970s and early 1980s, there has been increasing use of these methods in the last five years, particularly in the artificial intelligence and robotics communities; see [5] and [6]. It is shown in the numerical examples of Section VI that on a Pentium 2 personal processor, the algorithms are feasible for up to six states, three sensors, and six observation types in that the algorithms require in the order of several minutes to compute the optimal sensor schedule. The recent thesis of [5] demonstrates that these algorithms can be applied to problem sizes of up to 15 states and observations. For larger problems, the time taken can be excessive (several hours or more); for these cases, the suboptimal algorithms presented are useful.

Existing Works: Here is a brief survey of the literature in sensor scheduling and HMM control. The general problem of stochastic control of POMDPs, i.e., stochastic control of hidden Markov hodels, is treated in [17], [20], [14] (discrete-time), and [19] (continuous-time), as well as in the standard texts [4] and [12]. The website [6] and Ph.D. theses [5] and [11] contain excellent expositions and up-to-date references to the literature on POMDPs with an emphasis on AI robot navigation problems. A related problem to HMM sensor scheduling is the problem of optimizing the observer motion in bearings-only target tracking. The recent work in [21] formulates such a problem as a POMDP and seeks to optimize a functional of the Fisher information matrix that measures the information in measurements relative to target trajectory.

II. Signal Model and Problem Formulation

Let $k = 0, 1, \ldots$ denote discrete time. Assume that X_k is an S-state Markov chain with state space $\{e_1, \ldots, e_S\}$. Here, e_i denotes the S-dimensional unit vector with 1 in the ith position and zeros elsewhere. This choice of using unit vectors to represent the state space considerably simplifies our subsequent notation. Define the $S \times S$ transition probability matrix A as

$$A = [a_{ji}]_{S\times S} \text{ where } a_{ji} = P(X_k = e_i | X_{k-1} = e_j) \\ i, j \in \{1, \ldots, S\}.$$

Denote the initial probability vector π_0 of the Markov chain as

$$\pi_0 = [\pi_0(i)]_{S\times 1} \text{ where } \pi_0(i) = P(X_0 = i), \quad i \in \{1, \ldots, S\}. \tag{1}$$

A. Sensor Scheduling Problem

Assume there are L noisy sensors available that can be used to give measurements of X_k. At each time instant k, we are allowed to pick only one of the L possible sensor measurements. Motivated by the physical and computational constraints alluded to in the introduction, we assume that having picked this sensor, we are not allowed to look at any of the other $L-1$ observations at time k.

Let $u_k \in \{1, \ldots L\}$ denote the sensor picked at time k. The observation measured by this sensor is denoted as $y_k(u_k)$. Suppose at time k, we picked the lth sensor, i.e., $u_k = l$, where $l \in \{1, \ldots, L\}$. Assume that the measurement $y_k(l)$ of the lth sensor belongs to a known finite set of symbols $O_1(l), O_2(l), \ldots, O_{M_l}(l)$. That is, the lth sensor can yield one of M_l possible measurement values at a given time instant. For $u_k \in \{1, \ldots, L\}$, denote the symbol probabilities as

$$b_i(u_k = l, y_k(u_k) = O_m(l)) = P(y_k(u_k) = O_m(u_k) | X_k \\ = e_i, u_k = l), i = 1, 2, \ldots, S.$$

These represent the probability that an output $O_m(l)$ is obtained given that the state of the Markov chain is e_i and that the lth

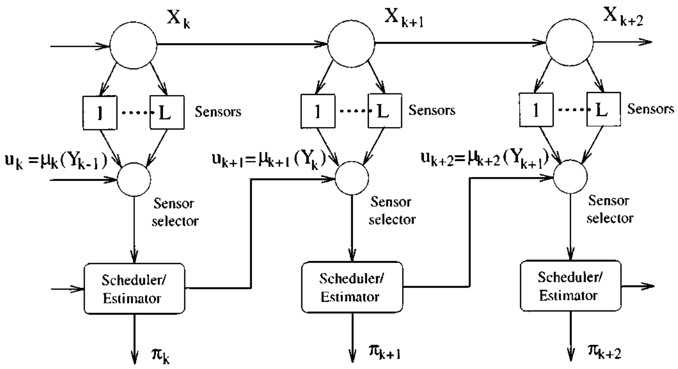


Fig. 1. HMM sensor scheduling and estimation problem.

sensor is chosen. The symbol probabilities are assumed known. Define the symbol probability matrix

$$B(u_k, O_m(u_k)) = \text{diag}\begin{bmatrix} b_1(u_k, O_m(u_k)) \\ \vdots \\ b_S(u_k, O_m(u_k)) \end{bmatrix}.$$

Finally, for notational convenience, let $\phi = (A, B(l, O_m(l))), l = 1, \ldots, L, \quad m = 1, \ldots, M_l$ denote the entire parameter vector that is comprised of the transition probability matrix and all the symbol probabilities of all the L sensors.

Remarks:

1) We have assumed that only one sensor is picked at each time. This is purely for convenience. It is straightforward to generalize the model to picking $\bar{L}$ sensors (where $\bar{L} < L$) at each time instant by merely increasing the dimension of u_k as follows. Define $u_k = (u_k(1), \ldots, u_k(\bar{L}))'$, where each $u_k(i) \in \{1, \ldots, L\}$. Then, the signal model is identical to above.
2) We have allowed for different sensors to have different output symbols and number of quantization levels. This is reflected in our notation by introducing the l in the output symbols $O_m(l)$ and the number of possible symbols M_l for the lth sensor $l \in \{1, \ldots, L\}$.

Let $Y_k = \{u_1, u_2, \ldots, u_k, y_1(u_1), y_2(u_2), \ldots, y_k(u_k)\}$ so that Y_k represents the information available at time k upon which to base estimates and sensor scheduling decisions. The sensor scheduling and estimation problem proceeds in three stages for each $k = 0, 1, \ldots, N-1$, where N is a fixed positive integer (see Fig. 1)

1) Scheduling: Based on Y_k, we generate $u_{k+1} = \mu_{k+1}(Y_k)$, which determines the sensor that is to be used at the next time step.

2) Observation: We then observe $y_{k+1}(u_{k+1})$, where u_{k+1} is the sensor selected in the previous stage.

3) Estimation: After observing $y_{k+1}(u_{k+1})$, we compute the optimal (MMSE) state estimate π_{k+1} of the Markov chain state X_{k+1} as

$$\pi_{k+1} = \mathbb{E}\{X_{k+1}|Y_{k+1}\}. \tag{2}$$

This is done using an HMM state filter [8] (which is also known as the "forward algorithm" [18]). Note that π_k is a column vector of dimension S. Because we have assumed that X_k is a unit vector $\in \{e_1, \ldots, e_S\}$, it straightforwardly follows that the elements of π_k are $\pi_k(i) = P(X_k = e_i|Y_k), i = 1, 2, \ldots, S$. (This property of the conditional expectation, being equal to the conditional probability vector, is one of the notational advantages of depicting the state space by unit vectors). In Section II-C, π_k will be defined as the "information state" of the scheduling problem. Note that the state estimate π_{k+1} is dependent on the scheduling sequence of sensors picked from time 1 to $k+1$, i.e., $u_1, \ldots, u_{k+1}$ (since it depends on Y_{k+1}).

Define the sensor scheduling sequence

$$\mu = \{\mu_1, \mu_2, \ldots, \mu_N\}$$

and say that the scheduling sequences are admissible if μ_{k+1} maps Y_k to $\{1, \ldots, L\}$. Note that μ is a sequence of functions.

We assume the following cost is associated with estimation errors and with the particular sensor schedule chosen. If, based on the observation at time k, the decision is made to choose $u_{k+1} = l$ (i.e., to choose the lth sensor at time $k+1$, where $l \in \{1, \ldots, L\}$), then the instantaneous cost incurred at time k comprises of two terms.

1) **State estimation error**:
$$\alpha_k(l)\,\|X_k - \pi_k\|_D \tag{3}$$
Here, $\alpha_k(l), l = 1, 2 \ldots, L$ are known positive scalar weights. The "distance" function D is assumed to be a convex function with $D : \mathbb{R}^S \to \mathbb{R}$. $\|X_k - \pi_k\|_D$ denotes the state estimation error (with respect to the distance function D) at time k due to choosing the sensor schedule $u_1, \ldots, u_k$. For example, if D is the l_2 norm, then $\|X_k - \pi_k\|_D$ denotes the Euclidean distance between X_k and its estimate π_k. In such a case, the instantaneous cost is the square error in the state estimate when using the sensor u_{k+1}.
2) **Sensor usage cost**: Let $c_k(X_k, l)$ denote the instantaneous cost of using the sensor $u_{k+1} = l$ when the state of the Markov chain is X_k. For example, in target tracking applications, active sensors such as radar are more expensive to use than passive sensors such as sonar.

Our aim is to find the optimal sensor schedule to minimize the total accumulated cost J_μ from time 1 to N over the set of admissible control laws

$$J_\mu = \mathbb{E}\left\{\sum_{k=0}^{N-1} \alpha_k(u_{k+1})\|X_k - \pi_k\|_D + \sum_{k=0}^{N-1} c_k(X_k, u_{k+1}) + \alpha_N\|X_N - \pi_N\|_D\right\} \tag{4}$$

where $u_{k+1} = \mu_{k+1}(Y_k)$.

The above objective (4) can be interpreted as follows. The minimization of the first summation yields the optimal sensor schedule $u_1, \ldots, u_N$ that minimizes the weighted error in the state estimate of the Markov chain state X_k. The weight terms $\alpha_k(l)$ allow different sensors $l \in \{1, 2, \ldots, L\}$ to be be weighed differently. The time index in α_k allows us to weigh the state estimate errors over time. The second summation term reflects the cost involved in using a sensor (i.e., the unit time sensor charge) when the the Markov chain is in a particular state. The final term in (4) is the terminal cost at time N.

Remark: The above problem is a finite horizon partially observed stochastic control problem. Such problems are often plagued by mathematical technicalities. For the cost function in (4) to be well defined, we assume that $\alpha_k(u)$, $c_k(X,u)$ are uniformly bounded from above and below (see [4, p.54]). With this boundedness assumption, together with the fact that S and $M_l, l=1,2,\ldots,L$ are finite sets, the above partially observed stochastic control problem is well defined, and an optimal policy exists; see [4] for details.

B. Example: Optimal Filtering Versus Prediction

As mentioned in Section I, there are numerous applications of sensor scheduling in communication and radar systems. Here, we outline one such application.

Consider the tracking problem of measuring the coordinates (state) of a target from radar derived measurements. Assume that the target's coordinates (state) evolve according to a finite-state Markov chain X_k with known transition probability matrix A. Assume that at each time instant k, we have two choices.

i) $u_k = 1$: Obtain a radar derived noisy measurement of the target position $y_k(u_k=1)$. Assume that the noise density and, hence, the symbol probability matrix $B(u_k=1, y_k)$ is known. After observing the target's position $y_k(u_k = 1)$, we compute the best *filtered* estimate of the target's position π_k by using the HMM filter. Let $c_k(X_k, 1)$ denote the cost of obtaining a noisy measurement $y_k(1)$ using the radar when the target's true position is X_k. For example, the cost $c_k(X_k, 1)$ would typically be large when the target X_k is close to the radar tracker.

ii) $u_k = 2$: Do not observe the target state. This is equivalent to choosing $B(u_k = 2, y_k) = I$ as the observation y_k then contains no information of the state of the Markov chain X_k. Without using the radar for observing the target, we can only compute the best *predicted* estimate of the target via an HMM state predictor. Let $c(X_k, 2)$ denote the cost of not using the radar.

In addition to the usage (operating) cost $c(X_k, u_k)$, we also incorporate into our cost function the mean-square estimation error of the target's coordinates. Suppose our aim is to chose at each time between $u_k = 1$ (obtaining a radar derived observation and using a HMM filter) versus $u_k = 2$ (not making a measurement and using a HMM predictor) to minimize the cost function in (4). Then, the problem is identical to the sensor scheduling problem posed above. Note that the above cost function depicts the tradeoff between sensor estimation accuracy and sensor usage costs; a predictor is cheaper to use than a filter but incurs a higher state estimation error cost since it is less accurate. In Section VI, we consider the aircraft identification problem and present numerical examples.

C. Information State Formulation

As it stands, the above HMM sensor scheduling problem is a partially observed infinite horizon stochastic control problem. As is standard with such stochastic control problems, in this section, we convert the partially observed stochastic control problem to a fully observed stochastic control problem defined in terms of the *information state* [12].

The information state at time k is merely the HMM conditional filtered density π_k that was already defined in (2).

Let $\mathcal{P}$ denote the set of all information states π. That is

$$\mathcal{P} = \{\pi \in \mathbb{R}^S : \mathbf{1}_S'\pi = 1,\ 0 \le \pi(i) \le 1 \text{ for all } i \in \{1\ldots,S\}\} \tag{5}$$

Note that $\mathcal{P}$ is a $S-1$-dimensional simplex and that the HMM filtered density (information state) π_k lives in $\mathcal{P}$ at each time k. The Markov chain states $e_1, e_2, \ldots, e_S$ are merely the S corner points of $\mathcal{P}$, We will subsequently refer to $\mathcal{P}$ as the *information state space simplex*.

The information state π_k is a sufficient statistic to describe the current state of an HMM (see [4] and [12]). The information state update is computed straightforwardly by the HMM state filter (which is also known as the "forward algorithm" [18])

$$\pi_{k+1} = \frac{B(u_{k+1}, y_{k+1}(u_{k+1}))A'\pi_k}{\mathbf{1}_S' B(u_{k+1}, y_{k+1}(u_{k+1}))A'\pi_k}, \qquad \pi_0 \in \mathcal{P} \tag{6}$$

where $\mathbf{1}_S$ represents an S-dimensional vector of ones.

Consider the cost functional (4). For notational convenience, define the S-dimensional vector

$$c_k(u_{k+1}) = [\, c_k(e_1, u_{k+1}) \quad \cdots \quad c_k(e_S, u_{k+1})]'.$$

Using the smoothing property of conditional expectation, the cost functional of (4) can be rewritten in terms of the information state π_k as (see [12, ch. 7] or [4, ch. 5] for details)

$$J_\mu = \mathbb{E}\left\{ C_N(\pi_N) + \sum_{k=0}^{N-1} C_k(\pi_k, u_{k+1}) \right\} \tag{7}$$

where $u_{k+1} = \mu_{k+1}(\pi_k)$

$$\begin{aligned} C_N(\pi_N) &= \alpha_N g(\pi_N)'\pi_N \\ C_k(\pi_k, u_{k+1}) &= \alpha_k(u_{k+1}) g(\pi_k)'\pi_k + c_k'(u_{k+1})\pi_k \\ k &\in \{0, \ldots, N-1\}. \end{aligned} \tag{8}$$

In the above equations, $g(\pi_k)$ denotes the S-dimensional "distance" vector

$$g(\pi_k) = [\, \|e_1 - \pi_k\|_D \quad \|e_2 - \pi_k\|_D \quad \cdots \quad \|e_S - \pi_k\|_D\,]'. \tag{9}$$

We now have a fully observed control problem in terms of the information state π. Find an admissible control law μ, which minimizes the cost functional of (7), subject to the state evolution equation of (6).

D. Stochastic Dynamic Programming Framework

In this subsection, we present the stochastic backward dynamic programming (DP) recursion for computing the optimal HMM sensor scheduling policy. The DP recursion is a functional equation and does not directly translate to practical solutions. In Section III, we will present algorithms for solving the

DP recursion. Please see [12, ch. 7] or [4, ch. 5] for details of DP.

Define the "value-to-go" function as

$$J_N(\pi) = C_N(\pi)$$
$$J_k(\pi) = \min_{\mu_{k+1},\ldots,\mu_N} \mathbb{E}\left\{ C_N(\pi_N) + \sum_{t=k}^{N-1} C_t(\pi_t, \mu_{t+1}(\pi_t)) \mid \pi_k = \pi \right\}.$$

Then, the dynamic programming recursion proceeds backward in time from $k = N$ to $k = 0$ as follows:

$$J_N(\pi) = C_N(\pi)$$

and for $k = N-1, N-2, \ldots, 0$

$$J_k(\pi) = \min_{u_{k+1}\in\{1,\ldots,L\}} \Big[C_k(\pi, u_{k+1}) + \sum_{m=1}^{M_{u_{k+1}}} J_{k+1}\left(\frac{B(u_{k+1}, O_m(u_{k+1}))A'\pi}{\mathbf{1}'_S B(u_{k+1}, O_m(u_{k+1}))A'\pi} \right) \times \mathbf{1}'_S B(u_{k+1}, O_m(u_{k+1}))A'\pi \Big]$$
$$\text{for all } \pi \in \mathcal{P}. \quad (10)$$

Finally, the optimal cost starting from the initial condition π_0 is given by $J_0(\pi_0)$ and if $u^*_{k+1} = \mu^*_{k+1}(\pi_k)$ minimizes the right-hand side of (10) for each k and each π_k, the optimal scheduling policy is given by $\mu^* = \{\mu^*_1, \mu^*_2, \ldots, \mu^*_N\}$. See [12, ch. 7] or [4, ch. 5] for proofs of the optimality of the above DP recursion.

As it stands, the above dynamic programming equations (10) have two major problems.

1) The information state $\pi \in \mathcal{P}$ is continuous valued. Hence, the dynamic programming equations (10) do not directly translate into practical solution methodologies. The fundamental problem is that at each iteration of the dynamic programming recursion, $J_k(\pi)$ needs to be evaluated at each $\pi \in \mathcal{P}$, which is an uncountably infinite set.
2) If $C_k(\pi, u)$ [which is defined in (8)] was a linear function of the information state π, then the above dynamic programming recursion is an instance of the well-known POMDP and can be solved explicitly using the algorithms given, for example, in [14] and [20]. However, note from (8) that $C_k(\pi_k, u_{k+1}) = \alpha_k(u_{k+1})g(\pi_k)'\pi_k + c'_k(u_{k+1})\pi_k$. The estimation error cost term $g(\pi)'\pi$ is not a linear function of π. It is for this reason that the HMM sensor scheduling problem is different from the standard POMDP. In particular, the above DP recursion cannot directly be solved in closed form using the POMDP techniques of [14] and [20].

In Section III, optimal algorithms for solving the above DP recursion are presented.

III. Dynamic Programming Based HMM Sensor Scheduling

A. *Optimal Scheduling Algorithm for Piecewise Linear Cost*

Here, we consider the case where the sensor estimation cost $g'(\pi)\pi$ defined in (9) is a piecewise linear and continuous function of π. Examples of such piecewise linear cost functions in the information state include the following.

i) We have the case when $\alpha_k(u_k) = 0$ in (4), i.e., the sensor estimation error cost is zero, and hence, (7) is linear in π. This is the well-known standard POMDP, which is widely studied in [5] and [14].
ii) D is a quantized (discrete) norm. For example, for $i = 1, 2, \ldots, S$

$$\|e_i - \pi_k\|_D \triangleq \begin{cases} 0, & \text{if } \|e_i - \pi_k\|_\infty \le \epsilon \\ \epsilon, & \text{if } \epsilon \le \|e_i - \pi_k\|_\infty \le 1-\epsilon \\ 1, & \text{if } \|e_i - \pi_k\|_\infty \ge 1-\epsilon \end{cases}$$

where $0 < \epsilon < 0.5$ is a fixed constant, and $\|\cdot\|_\infty$ denotes the l_∞ norm. This distance measure is particularly useful for subjective decision making, e.g., the estimate distance of a target to the base station computed via an HMM filter is quantized into three regions: close, medium, and far. [It is easy to check that with this quantized norm, $g(\pi)$, which defined in (9), is continuous in π.]
iii) Piecewise linear costs can be used to approximate nonlinear cost functions of the information state uniformly and arbitrarily closely, as will be shown in Section III-B.

Let $\mathcal{P}_1, \ldots, \mathcal{P}_R$ denote the partition of the information state space simplex $\mathcal{P}$ over which $g(\pi)$ is piece-wise linear. The piecewise linearity implies that there exist R vectors $\bar{g}_1, \bar{g}_2, \ldots, \bar{g}_R$ such that $g(\pi)$ can be represented as

$$g(\pi) = \sum_{r=1}^{R} \bar{g}_r I(\pi \in \mathcal{P}_r) \quad (11)$$

where $I(\cdot)$ denotes the indicator function

$$I(\pi \in \mathcal{P}_r) = \begin{cases} 1, & \text{if } \pi \in \mathcal{P}_r \\ 0, & \text{otherwise} \end{cases}$$

Due to the convexity and piecewise linearity, an equivalent and more convenient representation in terms of the sensor estimation cost $g'(\pi)\pi$ (8) is

$$g'(\pi)\pi = \min_{r\in 1,2\ldots,R} \bar{g}'_r \pi. \quad (12)$$

The following theorem gives an explicit solution of the dynamic programming recursion (10). It shows that the solution to the DP recursion is convex piecewise linear and, thus, completely characterized at each time instant k by a finite set of vectors (piecewise linear segments).

Theorem 3.1: At each time instant k, the value-to-go function $J_k(\pi)$ is convex and piece-wise linear. $J_k(\pi)$ has the explicit representation

$$J_k(\pi) = \min_{i\in\Gamma_k} \gamma'_{i,k}\pi \text{ for all } \pi \in \mathcal{P} \quad (13)$$

where Γ_k is a finite set of S-dimensional vectors.

Proof: The proof is by induction. At time N, from (10), $J_k(N) = \min_r \bar{g}_r' \pi_N$, which is of the form (13).

Assume at time $k+1$ that $J_{k+1}(\pi)$ has the form $\min_{i \in \Gamma_{k+1}} \gamma_{i,k+1}' \pi$. Then, substituting this expression in (10), we have

$$\begin{aligned} J_k(\pi) = & \min_{u \in \{1,\ldots,L\}} \Big[\alpha_k(u) \min_r [\bar{g}_r'(u)\pi] + c_k'(u)\pi \\ & + \sum_{m=1}^{M_u} \min_{i \in \Gamma_{k+1}} \gamma_{i,k+1}' B(u, O_m(u)) A' \pi \Big] \\ = & \min_{u \in \{1,\ldots,L\}} \Big[\Big(\alpha_k(u) \bar{g}_{r_\pi^*} + c_k(u) \\ & + \sum_{m=1}^{M_u} AB(u, O_m(u)) \gamma_{i^*(m,u,\pi)} \Big)' \pi \Big] \end{aligned} \quad (14)$$

where $r_\pi^* = \operatorname{argmax}_{r \in \{1,\ldots,R\}} \bar{g}_r' \pi$, and $i^*(m,u,\bar{\pi}) = \operatorname{argmax}_{\gamma_{i,k+1} \in \Gamma_{k+1}} \pi' AB(u, O_m(u)) \gamma_{i,k+1}$. The last expression in (14) is of the form $\min_{i \in \Gamma_k} \gamma_{i,k+1}' \pi$ because the sum of piecewise linear continuous convex functions is also piecewise linear continuous convex. □

Optimal Algorithm: Theorem 3.1 shows that the solution to the DP recursion (10) is convex piecewise linear and completely characterized at each time k by the finite set of vectors Γ_k defined in (13). Thus, we need to devise an algorithm for computing the set Γ_k at each time k. Given the piecewise linear convex nature of the value function, programs for solving standard POMDP (with appropriate modifications) can be used to solve the above DP recursion. There are numerous linear programming-based algorithms in the POMDP literature such as Sondik's algorithm [20], Monahan's algorithm [17], Cheng's algorithm [14], and the Witness algorithm [5] that can be used to compute the finite set of vectors Γ_k. See [6] for an excellent tutorial exposition with graphics of these various algorithms. Any of these algorithms will equally well, although not with the same computational efficiency (as discussed later), produce the desired solution set of vectors $\Gamma_k = \{\gamma_{k,i}\}$, together with the optimal actions $\{u_{k,i}^*\}$, $i = 1, 2, \ldots, |\Gamma_k|$, where $|\Gamma_k|$ denotes the number of vectors in the set Γ_k.

```
HMM Scheduling Algorithm for Piecewise
Linear Cost
```
Off-line Dynamic Programming: Run the POMDP algorithm to compute $\Gamma_k = \{\gamma_{k,i}\}$ together with the optimal actions $\{u_{k,i}^*\}$, $i = 1, 2, \ldots, |\Gamma_k|$

Real Time Scheduling: Given state estimate π_k from HMM filter, choose optimal action $u_{k,i(\pi_k)}^*$

Note that the entire dynamic programming algorithm and, hence, the computing the value-to-go function vectors Γ_k are offline and independent of the data.

B. HMM Sensor Scheduling for Quadratic Costs in Information State

In deriving the optimal scheduling algorithm in Section III-A, we assumed that the state estimation part of the cost function $\alpha_k(u_k) \|X_k - \pi_k\|_D$ was piecewise linear in the information state π_k. Here, we consider the scheduling of HMM sensors where the estimation error costs are quadratic functions of the information state such as the l_1 cost, l_2 cost, and l_∞ cost. For example, optimizing the l_2 sensor estimation error cost yields the the best sensor, which minimizes the weighted mean square error in the state estimate. We will approximate these costs by piecewise linear interpolations on the information state space.

Consider the cost function (4) and (7), which is repeated in the following for convenience:

$$\begin{aligned} J_\mu = & \mathbb{E} \left\{ \sum_{k=0}^{N-1} \alpha_k(u_{k+1}) \|X_k - \pi_k\|_D \right. \\ & \left. + \sum_{k=0}^{N-1} c_k(X_k, u_{k+1}) \} + \alpha_N \|X_N - \pi_N\|_D \right\} \\ = & \mathbb{E} \left\{ \sum_{k=0}^{N-1} \alpha_k(u_{k+1}) g(\pi_k)' \pi_k \right. \\ & \left. + c_k'(u_{k+1}) \pi_k + \alpha_N g(\pi_N)' \pi_N \right\}. \end{aligned}$$

In this section, we consider the following three distance functions D:

$$l_1\text{cost}: \quad \|X_k - \pi_k\|_D = \sum_{i=1}^{S} |X_k(i) - \pi_k(i)| \quad (15)$$

$$l_2\text{cost}: \quad \|X_k - \pi_k\|_D = \sum_{i=1}^{S} |X_k(i) - \pi_k(i)|^2 \quad (16)$$

$$l_\infty\text{cost}: \quad \|X_k - \pi_k\|_D = \max_{i \in \{1,\ldots,S\}} |X_k(i) - \pi_k(i)|. \quad (17)$$

In terms of the information-state formulation, we can re-express the above cost as follows.

Lemma 3.2: In terms of the information state, the cost (4) can be re-expressed as (7), where

$$\begin{aligned} l_1\text{cost}: \quad g(\pi) = & 2(\mathbf{1}_S - \pi) \\ C_k(\pi_k, u_{k+1}) = & 2\alpha_k(u_{k+1})(1 - \pi_k'\pi_k) \\ & + c_k(u_{k+1})'\pi_k \end{aligned} \quad (18)$$

$$\begin{aligned} l_2\text{cost}: \quad g(\pi) = & \mathbf{1}_S - \pi \\ C_k(\pi_k, u_{k+1}) = & \alpha_k(u_{k+1})(1 - \pi_k'\pi_k) \\ & + c_k(u_{k+1})'\pi_k \end{aligned} \quad (19)$$

$$\begin{aligned} l_\infty\text{cost}: \quad g(\pi) = & \mathbf{1}_S - \pi \\ C_k(\pi_k, u_{k+1}) = & \alpha_k(u_{k+1})(1 - \pi_k'\pi_k) \\ & + c_k(u_{k+1})'\pi_k. \end{aligned} \quad (20)$$

The proof of the above lemma is straightforward and has been omitted. Because the l_1, l_2, and l_∞ estimation error costs are

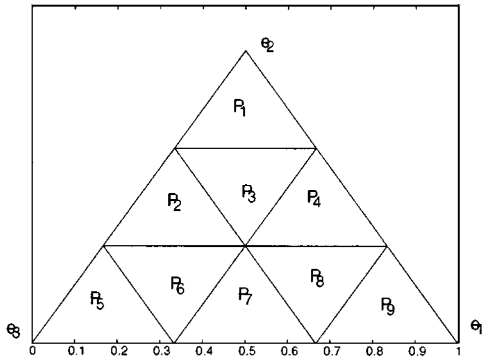


Fig. 2. For a three-state HMM ($S = 3$), the information state simplex $\mathcal{P}$ is an equilateral triangle. The corners are the Markov chain states e_1, e_2, ande_3. The $R = 9$ Freudenthal triangularization regions over which the l_2 cost is approximated by piecewise linear segments is also shown.

identical (modulo a constant scaling factor), subsequently, only l_2 costs will be considered.

With the l_2 state estimation cost, the dynamic programming recursion (10) becomes

$$\begin{aligned} J_k(\pi) = \min_{u\in\{1,\ldots,L\}} & \Big[\alpha_k(u)(1-\pi'\pi) + c_k'(u)\pi \\ & + \sum_{m=1}^{M_u} J_{k+1}\left(\frac{B(u,O_m(u))A'\pi}{\mathbf{1}_S' B(u,O_m(u))A'\pi}\right) \\ & \times \mathbf{1}_S' B(u,O_m(u))A'\pi \Big], \text{ for all } \pi\in\mathcal{P}. \quad (21) \end{aligned}$$

Because of the nonlinear term $g'(\pi)\pi = (1-\pi'\pi)$ in the above recursion, it is impossible to find a closed-form solution to the dynamic programming equation, in particular, the solution is definitely not piecewise linear! However, because the information state π resides in a compact space defined by the simplex $\mathcal{P}$, these nonlinear cost functions can be approximated from above and below by piecewise linear cost functions arbitrarily closely. In the following, we construct such piecewise linear approximations.

Lower Bound Using Freudenthal Triangularization: Given the simplex structure of the information state space $\mathcal{P}$ and because $g'(\pi)\pi = 1-\pi'\pi$ is strictly concave, the most natural piecewise linear approximation for lower bounding the cost function is to perform a uniform triangular interpolation of the terms $g'(\pi)\pi = 1-\pi'\pi$ that arise in the l_1, l_2, and l_∞ costs defined previously. In the numerical examples presented in Section VI, we will use this piecewise linear interpolation. The uniform triangular interpolation we describe later is called Freudenthal triangularization after the German mathematician who introduced it in 1942. Freudenthal triangularization has been used by Lovejoy [13] to derive efficient suboptimal algorithms for solving POMDPs; see Section V-B.

To graphically illustrate the procedure, consider the three-state HMM example (i.e., $S = 3$). The procedure is trivially extended to the case $S > 3$. We represent the states of the three-state Markov chain by the unit vectors e_1, e_2, e_3. Note that in the three-state case, the information state space simplex $\mathcal{P}$ defined in (5) is merely the equilateral triangle depicted in Fig. 2, where the three corners of the triangle denote the three Markov chain states e_1, e_2 and e_3, respectively.

We construct a piecewise linear interpolation of this l_2 cost on the information state space simplex. For $S = 3$, the Ith degree interpolation consists of partitioning the simplex $\mathcal{P}$ into $R = I^{S-1}$ triangular patches $\mathcal{P}_1,\ldots,\mathcal{P}_R$. As an example, Fig. 2 shows the $R = 9$ triangular patches that arise for the $I = 3$ interpolation of a $S = 3$-state HMM. Note that each region $\mathcal{P}_r$, $r = 1,2,\ldots,R$ is an equilateral triangle simplex.

Let $\pi_{r,1},\pi_{r,2},\ldots,\pi_{r,S}$ denote the corners of the rth patch $\mathcal{P}_r$.

For each region $\mathcal{P}_r$, the function $g'(\pi)\pi$ is approximated by the piecewise linear cost $\bar{g}_r'\pi$ as follows:

$$\bar{g}_r' = \begin{bmatrix} \pi_{r,1}' \\ \pi_{r,2}' \\ \vdots \\ \pi_{r,S}' \end{bmatrix}^{-1} \begin{bmatrix} g'(\pi_{r,1})\pi_{r,1} \\ g'(\pi_{r,2})\pi_{r,2} \\ \vdots \\ g'(\pi_{r,S})\pi_{r,S} \end{bmatrix}, \qquad r = 1,2,\ldots,R. \quad (22)$$

It is clear that the above matrix is invertible since any S distinct information states in the simplex $\mathcal{P}$ are linearly independent; this fact is straightforward to establish from the definition of $\mathcal{P}$ in (5). The resulting piecewise linear interpolation can be expressed similarly to (11) or (12) as $\sum_{r=1}^{R}\bar{g}_r' I(\pi\in\mathcal{P}_r)\pi$, or equivalently, $\min_{r\in R}\bar{g}_r'\pi$.

Because $g'(\pi)\pi = (1-\pi'\pi)$ is strictly concave, it is lower bounded by the above piecewise linear interpolation $\min_{r\in R}\bar{g}_r'\pi$, i.e., $\min_{r\in R}\bar{g}_r'\pi \le (1-\pi'\pi)$. The following result gives an expression for the maximum approximation error between $g'(\pi)\pi = (1-\pi'\pi)$ and the piecewise linear interpolation $\min_{r\in R}\bar{g}_r'\pi$.

Theorem 3.3: The maximum approximation error at a corner simplex of the above piecewise linear Freudenthal approximation comprising of R piecewise linear segments is given by

$$\max_{\pi\in\mathcal{S}} \mid 1-\pi'\pi - \min_{r\in R}\bar{g}_r'\pi \mid \le \frac{1}{R^2}\left(1-\frac{1}{S}\right). \quad (23)$$

Proof: It is easy to show that the approximation error for any segment error $|1-\pi'\pi-\bar{g}_r'\pi|$ is maximized at

$$\pi = \mathbf{1}_S + \frac{1}{2S}\mathbf{1}_S'\bar{g}\mathbf{1}_S - \frac{1}{2}\bar{g}_r. \quad (24)$$

Because of the symmetry in the uniform Freudenthal triangularization for $S \le 3$, the maximum error is achieved in each of the R interpolated regions. Therefore, we need consider only any one region to evaluate this error. Consider the corner simplex comprising the S information state points

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1-\frac{1}{R} \\ \frac{1}{R} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1-\frac{1}{R} \\ 0 \\ \frac{1}{R} \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1-\frac{1}{R} \\ 0 \\ 0 \\ \vdots \\ \frac{1}{R} \end{bmatrix}.$$

Then, $\bar{g}_r$ is obtained by solving (22). It is not difficult to see that the solution $\bar{g}_r = 2(\mathbf{1}_S - e_1)(1-1/R)$. Thus, from (24), the maximum error is attained at $\pi = 1/(RS) + (1-1/R)e_1$. Thus, the maximum error $|1-\pi'\pi-\bar{g}_r'\pi|$ evaluated at this value of π and $\bar{g}_r$ is given by (23). □

Let $\bar{J}_k(\pi)$ denote the value function at time k obtained by replacing $g'(\pi)\pi = 1-\pi'\pi$ in (21) by its lower bound piecewise linear approximation $\min_{r\in R} \bar{g}_r'\pi$. Clearly, $\bar{J}_k(\pi)$ is piecewise linear (as shown in Section III-A) and can be computed explicitly using the POMDP algorithm of Section III-A. In Theorem 3.4, we will show that $\bar{J}_k(\pi) < J_k(\pi)$, meaning that the policy obtained by solving the POMDP with value function $\bar{J}_k(\pi)$ forms a lower bound to the optimal scheduling policy.

Upper Bound Using Tangents: Because $g'(\pi)\pi = 1 - \pi'\pi$ is strictly concave, it can be upper bounded by a piecewise linear interpolation comprising of tangents to $g'(\pi)\pi$ at arbitrary points $\pi_1, \pi_2, \ldots, \pi_R$ in $\mathcal{P}$. Using elementary computations, it follows that the tangent at a point π_r is the linear segment

$$\underline{g}_r'\pi \qquad \text{where } \underline{g}_r = (1 - \pi_r'\pi_r)\mathbf{1}_S - 2\pi_r, \quad r = 1, 2, \ldots, R. \tag{25}$$

Thus, $g'(\pi)\pi = 1 - \pi'\pi$ can be upper bounded by the piecewise linear function $\min_{r\in R} \underline{g}_r'\pi$, which is comprised of tangent vectors, i.e., $\min_{r\in R} \underline{g}_r'\pi \geq (1 - \pi'\pi)$. Let $\underline{J}_k(\pi)$ denote the value function at time k obtained by replacing $g'(\pi)\pi = 1 - \pi'\pi$ in (21) by its upper bound piecewise linear approximation $\min_{r\in R} \underline{g}_r'\pi$. Clearly, $\underline{J}_k(\pi)$ is piecewise linear (as shown in Section III-A) and can be computed explicitly using the POMDP algorithm of Section III-A.

Sandwiching Theorem and Characterization of Optimal Scheduling Policy: Here, we show that the value functions $\bar{J}_k(\pi)$ and $\underline{J}_k(\pi)$ themselves form lower and upper bounds for the optimal value function $J_k(\pi)$. That is, the resulting value functions and policies form lower and upper bounds for the optimal sensor scheduling value function and policy.

Theorem 3.4: Let $\bar{J}_k(\pi)$ and $\underline{J}_k(\pi)$ denote the value functions at time k obtained by replacing $g'(\pi)\pi = 1 - \pi'\pi$ in (21) by its lower and upper bound piecewise linear approximations, i.e., $\min_{r\in R} \bar{g}_r'\pi$ and $\min_{r\in R} \underline{g}_r'\pi$, respectively. Then, the optimal value function $J_k(\pi)$of (21) satisfies

$$\bar{J}_k(\pi) \leq J_k(\pi) \leq \underline{J}_k(\pi) \qquad \text{for all } \pi \in \mathcal{S} \text{ and } k = N, N-1, \ldots, 0. \tag{26}$$

Moreover, the approximation error of the upper bound satisfies

$$\max_{\pi\in\mathcal{P}} |J_0(\pi) - \bar{J}_0(\pi)| \leq \frac{N}{R^2}\left(1 - \frac{1}{S}\right). \tag{27}$$

Proof: The proof is by induction. At time N, by construction of the lower and upper bound approximations seen previously

$$\bar{J}_N(\pi) = \min_{r\in R} \bar{g}_r'\pi \leq J_N(\pi) = g'(\pi)\pi \leq \min_{r\in R} \underline{g}_r'\pi.$$

Suppose (26) is true at time k. It is convenient to introduce the operator $\mathcal{H}$ for the right-hand side of (21) so that the DP recursion (21) can be written as

$$J_k = \mathcal{H} \circ J_{k+1}.$$

The well-known monotone property of the operator $\mathcal{H}$ (see [4, p. 7] or [13]) implies that

$$\bar{J}_k \leq J_k \leq \underline{J}_k \Longrightarrow \mathcal{H} \circ \bar{J}_k \leq \mathcal{H} \circ J_k \leq \mathcal{H} \circ \underline{J}_k.$$

However, $\mathcal{H} \circ \bar{J}_k \geq \bar{J}_{k-1}$ since $\min_{r\in R} \bar{g}_r'\pi \leq (1 - \pi'\pi)$ by construction. Similarly, $\mathcal{H} \circ \bar{J}_k \leq \underline{J}_{k-1}$ since $\min_{r\in R} \underline{g}_r'\pi \geq (1 - \pi'\pi)$ by construction. Hence, (26) holds at time $k-1$. Finally, (27) follows directly from (23). □

We conclude this section by showing that although the optimal value function $J_k(\pi)$ cannot be computed in closed form (i.e., it is not piecewise linear), the piecewise linear value function $\bar{J}_k(\pi)$ for sufficiently large R yields the optimal scheduling policy. This further justifies the piecewise linear approximation used above.

Theorem 3.5: There exists a finite integer $R_0 > 0$ such that for all integers $R \geq R_0$, the scheduling policy μ, which maximizes the piecewise linear interpolated cost

$$\bar{J}_\mu(R) \triangleq \mathbb{E}\left\{\sum_{k=0}^{N-1} \alpha_k(u_{k+1}) \sum_{r=1}^{R} \bar{g}_r'\pi_k + c_k'(u_{k+1})\pi_k + \alpha_N \sum_{r=1}^{R} \bar{g}_r'\pi_N\right\} \tag{28}$$

is identical to the optimal policy that maximizes the cost J_μ defined in (4) or (7).

Proof: It is clear from (28) and (4) that $\lim_{R\to\infty} \bar{J}_\mu(R) = J_\mu$. Note that for any finite horizon N, there are only a finite number (M^N) of possible scheduling sequences. Therefore, the convergence is uniform in μ, meaning that

$$\limsup_{R\to\infty} \bar{J}_\mu(R) = J_\mu.$$

Uniform convergence in μ implies that there exists an integer R_0 such that for all $R \geq R_0$

$$\operatorname*{argmax}_\mu \bar{J}_\mu(R) = \operatorname*{argmax}_\mu J_\mu.$$

□

The previous theorem shows that although one cannot determine the optimal value function $\max_\mu J_\mu$, one can determine the optimal scheduling policy $\operatorname{argmax}_\mu J_\mu$ by optimizing the piecewise linear interpolated cost $\bar{J}_\mu(R)$ for a sufficiently large integer R. However, we have been unable to give a lower bound to R_0.

IV. Constraints on Estimation Error, Sensor Management Issues and Stationary Schedulers

Thus far, we have presented HMM sensor scheduling algorithms for unconstrained finite horizon cost functions. Here, we consider constraints on the sensor estimation error, sensor management issues, and infinite horizon problems. The constraints considered can be formulated as local-in-time constraints and, hence, are amenable to DP methods. Global constraints (e.g., requiring $\mathbb{E}\{\sum_{k=1}^{N} g(X_k, u_{k+1})\} < \nu$ for some bounded measurable function $g(\cdot,\cdot)$ and constant ν) are not considered in this paper. These require the use of Lagrange multipliers (see [1] for an excellent exposition) and do not have closed-form solutions for HMMs.

A. Usage Costs With Quadratic Constraints on Estimation Error

Instead of minimizing the estimation error plus sensor usage cost, it often makes sense to minimize the sensor usage cost subject to constraints on the average sensor estimation error at each time instant k. Here, we consider minimizing sensor usage costs (which are linear in the information state) subject to quadratic constraints on the sensor estimation error. Such local constraints easily fit within the framework of stochastic dynamic programming. For convenience, we consider the l_2 norm in the constraints.

Let $\mathcal{L} = \{1,\ldots,L\} = \{\mathcal{L}_c \bigcup \mathcal{L}_{\bar{c}}\}$, where $\mathcal{L}_c$ denotes the user specified set of sensors with state estimation error constraints, and $\mathcal{L}_{\bar{c}}$ denotes the set of sensors without constraints.

The sensor usage cost we consider is to compute $\min_\mu J_\mu$, where

$$J_\mu = \mathbb{E}\left\{\sum_{k=0}^{N-1} c_k(X_k, u_{k+1})\right\} = \mathbb{E}\left\{\sum_{k=0}^{N-1} c'_k(u_{k+1})\pi_k\right\}$$

subject to the following average constraints on each of the sensors in $\mathcal{L}_c$:

$$\mathbb{E}_{y_{k+1}}\left\{\alpha_{k+1}(l)\|X_{k+1} - \pi_{k+1}\|_2 \mid \pi_k = \pi, u_{k+1} = l\right\} \le K_l \qquad l \in \mathcal{L}_c, \quad k = 1,\ldots,N$$

where $0 < K_l < 1$ are user specified constants. The constraint says that sensor l can only be used at time k if the mean square state estimation error at time $k+1$ is less than the user-specified constant K_l. Note that the above constraints can be explicitly expressed as a local constraint at each time k in terms of the information state $\pi_k = \pi$ as

$$\sum_{m=1}^{M_u} \alpha_{k+1}(l)\left(1 - \frac{\pi' AB^2(u, O_m(u))A'\pi}{\mathbf{1}'_S B(u, O_m(u))A'\pi}\right) < K_l, \quad l \in \mathcal{L}_c. \tag{29}$$

Let D_l denote the closed subsets of the information state simplex $\mathcal{P}$ such that

$$D_l \triangleq \left\{\pi : \sum_{m=1}^{M_u} \alpha_{k+1}(l)\left(1 - \frac{\pi' AB^2(u, O_m(u))A'\pi}{\mathbf{1}'_S B(u, O_m(u))A'\pi}\right) < K_l\right\}, l \in \mathcal{L}_c. \tag{30}$$

Let U_{k+1} denote the set of admissible sensors at time $k+1$, i.e.,

$$U_{k+1} = \{\mathcal{L}_{\bar{c}} \bigcup (\cup_{l\in\mathcal{L}_c} l : \pi \in D_l)\}.$$

Then, the dynamic programming recursion is

$$J_k(\pi) = \min_{u\in U_{k+1}} c'_k(u)\pi + \sum_{m=1}^{M_u} J_{k+1}\left(\frac{B(u, O_m(u))A'\pi}{\mathbf{1}'_S B(u, O_m(u))A'\pi}\right) \mathbf{1}'_S B(u, O_m(u))A'\pi\ ], \qquad \text{for all } \pi_k \in \mathcal{P}. \tag{31}$$

The finite-dimensional sensor scheduling algorithm of Section III-A can be used as follows.

```
HMM Sensor Scheduling Algorithm With State Estimation Constraints
```

Off-Line Dynamic Programming: Run the POMDP algorithm twice with different action sets as follows:

i) Run the POMDP program with action set $\mathcal{L} = \{1,\ldots,L\}$. Let $\gamma^{\mathcal{L}}_{k,i}$, $u^{\mathcal{L},*}_{k,i}$ be the vectors and associated optimal actions, $k = N-1,\ldots,0$ computed by the POMDP program.

ii) Run the POMDP program with action set $\mathcal{L}_{\bar{c}}$. Let $\gamma^{\mathcal{L}_{\bar{c}}}_{k,i}$, $u^{\mathcal{L}_{\bar{c}},*}_{k,i}$ be the vectors and associated optimal actions, $k = N-1,\ldots,0$.

Real-Time Scheduling: Given the information state π_k from the HMM filter, let $\{\gamma^{\mathcal{L}}_{k,i(\pi_k)}, u^{\mathcal{L},*}_{k,i(\pi_k)}\}$ and $\{\gamma^{\mathcal{L}_{\bar{c}}}_{k,i(\pi_k)}, u^{\mathcal{L}_{\bar{c}},*}_{k,i(\pi_k)}\}$ denote the corresponding optimal vector and action pairs.

If $\pi_k \in D_{u^{\mathcal{L},*}_{k,i(\pi_k)}}$, i.e., π_k satisfies **(29)** for $l = u^{\mathcal{L},*}_{k,i(\pi_k)}$, **then** set $u_{k+1} = u^{\mathcal{L},*}_{k,i(\pi_k)}$.

Else set $u_{k+1} = u^{\mathcal{L}_{\bar{c}},*}_{k,i(\pi_k)}$.

Example: Optimal HMM Filtering Versus HMM Prediction (Continued): Consider the optimal filtering versus prediction scheduling problem of Section II-B. Assume that the predict sensor $u_k = 2$ is constrained, i.e., $\mathcal{L} = \{1, 2\}$, $\mathcal{L}_c = \{2\}$, $\mathcal{L}_{\bar{c}} = \{1\}$. Because $B(u_k = 2, y_k) = I$, constraint (29) on the predict sensor is (where K_2 is a user specified parameter)

$$\alpha_{k+1}(2)(1 - \pi' AA'\pi) < K_2 \qquad \text{or equivalently} \quad \pi' AA'\pi > 1 - \frac{K_2}{\alpha_{k+1}(2)}. \tag{32}$$

This constraint describes a region $\mathcal{D}_2$ (30) in the information state space simplex $\mathcal{P}$. It says that the predict sensor can only be used at time k if the resulting state estimate variance at time $k+1$ lies within $\mathcal{D}_2$.

The feasibility of the above constraint (32) can be checked *a priori* as follows.

Theorem 4.1: Assume that AA' is positive definite. The constraint (32) is active and holds for some nonempty subset $\mathcal{D}_2$ of the information state space $\mathcal{P}$ if

$$1 - \max_i (AA')_{i,i} < \frac{K_2}{\alpha_{k+1}(2)} < 1 - \frac{1}{\mathbf{1}_S (AA')^{-1}\mathbf{1}_S}.$$

If $K_2/\alpha_{k+1}(2) > 1 - 1/(\mathbf{1}_S(AA')^{-1}\mathbf{1}_S)$, then the constraint (32) is always met for all $\pi \in \mathcal{P}$, i.e., $\mathcal{D}_2 = \mathcal{P}$, and the problem is identical to a standard unconstrained POMDP. Finally, if $K_2/\alpha_{k+1}(2) < 1 - \max_i(AA')_{i,i}$, then the constraint (32) is never met for any π, i.e., $\mathcal{D}_2$ is the null set. (In such a case, the HMM predictor can never be used.)

Proof: Consider the following quadratic programming problem:

$$\min \pi' AA'\pi \qquad \text{subject to } \pi'\mathbf{1}_S = 1.$$

From [15, pp. 424—425], it follows that the minimum value is $1/(\mathbf{1}_S(AA')^{-1}\mathbf{1}_S)$. Thus, if $1/(\mathbf{1}_S(AA')^{-1}\mathbf{1}_S) \ge 1 - K_2/\alpha_{k+1}(2)$, then the constraint (32) is always met for all $\pi \in \mathcal{P}$.

It can be shown that $\max \pi' AA'\pi$ subject to $\pi'\mathbf{1}_S = 1$ is given by $\max_i(AA')_{i,i}$. Therefore, clearly, if $\max_i(AA')_{i,i} < 1 - K_2/\alpha_{k+1}(2)$, then the constraint (32) is never met for any π, i.e., $\mathcal{D}_2$ is the null set.

Finally, for $1 - \max_i(AA')_{i,i} < K_2/\alpha_{k+1}(2) < 1/(\mathbf{1}_S(AA')^{-1}\mathbf{1}_S)$, the constraint (32) is active and holds for some subset $\mathcal{D}_2$ of the information state space $\mathcal{P}$. □

B. Sensor Management With Usage Constraints

The aim here is to derive optimal HMM sensor scheduling algorithms when there are constraints on the total number of times particular sensors can be used. Such constraints are often used in sensor resource management.

Consider an N horizon problem where sensor 1 can be used at most N_1 times. For simplicity, we assume that there are no constraints on the usage of the other sensors. The aim is to derive a sensor schedule to optimize the cost function (4).

Let $S_1 = \{f_1, \dots, f_{N_1+1}\}$ denote the set of $N_1 + 1$ dimensional unit vectors, where f_i has 1 in the ith position. We will use process z_k to denote the number of times sensor 1 is used. Specifically, let z_k be a N_1 state Markov chain with state space S_1. Let $z_k = f_i$ if sensor 1 has been used $i-1$ times. The dynamics of z_k are as follows. If sensor 1 is used (i.e., $u_k = 1$), then z_k jumps to state f_{i+1}. If any other sensor is used, then z_k remains unchanged. It is easily seen that z_k is a deterministic Markov chain with dynamics given by

$$z_k = Q(u_k) z_{k-1}, \qquad z_0 = e_1, z_N = e_{N_1+1} \tag{33}$$

where the transition probability matrix $Q(\cdot)$ is defined as

$$Q(u_k = 1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$
$$\text{and } Q(u_k) = I_{(N_1+1)\times(N_1+1)} \text{ if } u_k \neq 1.$$

We consider two types of constraints on the number of times a sensor can be used.

Equality Constraints: Suppose HMM sensor 1 must be used exactly N_1 times. Then, the action space U_{k,z_k} is defined as follows.

- For $k = 0, \dots, N - N_1, U_{k,z_k} = \{1, 2, \dots, L\}$.
- For $k = N - N_1, \dots, N$

$$U_{k,z_k} = \begin{cases} \{1\}, & \text{if } z_k = e_1 \\ \{1, 2, \dots, L\}, & \text{if } z_k \in \{e_2, \dots, e_{N_1-1}\} \\ \{2, \dots, L\}, & \text{if } z_k = e_{N_1}. \end{cases}$$

Inequality Constraints: Suppose HMM sensor 1 can be used at most N_1 times. The action space U_{k,z_k} is defined as follows.

- For $k = 0, \dots, N - N_1, U_{k,z_k} = \{1, 2, \dots, L\}$.
- For $k = N - N_1, \dots, N$

$$U_{k,z_k} = \begin{cases} \{1, 2, \dots, L\}, & \text{if } z_k \in \{e_2, \dots, e_{N_1-1}\} \\ \{2, \dots, L\}, & \text{if } z_k = e_{N_1}. \end{cases}$$

For both the equality and inequality constraint cases, the optimal scheduling policy is given by solving the following stochastic dynamic programming recursion:

$$J_N(\pi_N) = C_N(\pi_N)$$

and for $k = N-1, N-2, \dots, 0$

$$\begin{aligned} J_k(\pi, z) = \min_{u \in U_{k+1, z_{k+1}}} \Bigg[& C_k(\pi, u) \\ & + \sum_{m=1}^{M_u} J_{k+1} \left(\frac{B(u, O_m(u)) A' \pi}{\mathbf{1}'_S B(u, O_m(u)) A' \pi}, \ Q' z \right) \\ & \times \mathbf{1}'_S B(u, O_m(u)) A' \pi_k \Bigg]. \end{aligned} \tag{34}$$

The above dynamic programming recursion can be recast into a form similar to (10) by the following coordinate change: Consider the augmented Markov chain (X_k, z_k). This has transition probability matrix $\bar{A} = A \otimes Q$, where $\otimes$ denotes tensor (Kronecker product). Because z_k is a fully observed Markov chain, the information state of (X_k, z_k) is $\bar{\pi}_k \stackrel{\triangle}{=} \pi_k \otimes z_k$ with observation probability matrix $\bar{B}(u, O_m(u)) = B(u, O_m(u)) \otimes I$. Thus, the augmented information state $\bar{\pi}_k$ evolves according to the standard HMM filter (6) with A, B replaced by $\bar{A}$, $\bar{B}$. Define the value function

$$\bar{J}_k(\bar{\pi}) \stackrel{\triangle}{=} J_k(\pi, z), \qquad \bar{\pi} = \pi \otimes z.$$

The DP recursion (34) can be rewritten in terms of $\bar{J}_k(\bar{\pi})$ as (10), meaning that the optimal algorithms of Section III can be applied.

C. Infinite Horizon Scheduling Algorithm

The aim here is to determine the optimal stationary scheduling policy to minimize the infinite horizon discounted cost

$$J_\mu = \mathbb{E} \left\{ \sum_{k=0}^{\infty} \beta^k \left[\alpha(u_{k+1}) \| X_k - \pi_k \|_D + c(X_k, u_{k+1}) \right] \right\} \tag{35}$$

where $u_{k+1} = \mu(Y_k)$ and $0 < \beta < 1$ denotes the discount factor. (Notice that unlike the finite horizon case, α and $c(\cdot)$ are no longer explicit functions of time k.) Our aim is to minimize J_μ over the class of stationary policies μ, where $\mu : Y_k \to \{1, \dots, M\}$. It is well known [4] that the optimal stationary policy exists as long as $\alpha(u_{k+1}) \| X_k - \pi_k \|_D$ and $c(X_k, u_{k+1})$ are uniformly bounded from below and above.

The above problem can be expressed in terms of the information state π as $J_\mu = \mathbb{E}\{\sum_{k=0}^{N-1} \beta^k C(\pi_k, u_{k+1})\}$, where $C(\cdot)$ is defined as in (8).

The optimal stationary policy can be determined in principle by solving the following DP functional equation called "Bellman's equation" for the value function $V(\pi)$; see [14]):

$$\begin{aligned} V(\pi) = \min_{u \in \{1, \dots, L\}} \Bigg[& C(\pi, u) \\ & + \beta \sum_{m=1}^{M_u} V \left(\frac{B(u, O_m(u)) A' \pi}{\mathbf{1}'_S B(u, O_m(u)) A' \pi} \right) \\ & \times \mathbf{1}'_S B(u, O_m(u)) A' \pi \Bigg], \quad \pi \in \mathcal{P}. \end{aligned} \tag{36}$$

One approach to solving Bellman's equation is via the *value-iteration* algorithm; see [5] or [13] for details. This is merely a finite horizon approximation to the infinite horizon cost function (35). That is, pick a sufficiently large horizon N, and run the following slightly modified version of the finite horizon DP recursion (10):

$$V_0(\pi) = \min_{u \in \{1, \dots, L\}} C(\pi, u)$$

and for $k = 0, 1, \dots, N-1$

$$\begin{aligned} V_{k+1}(\pi) = \min_{u \in \{1, \dots, L\}} \Bigg[& C(\pi, u) \\ & + \beta \sum_{m=1}^{M_u} V_k \left(\frac{B(u, O_m(u)) A' \pi}{\mathbf{1}'_S B(u, O_m(u)) A' \pi} \right) \\ & \times \mathbf{1}'_S B(u, O_m(u)) A' \pi \Bigg]. \end{aligned} \tag{37}$$

The finite horizon algorithms detailed in earlier sections can be used to solve the above value-iteration recursion. It can be shown [14] that $\lim_{N\to\infty} V_N(\pi) \to V(\pi)$ uniformly in π. The obvious advantage of a stationary scheduler is that only the piecewise linear representation $\{\gamma_{N,i}\}$ of $V_N(\pi)$ and its associated optimal decision $\{u_N^*, i(\pi)\}$ need be stored in memory for the real-time implementation.

V. Suboptimal Algorithms

Here, we outline two suboptimal algorithms: a greedy one-step-ahead algorithm and Lovejoy's algorithm. See [11] for other suboptimal algorithms such as the "blind" policy iteration algorithm.

A. One-Step-Ahead Greedy Algorithm

For large state space problems, the above algorithms can be prohibitively expensive for real-time implementation. In this section, we outline a one-step-ahead suboptimal solution to the HMM sensor scheduling problem. The idea behind the one-step-ahead algorithm is to compute the expected posterior density for the target state for each HMM sensor based on the current posterior density and the known measurement models and then make a measurement using the HMM sensor that gave the best predicted cost. In [4], these are called limited lookahead policies.

To simplify notation, assume that the weight α in (3) and (4) is a constant. The one-step-ahead algorithm proceeds recursively as follows: Assume that the HMM filtered density π_{k-1} has been computed at time $k-1$.

Step 1) Minimize at time k the one-step-ahead cost. Using the dynamic programming recursion (10), the above cost function is straightforwardly minimized by

$$\begin{aligned} V_k(\pi_k) =& \alpha\left(1-{\pi_k}'\pi_k\right) \\ u_k^* =& \arg\min_{u\in\{1,\ldots,L\}} c'_{k-1}(u)\pi_{k-1} \\ &+\alpha\sum_{m=1}^{M}\left[1-\frac{\pi'_{k-1}AB^2(O_m(u_k))A'\pi_{k-1}}{\left(\mathbf{1}'_S B(O_m(u_k))A'\pi_{k-1}\right)^2}\right]. \end{aligned} \quad (38)$$

Step 2) Using the observation from sensor u_k^*, compute π_k using the HMM filter as

$$\pi_k = \frac{B(u_k, y_k(u_k^*))A'\pi_{k-1}}{\mathbf{1}'_S B(u_k, y_k(u_k^*))A'\,\pi_{k-1}}.$$

Step 3) $k \to k+1$.

Steps 1 and 2 involve implementing ML HMM filters at each time instant; each HMM filter involves a complexity of $O(S^2)$ computations.

B. Lovejoy's Approximation

In the worst case, the number of linear segments that characterize the piecewise linear value functions $\bar{J}_k(\pi)$ and $\underline{J}_k(\pi)$ (which lower and upper bound the optimal value function $J_k(\pi)$ defined in Section III-B) can grow exponentially as $O(L^{M(N-k)})$; hence, the associated computational costs of the near optimal sensor scheduling algorithms for quadratic cost functions can be prohibitive. It is obvious that by considering only a subset of the piecewise linear segments that characterize the upper bound value function $\underline{J}_k(\pi)$ [which is defined in (25)] and discarding the other segments, one can reduce the computational complexity. This is the basis of Lovejoy's [13] upper bound approximation.

Lovejoy's algorithm [13] operates as follows.

Step 1) Given $\tilde{\Gamma}_{k+1}$ and the piecewise linear upper bound costs $\underline{g}_r$ of (25), compute the set of vectors Γ_k using any of the POMDP algorithms described in Section III-A.

Step 2) Construct the set $\tilde{\Gamma}_k$ by pruning Γ_k as follows: Pick any $\bar{R}$ points $\pi_1, \pi_2, \ldots, \pi_{\bar{R}}$ in the information state simplex $\mathcal{S}$. Then, set

$$\tilde{\Gamma}_k = \{\arg\min_{\gamma\in\Gamma_k} \gamma'\pi_r, \quad r = 1, 2, \ldots, \bar{R}\}.$$

Step 3) $k \to k-1$.

The resulting value function $\tilde{J}_k(\pi) = \min_{\gamma\in\tilde{\Gamma}_k} \gamma'\pi$ is represented completely by $\bar{R}$ piecewise linear segments. Furthermore, as shown in [13], $\tilde{J}_k(\pi)$ is an upper bound to $\underline{J}_k(\pi)$, i.e.,

$$\underline{J}_k(\pi) \le \tilde{J}_k(\pi) \text{ for all } \pi \in \mathcal{P}.$$

Note that $\underline{J}_k(\pi)$ itself is an upper bound to the optimal value function $J_k(\pi)$, as described in Section III-B

Lovejoy's algorithm yields a suboptimal HMM sensor scheduling policy at an assured computational cost of no more than $\bar{R}$ evaluations per iteration k. Lovejoy also provides a constructive procedure for computing an upper bound to $\limsup_{\pi\in\mathcal{P}} |\underline{J}_k(\pi) - \tilde{J}_k(\pi)|$.

VI. Numerical Examples–Aircraft Identification Problem

The scenario involves an incoming aircraft where using various forms of sensors available at a base station; the task is to determine if the aircraft is a threat or not [5]. The choice of deciding between various sensors arises because the better sensors tend to make the location of the base station more easily identifiable or visible to the aircraft, whereas the more stealthy sensors tend to be more inaccurate. The sensors give information about the aircraft's type and distance, although the distance information is generally more reliable than the aircraft type information. We consider two scenarios.

1) Scenario 1 is a three-state HMM sensor scheduling problem and comprises two sensors and distance measurements.
2) Scenario 2 is a six-state HMM sensor scheduling problem and comprises of three sensors. The measurements comprise distance and aircraft-type measurements.

A. Dynamic Scheduling Between Active and Passive Sensors

State Space: The state space for this problem comprises how far the aircraft is currently from the base station discretized into three distinct distances $d_1 = 10$, $d_2 = 5$, and $d_3 = 1$. We have chosen a three-state model to graphically illustrate our results. (Scenario 2 considers a six-state model).

We now specify the transition probabilities of the Markov chain. Assume that in one unit of time, it is impossible for the distance of the aircraft to increase or to decrease by more than one discrete location. The probability that the aircraft remains at the same discrete location d_i is 0.8. Apart from when the aircraft is at a maximum distance d_1 or minimum distance d_3, the probability that the aircraft distance will increase by 1 discrete location equals the probability that the aircraft distance decreases by one discrete location where each probability is 0.1. With these assumptions, the transition probability matrix of X_k is

$$A = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix}. \tag{39}$$

Sensors: Assume that two sensors are available, i.e., $L = 2$.

- *Active:* This active sensor (e.g., radar) yields accurate measurements of the distance of the aircraft but renders the base more visible to the aircraft. Thus, the active sensor is merely an HMM state filter.
- *Predict:* Employ no sensor; predict the state of the aircraft. Thus, the predict sensor is merely an HMM state predictor.

At any time instant $k, u_k \in \{\texttt{active}, \texttt{predict}\}$ denotes which of the above two sensors is used.

Observation Symbols: When using the active sensor, the observation symbols at each time k consists of distance measurements to the base station d_1, d_2, and d_3. In addition, there is an additional observation symbol: nothing that results when the predict sensor is used (i.e., no observation is made). In terms of the notation of Section II, the number of possible observation symbols is $M = 4$ and $O_1 = d_1, O_2 = d_2, O_3 = d_3$, $O_4 = \texttt{nothing}$.

We assume that the distance the active sensor reports is never more than one discrete location away from the true distance. The active sensor will detect the true distance with probability p. The predict sensor only records the observation nothing with probability 1. In particular, defining the $N \times M$ matrix of symbol probabilities $\bar{B}(u_k) = [\bar{B}_{ij}(u_k)] = P\{y_k(u_k) = O_j | X_k = e_i\}$, we assign

$$\bar{B}(u_k = \texttt{active}) = \begin{bmatrix} p & 1-p & 0 & 0 \\ \frac{1-p}{2} & p & \frac{1-p}{2} & 0 \\ 0 & 1-p & p & 0 \end{bmatrix}$$

$$\bar{B}(u_k = \texttt{predict}) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Costs: Our cost function is given by (4) and comprises two components.

1) *Sensor Usage Costs:* The sensor costs are assigned the following values at each time instant: For $i \in \{1, \ldots, S\}$

$$c(X_k = e_i, u_{k+1} = \texttt{active}) = \frac{r^{\texttt{active}}}{d_i} + \rho^{\texttt{active}}$$

$$c(X_k = e_i, u_{k+1} = \texttt{predict}) = \frac{r^{\texttt{predict}}}{d_i} + \rho^{\texttt{predict}}. \tag{40}$$

We assume $\rho^{\texttt{active}} = 8$, $\rho^{\texttt{predict}} = 5$, meaning that the operating cost of using the active sensor is higher than the predict sensor. In addition, the cost incurred is inversely proportional to the distance of the aircraft. This reflects the fact that when the aircraft is close to the base station, the threat is greater. We chose the gains $r^{\texttt{active}} = 2$ and $r^{\texttt{predict}} = 5$.

2) *State Estimation Error Cost:* For the state estimation error cost component, we consider the l_2 cost $\alpha_k(1 - \pi_k' \pi_k)$. We chose $\alpha_k = 10$.

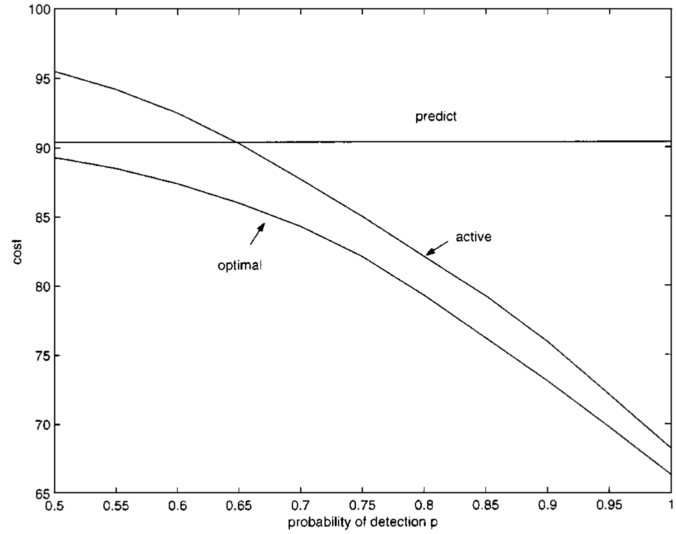


Fig. 3. Finite-horizon HMM sensor scheduling with two sensors: active and predict. The figure shows that dynamically switching between these two sensors according to the optimal sensor schedule results in a lower cost than using each sensor individually.

Results: With the aforementioned setup, we used the POMDP program available from the website in [6] to optimally solve the HMM sensor scheduling problem. The POMDP program solves the backward DP recursion at each time instant by outputting the the solution vector set Γ_k. However, the POMDP program is designed only for a linear cost. To deal with the piecewise linear cost function, we wrote a preprocessing program that, at each time k, takes Γ_k from the POMDP program and adds our piecewise linear cost function. The resulting augmented set of vectors is input to the POMDP program at the next time iteration $k-1$ (and so on).

All our simulations were run on a Pentium-2 400–MHz personal computer. The POMDP program allows the user to choose from several available algorithms. We used the "Incremental Pruning" algorithm developed by Cassandra *et al.* in 1997 [7]. This is currently one of the fastest known algorithms for solving POMDPs.

1) We ran the POMDP program for the previous parameters over a horizon of $N = 7$ for different values of p (probability of detection). In all cases, no *a priori* assumption was made on the initial distance (state) of the target; thus, we chose the information state (filtered density) at time 0 as $\pi_0 = [1/3,\ 1/3,\ 1/3]'$; see (1). We approximated the l_2 cost function by the $I = 3$ piecewise linear interpolation, as illustrated in Section III-B and Fig. 2, i.e., over $R = 9$ triangular patches.

Fig. 3 shows the costs incurred versus detection probability p if only the active or predict sensor is used alone. The cost incurred by dynamically switching between the active and predict sensors based on the optimal sensor schedule is also shown. It can be inferred

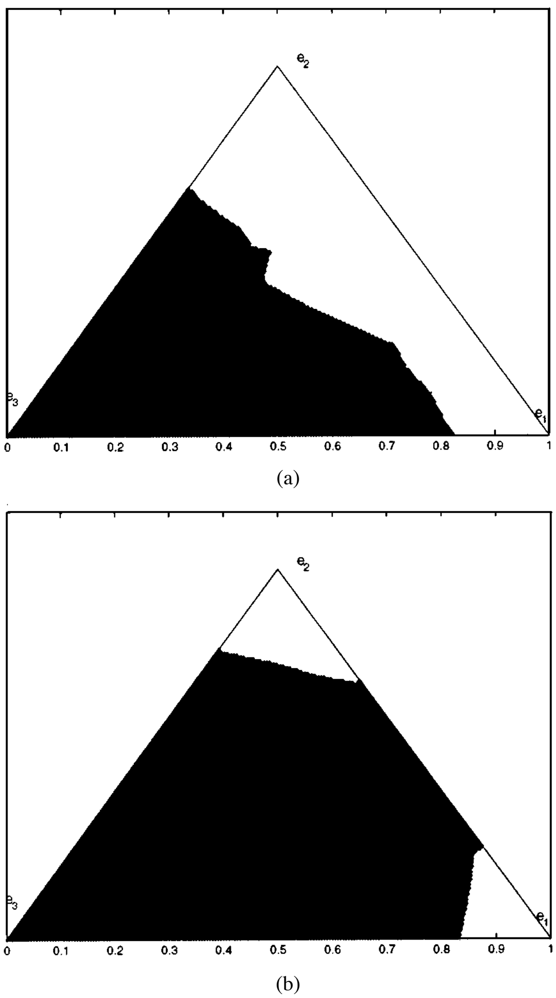


Fig. 4. Infinite-horizon HMM sensor scheduling. The optimal stationary policy is depicted on the information state simplex S. The shaded region denotes values of information state π for which it is optimal to use the active sensor. The clear region denotes values of π for which it is optimal to use the predict sensor. p denotes the probability of detection of the active sensor.

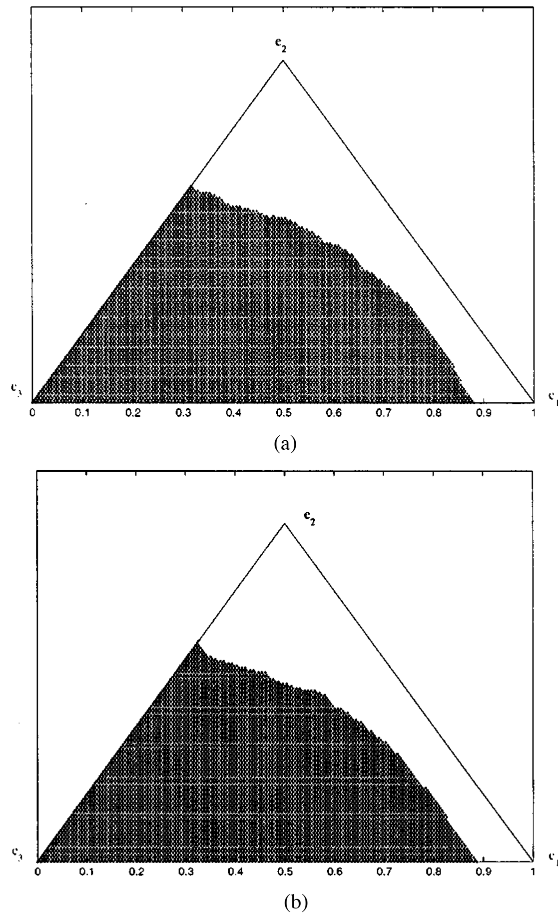


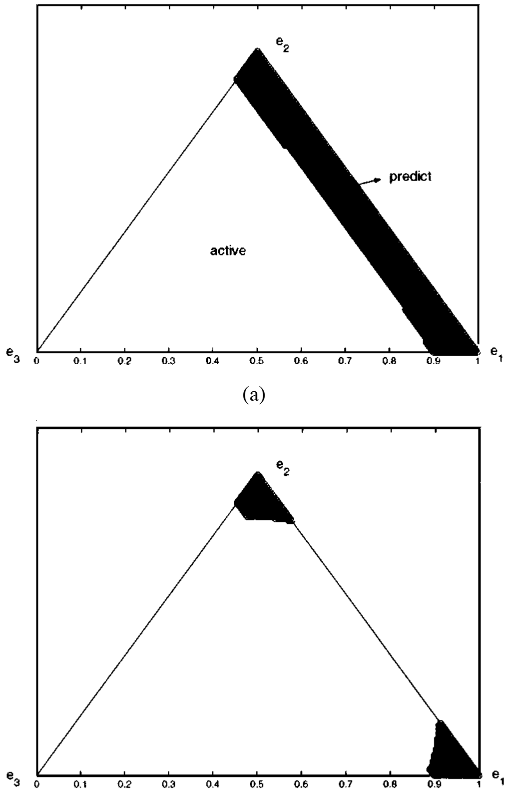
Fig. 5. Freudenthal interpolation accuracy for infinite horizon HMM scheduling. The stationary policy is shown for the $R = 64$ and $R = 100$ regions. The shaded region depicts $u_k = 1$ (active sensor).

from Fig. 3 that when the probability of detection is low ($0.5 < p < 0.65$), using the predict sensor alone does better than using the active sensor. The reason is that for low probability of detection, the active sensor (HMM filter) yields inaccurate state estimates. This, together with the fact that the active sensor has a higher operating cost than the predict sensor, means that for low p, it costs less to use an HMM predictor and not incur any cost obtaining extremely inaccurate measurements. Note that in all cases, the optimal sensor schedule (which dynamically selects between active and predict) incurs the smallest cost.

2) *Stationary HMM Scheduling:* Here, we consider the infinite horizon discounted cost function (35) with $\beta = 0.95$. We used the Freudenthal triangularization with $R = 9$ regions for the piecewise linear approximation of the quadratic cost function. The POMDP program (value iteration algorithm of Section IV-C) was run for a horizon $N = 100$. Fig. 4 shows the stationary scheduling policy on the information state simplex. The shaded region of the information simplex depicts usage of the active sensor, and the clear region depicts usage of the predict sensor. Fig. 4(a) shows that when the detection probability $p = 0.55$, it is optimal to use the active sensor (HMM state filter) when the target is closest to the base station (i.e., distance $= 1$ or $X_k = e_3$), whereas when the target is further away (e.g., $X_k = e_1$ or e_2), it is optimal to use the predict sensor (HMM state predictor). This makes intuitive sense since the closer the target is to the base station, the threat is greater, and the more accurate active sensor is used to track the target.

Fig. 4(b) shows that when the detection probability is $p = 0.80$, the active sensor is used most of the time; only when the information state π_k computed by the HMM filter is very close to e_1 or e_2 is the predict sensor used. The increased usage of the active sensor is due to the fact that the active sensor is now more accurate; hence, the mean square error state estimate is lower compared with the case $p = 0.55$.

3) *Interpolation accuracy:* We illustrate the performance of the HMM scheduling algorithm with increasing accuracy of the Freudenthal interpolation using the piecewise linear approximation $\min_r \bar{g}_r' \pi$ (see Section III-B). We ran the value iteration algorithm for the $I = 8$ interpolation ($R = 64$ regions) and $I = 10$ interpolation ($R = 100$ regions). Fig. 5 shows the optimal stationary scheduling policy as



(a)

(b)

Fig. 6. Optimal HMM sensor scheduling with quadratic constraints on estimation error. The cost minimized is the sensor usage cost. The shaded region depicts $u_k = 2$ (predict sensor), and the clear region depicts $u_k = 1$ (active sensor).

a function of the information state π. Fig. 5 shows that $R = 64$ and $R = 100$ regions yield similar stationary policies. We found that for $R > 100$, the change in the stationary policy is negligible.

Quadratic Constraints in State Estimation Error: Here, we consider the setup of Section IV-A with constraints on the mean square estimation error of the prediction sensor, as in (32). The cost function, which only includes sensor usage costs, is of the form (40). The predict sensor can only be used at time k, providing the mean square state estimation error at time $k+1$ is less than a specified value $K_2 = 0.45$, i.e., $1 - \pi' A A' \pi < K_2 = 0.45$. We chose $\rho^{\texttt{active}} = 6$, $\rho^{\texttt{predict}} = 5$, $r^{\texttt{active}} = 2$, and $r^{\texttt{predict}} = 5$. Fig. 6(a) shows the information simplex without constraints. The feasibility of the above constraint can be verified using Theorem 4.1. Fig. 6(a) shows the information state simplex and the optimal stationary policy for the unconstrained case; the shaded region indicates the values of π for which the predict sensor is optimal. When the above constraint is adopted on the predict region, Fig. 6(b) shows the information state simplex with the shaded region indicating the values of π for which the predict sensor is optimal, subject to the above constraint. It can be seen from Fig. 6(b) that the predict sensor can only be used at time k if the current HMM filtered density π_k is very close to either e_2 or e_1. For these shaded regions, it is guaranteed that if the predict sensor is used at time $k+1$, the mean square estimation error at time $k+1$ is less than $K_2 = 0.45$.

B. Dynamic Scheduling Between Active, Coarse, and Passive Sensors

State Space: The state space for this problem comprises of two main components:

- *Aircraft type:* The aircraft is either a friend or hostile.
- *Distance:* How far is the aircraft currently from the base station discretized into $S = 3$ distinct distances $d_1 = 10$, $d_2 = 5$, and $d_3 = 1$?

Thus, the total number of states are $2S = 6$. We represent the states of the $2S$ state Markov chain by the unit vectors $e_1, \ldots, e_{2S}$. Here, the states e_i, $1 \le i \le S$ correspond to the aircraft being at at distance d_i and friendly, whereas e_{i+S}, $1 \le i \le S$ corresponds to the aircraft being at a distance d_i and hostile.

Assume that the distance of the aircraft evolves according to the same transition probabilities as Scenario 1. The aircraft-type portion of the state never changes with time (i.e., a friendly aircraft cannot become a hostile aircraft and vice versa; hence, the transition probability matrix is block diagonal. With the above assumptions, the transition probability matrix of X_k is

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_1 \end{bmatrix}, \qquad A_1 = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix}. \tag{41}$$

Sensors: We assume that three sensors are available, i.e., $L = 3$.

- *Active:* This active sensor (e.g., radar) yields accurate measurements of the distance of the aircraft but renders the base more visible to the aircraft. The active sensor also yields less accurate information on the aircraft type (friend or hostile).
- *Passive:* This is a passive imaging sensor. It yields accurate information on the aircraft type but less accurate measurements of the aircraft distance. The passive sensor does not make the base station too visible to the incoming aircraft.
- *Predict:* Employ no sensor; predict the state of the aircraft.

Observation Symbols: The observation $y_k(u_k)$ at each time k consists of two independent components: the aircraft type (friend or hostile) and its distance to the base station d_i, $i \in \{1, \ldots, S\}$. In addition to these $2S$ possible symbols, there is an additional observation symbol; nothing that results when the predict sensor is used (i.e., no observation is made).

For simplicity, assume that the distance the sensors report is never more than one discrete location away from the true distance. The active sensor will detect the true distance with probability 0.95, whereas the passive sensor detects the true distance with probability $0.5 + p$, where p is a parameter we will vary. The remaining probability mass for both sensors are equally distributed among detecting the distance as being one location too close and one location too far.

The sensors' detection of the aircraft type (friend or hostile) is independent of the distances reported by the sensors. The active sensor will detect the correct type with probability $0.5 + p$, and the passive sensor succeeds with probability 0.95.

Costs: Our cost function is given by (4) and comprises two components.

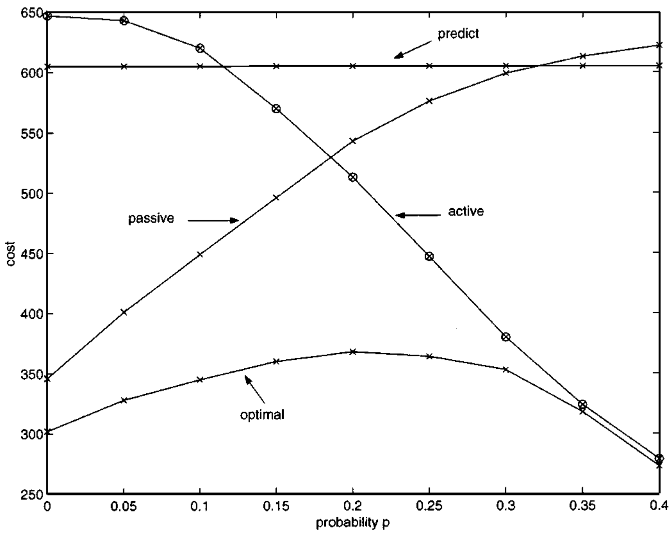


Fig. 7. Finite-horizon HMM sensor scheduling with three sensors: active, predict, and passive. The figure shows that dynamically switching between these three sensors according to the optimal sensor schedule results in a significantly lower cost than using any of the sensors individually.

i) The sensor usage costs are assigned the following values at each time instant: For $i \in \{1, \ldots, S\}$

$$\begin{aligned} c(X_k = e_i, u_{k+1} = \texttt{active}) &= \rho^{\texttt{active}} \\ c(X_k = e_{i+S}, u_{k+1} = \texttt{active}) &= \frac{r^{\texttt{active}}}{d_i} + \rho^{\texttt{active}} \\ c(X_k = e_i, u_{k+1} = \texttt{passive}) &= \rho^{\texttt{passive}} \\ c(X_k = e_{i+S}, u_{k+1} = \texttt{passive}) &= \frac{r^{\texttt{passive}}}{d_i} + \rho^{\texttt{passive}} \\ c(X_k = e_i, u_{k+1} = \texttt{predict}) &= \rho^{\texttt{predict}} \\ c(X_k = e_{i+S}, u_{k+1} = \texttt{predict}) &= \frac{r^{\texttt{predict}}}{d_i} + \rho^{\texttt{predict}} . \end{aligned}$$

Recall that $X_k = e_i, i \in 1, 2 \ldots, S$ denotes a friendly aircraft at distance d_i, whereas $X_k = e_{i+S}$ denotes a hostile aircraft at distance d_i. Thus, for a friendly aircraft, the cost of using a sensor is independent of the distance of the aircraft and only depends on the sensor. We assume $\rho^{\texttt{active}} > \rho^{\texttt{passive}} > \rho^{\texttt{predict}}$, meaning that an accurate sensor costs more than the coarse sensor, etc. For a hostile aircraft, the cost incurred is inversely proportional to the distance of the aircraft.

ii) For the state estimation error cost component, assume that $\alpha_k = 200, k = 0, 1, \ldots, N-1$, and $\alpha_N = 0$ in (4). We consider the state estimate error cost as the following seven-piece linear interpolation of the l_2 cost $\alpha_k(1 - \pi'\pi)$:

$$\|X_k - \pi_k\|_D = \begin{cases} g_i' \pi_k, & \text{if } \|e_i - \pi_k\|_\infty < 0.3, \quad i = 1, \ldots, 6 \\ 841\mathbf{1}'\pi_k, & \text{otherwise.} \end{cases} \tag{42}$$

Here, g_i^p is a six-dimensional vector with ith element 1 and all other elements 282.333.

For a horizon length of $N = 7$, Fig. 7 compares the performance of the optimal HMM sensor scheduling algorithm versus using the predict, passive, or active sensor alone.

VII. Conclusions and Future Work

We have presented optimal and suboptimal algorithms for the scheduling of sensors for (finite-state) hidden Markov models using a stochastic dynamic programming framework. One disadvantage of the algorithms proposed is that they require knowledge of the underlying parameters: transition and observation probabilities of the HMMs. In future work, we will look at efficient simulation-based algorithms based on neuro-dynamic programming [3]. In target tracking applications, it is often more appropriate to use jump Markov Linear systems (JMLSs) rather than HMMs. See [10] for some preliminary results and algorithms.

Acknowledgment

The author acknowledges [6] for providing freely available software for solving POMDPs. Part of this work was completed at the Department of Sensors, Signals, and Systems, Royal Institute of Technology, Sweden. The author thanks Prof. B. Wahlberg for several useful technical discussions.

References

[1] E. Altman, "Stochastic modeling," in *Constrained Markov Decision Processes*. London, U.K.: Chapman and Hall, 1999.
[2] M. Athans, "On the determination of optimal costly measurement strategies for linear stochastic systems," *Automatica*, vol. 8, pp. 397–412, 1972.
[3] D. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. 1 and 2.
[5] A. R. Cassandra, "Exact and approximate algorithms for partially observed markov decision process," Ph.D. dissertation, Brown Univ., Providence, RI, 1998.
[6] ——, Tony's POMDP page. [Online]. Available: http://www.cs.brown.edu/research/ai/pomdp/index.html.
[7] A. R. Cassandra, M. L. Littman, and N. L. Zhang, "Incremental pruning: A simple fast exact method for partially observed markov decision processes," in *Proc. 13th Annu. Conf. Uncertainty Artif. Intell.*, Providence, RI, 1997.
[8] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models – Estimation and Control*. New York: Springer-Verlag, 1995.
[9] J. S. Evans and V. Krishnamurthy, "Optimal sensor scheduling for hidden Markov model state estimation," *Int. J. Contr.*, vol. 74, no. 18, pp. 1737–1742, Dec. 2001.
[10] R. Evans, V. Krishnamurthy, and G. Nair, "Sensor adaptive target tracking over variable bandwidth networks," in *Model Identification and Adaptive Control – A Volume in Honor of Brian Anderson*, G. Goodwin, Ed. New York: Springer-Verlag, 2001, pp. 115–124.
[11] M. Hauskrecht, "Planning and control in stochastic domains with imperfect information," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1997.
[12] P. R. Kumar and P. Varaiya, *Stochastic Systems–Estimation, Identification, and Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
[13] W. S. Lovejoy, "Computationally feasible bounds for partially observed Markov decision processes," *Oper. Res.*, vol. 39, no. 1, pp. 162–175, Jan. 1991.
[14] ——, "A survey of algorithmic methods for partially observed Markov decision processes," *Ann. Oper. Res.*, vol. 28, pp. 47–66, 1991.
[15] D. G. Luenberger, *Linear and Nonlinear Programming*, Second ed. Reading, MA: Addison-Wesley, 1984.

[16] L. Meier, J. Perschon, and R. M. Dressier, "Optimal control of measurement systems," *IEEE Trans. Automat. Contr.*, vol. AC–12, pp. 528–536, Oct. 1967.

[17] G. E. Monahan, "A survey of partially observable Markov decision processes: theory, models and algorithms," *Manage. Sci.*, vol. 28, no. 1, Jan. 1982.

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

[19] A. Segall, "Optimal control of noisy finite-state Markov processes," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 179–186, Apr. 1977.

[20] R. D. Smallwood and E. J. Sondik, "Optimal control of partially observable Markov processes over a finite horizon," *Oper. Res.*, vol. 21, pp. 1071–1088, 1973.

[21] O. Tremois and J. P. Le Cadre, "Optimal observer trajectory in bearings-only tracking for maneuvering sources," *Proc. Inst. Elect. Eng.–Radar, Sonar, Navigat.*, vol. 146, no. 1, pp. 31–39, 1999.

Vikram Krishnamurthy (SM'99) was born in 1966. He received the B.E.E. degree from the University of Auckland, Auckland, New Zealand, in 1988 and the Ph.D. degree from the Australian National University, Canberra, in 1992.

He is currently a Professor with the Department of Electrical Engineering, University of Melbourne, Parkville, Australia, and serves as Deputy Head of the department. Starting August 2002, he will be a Professor with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada. His research interests span several areas including stochastic scheduling and network optimization, time-series analysis, and statistical signal processing. He is an Associate Editor of *Systems and Control Letters*.

Dr. Krishnamurthy is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He has served on the technical program committee of several conferences including the 37th IEEE Conference on Decision and Control, held in Tampa, FL, in 1998, and the IFAC Symposium on System Identification in 2000 (SYSID'2000), held in Santa Barbara, CA.