

Networked Sensor Management and Data Rate Control for Tracking Maneuvering Targets

Robin Evans, *Fellow, IEEE*, Vikram Krishnamurthy, *Fellow, IEEE*, Girish Nair, and Len Sciacca

Abstract—This paper presents sensor and data rate control algorithms for tracking maneuvering targets. The maneuvering target is modeled as a jump Markov linear system. We present novel extensions of the Interacting Multiple Model (IMM), Particle filter tracker, and Probabilistic Data Association (PDA) algorithms to handle sensor and data rate control. Numerical studies illustrate the performance of these sensor and data rate control algorithms.

Index Terms—IMM algorithm, particle filters, PDA algorithm, sensor selection, target tracking.

I. INTRODUCTION

TRACKING a maneuvering target is often formulated as a problem of estimating the state of a partially observed jump Markov linear system. For example, consider a radar track-while-scan system where noisy range and bearing and possibly Doppler measurements are obtained at regular intervals (the radar scan rate) on all detectable objects including objects of interest termed targets such as aircraft or ships, and nuisance objects termed clutter, e.g., clouds, terrain, trees, etc. The purpose of a target tracking system is to search through these measurements in real time looking for trails (time space sequences) of measurements that correspond to targets and then to track these targets, updating estimates of their kinematic parameters such as range, speed and heading as new measurements arrive. False tracks can arise from consistent trails of noise and clutter and an important measure of a tracking algorithm's performance is its ability to minimize the number and duration of false tracks while rapidly and reliably finding and following real targets.

A typical multitarget tracking system [2], [5] contain three important subsystems: i) a *track initiation procedure*, which detects real targets and initializes track state estimators, ii) a *track state estimator*, which generates estimates of current and future

values of the target kinematic parameters (based on the available history of noisy measurements), and iii) a *data association process*, which allocates measurements to existing tracks and hopefully discards false measurements that do not correspond to real targets.

Increasingly, target tracking systems attempt to combine the above three processes into a single integrated algorithm as follows. 1) The trajectories of targets of interest are modeled by a Markov sequence based on target kinematics. 2) The “existence” behavior of real targets, i.e., the probabilities of real targets appearing and disappearing, is modeled by a finite-state Markov chain with state representing the probability of target's existence. 3) Finally, statistical measurement models are employed to describe both real target measurements and false (clutter) measurements. The optimal estimator for this “complete” target and measurement model is then approximated in some computationally feasible fashion. Target tracking algorithms such as Integrated Probabilistic Data Association (IPDA) [22], MHT [25], and Hidden Markov Model (HMM)-based trackers [29] are all based on approximate solutions to this complete target and measurement model estimation problem.

Here, we consider a further enhancement and integration of tracking system functionality by extending existing algorithms to handle flexible multimode measurement sensors and variable bandwidth data links between the sensors and the tracker (tracking computer). Flexible multimode sensors are capable of rapidly switching between various measurement modes, e.g., radar transmit waveforms, beam pointing directions, etc, so that the tracking system is able to tell the sensor which sensor mode to use at the next measurement time. Significant improvements in tracking performance have been demonstrated for such systems even with relatively simple tracker driven sensor measurement mode selection (control) strategies [12], [13]. Similarly when sensors are connected to a tracker via a data-link with low and/or time-varying bandwidth data rate, the tracker must dynamically decide which sensor data is the most valuable to transfer to the tracking system via the currently available data link capacity during each measurement interval. We distinguish two cases. If the data link is only one-way, from the sensor to the tracker, then optimal use of the available communication capacity must be determined by the sensor without direct knowledge of the current tracker state [23]. Alternatively, if there is a two-way link between the tracker and the sensor then the tracker can control the sensor and instruct the sensor as to what data to send, based on current tracking performance. This latter situation is clearly very similar to the flexible sensor mode control situation just described.

Manuscript received December 24, 2002; revised May 13, 2004. This work was supported by ARC, DSTO, DARPA, NSERC, and NICTA. Portions of this work appeared in the proceedings of the workshop “Model Identification and Adaptive Control,” edited by G. C. Goodwin, Springer Verlag, pp. 115–124, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrew C. Singer.

R. Evans is with National ICT Australia Victoria Research Laboratory, University of Melbourne, Victoria 3010, Australia.

G. Nair is with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: g.nair@ee.mu.oz.au).

V. Krishnamurthy is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada (e-mail: vikramk@ece.ubc.ca).

L. Sciacca is with the Electronic Warfare and Radar Division, Systems Sciences Laboratory, Defense Science and Technology Organization, Canberra 5111, Australia (e-mail: len.sciacca@dsto.defence.gov.au).

Digital Object Identifier 10.1109/TSP.2005.847829

Below, we focus on the flexible sensor and two-way data link problem whereby the tracker is able to control the sensor mode and/or the data link usage. This is shown, not surprisingly, to be a stochastic control problem. Furthermore, since the tracker controls the sensor mode and/or data link usage to optimize tracking performance without direct measurement of the tracking performance (the tracker does not know the true target state), then we have a *partially observed stochastic control* problem [3]. Thus, in this paper, we formulate and solve the problem of sensor and data-rate control for optimal tracking of maneuvering targets as a partially observed stochastic control problem for a jump Markov linear systems (JMLS).

JMLS have been widely used in areas such as target tracking to model the maneuvering behavior of real tracks [9], [15] and CDMA spread spectrum telecommunication systems [17]. JMLS are a significant generalization of the concept of an HMM in that the key assumption of conditional independence of the observations given the state is relaxed. Moreover, JMLS are a generalization of linear Gauss–Markov models since they allow the parameters of the system to evolve randomly, governed by a finite state Markov chain with known transition probabilities. Unfortunately, unlike an HMM or a Gauss–Markov model where optimal state estimation can be performed via the HMM filter and the Kalman filter, respectively, computing the optimal (conditional mean) state estimate of a JMLS requires exponential complexity; see [7] and [19] for an exposition of optimal and suboptimal state estimation algorithms for JMLS. Importantly, however, the optimal filtered state estimate—called the “information state”—is essential for computation of the optimal solution to the partially observed stochastic sensor/data rate control problem described above. This paper presents optimal and suboptimal solutions to this problem.

Before diving into technical details, we now briefly describe two practical applications that serve to further illustrate and motivate the ideas described above.

Example 1—Beam Scheduling in Electronically Scanned Arrays: Consider a problem where P targets (e.g., aircraft) are being tracked by an electronically scanned antenna array (ESA) with one steerable beam. The kinematic state of each target $x_t^{(p)}$, $p = 1, 2, \dots, P$ evolves independently of other targets according to a JMLS. The goal is to obtain the “best in some sense” (conditional mean) estimate of the state of all targets, however since there is only one steerable beam, we can obtain noisy measurements $y_t^{(p)}$ of **only one target** at any given time instant. Thus, we must face the scheduling or control problem of *which single target should the tracker choose to observe at each time instant in order to optimize some specified cost function?* This is precisely the flexible sensor situation described above and a “solution” to this problem forms an integral part of any real ESA antenna beam pointing system [4], [5]. Practical implementation of real ESA systems is, of course, quite complex since the beam scheduling function must handle a variety of search, identification and tracking functions. Here we have only described a small but nevertheless important part of the real ESA control problem. Fig. 1 shows a schematic representation of the situation we have in mind. As shown in [16], the above problem is a special case of a sensor selection problem for JMLS where the stochastic control problem has a multiarmed bandit structure.

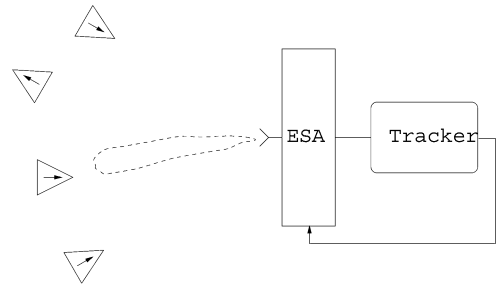


Fig. 1. Multitarget tracking with one multimode sensor.

It is well known that the multiarm bandit problem has a rich structure which results in the selection problem decoupling into P independent optimization problems. Indeed, it turns out that the optimal policy (control law) has an *indexable rule* [28]: For each target p , there is a function $l^{(p)}(x_t^{(p)})$ called the *Gittins index*, which is a function only of p and the information state $x_t^{(p)}$ (or in the partially observed case the posterior probability of $x_t^{(p)}$ given the history of measurements $y_t^{(p)}$). The optimal¹ beam steering policy at time t is to steer the beam toward the target with the smallest Gittins index, i.e., steer the beam toward target q , where

$$q = \min_{p \in \{1, \dots, P\}} \left\{ l^{(p)}(x_t^{(p)}) \right\}. \quad (1)$$

Example 2—Data-Rate Limited Target Tracking: A second motivating problem is illustrated in Fig. 2. Several sensors are connected to a multiplexer, which in turn is connected via a limited and/or time-varying data rate communication channel to a computer which performs target tracking based on observations provided by the sensors [26]. Clearly, if the channel data rate is adequate to allow all the sensor data to reach the tracking computer in a timely manner then we have a straightforward multisensor tracking problem that can be solved using standard methods. However, if the channel data rate is limited then we must decide how best to allocate the sensor data to fit the available communication capacity of the channel so that “optimal” tracking under the data rate constraint is achieved. For example, consider a scenario where 20 sensors each capable of making 16 bit measurements every 1 ms (i.e., 320 kbits/s sensor data) are connected to a multiplexer which is connected to the tracking computer via a 32 kbits/sec communication channel. The required 10:1 data rate reduction can be achieved by any number of static or dynamic strategies such as, for example, only sending the most significant bit from each sensor (20 kbits/s), or sending 16 bit data from two different sensors during each 1-ms time slot resulting in (32 kbits/s), and so on. We now describe an approach to this problem based on the sensor selection ideas discussed above. We create a set of *virtual sensors* each supplying information within the data rate limit. The data rate constrained tracking problem then becomes a problem of switching (at each time slot) between these virtual sensors, i.e., a sensor selection problem. Continuing with the example above we might create virtual sensor 1 by taking the two most significant bits from real sensors 1 and 2, taking the 11 most significant bits from real

¹The optimality is with respect to a discounted infinite horizon cost function; see [16]

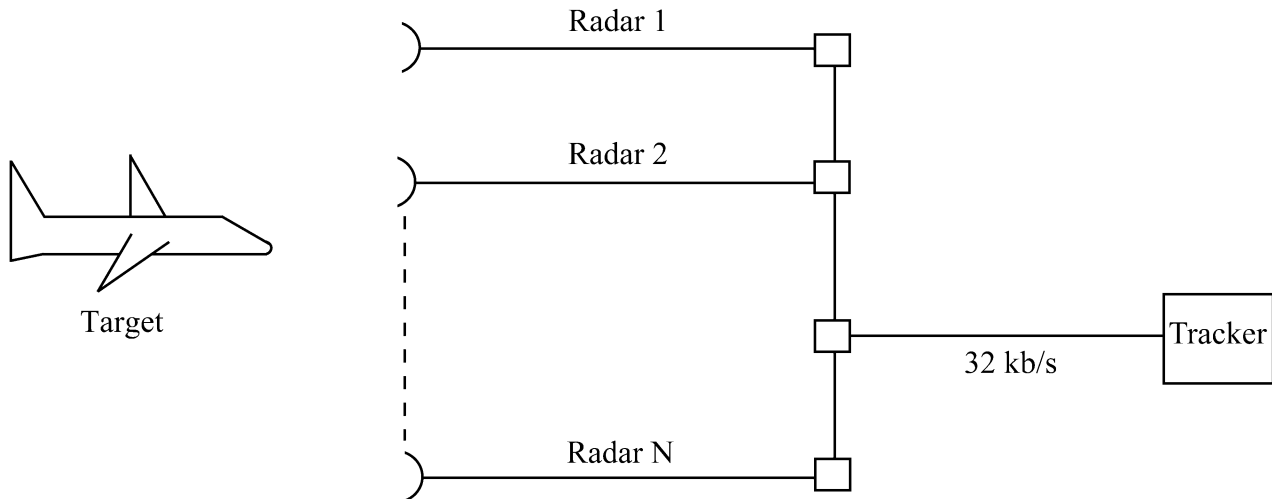


Fig. 2. Bandlimited tracking.

sensor 6 and taking the most significant bit from the other 17 real sensors. Thus, virtual sensor 1 consists of 32 bits of data taken in different measure from all the sensors and meeting the 32-kbits/s data rate limit. Proceeding in this way, we can create a set (possibly very large) of virtual sensors each supplying data at an acceptable rate. Thus, the problem for the tracker to dynamically choose between virtual sensors in an optimal fashion and transmit the choice of virtual sensor to be used for the next measurement to the real sensors so that after a measurement is made they can create the required virtual sensor data packet and send it back to the tracker over the channel (which now has exactly the right capacity to carry the virtual sensor data).

This approach centralises decisions at the tracker and is most suitable when there is limited computational power available at the sensors (they only have to pack the measured data into the specified virtual sensor packet). If there are no such computational constraints at the measurement sensors, then clearly a better approach is to perform additional processing at the sensors, using the real sensor measurements as well as any feedback information received from the tracker. In the single sensor case, feedback from the tracker is redundant since all the data available to it originates at that sensor in the first place, and hence the optimal scheme locates all decisions at the sensor. This is further explored in Section IV-E.

This paper presents optimal and practically implementable suboptimal algorithms which integrate sensor and/or data link control into a maneuvering target tracker. The optimal sensor adaptive algorithm is presented in terms of stochastic dynamic programming for a finite horizon cost function. The suboptimal sensor control algorithms we present are based on two approximations: i) Instead of optimizing a finite horizon cost function, we show that satisfactory results can be obtained by optimizing the expected instantaneous cost at each time, and ii) approximating the optimal state filter by the Interacting Multiple Model (IMM) algorithm (or some other suboptimal algorithm).

The remainder of this paper is organized as follows. In Section II, the sensor control tracking problem is formulated. Section III presents special cases where optimal processing is possible. In Section IV, a suboptimal one-step ahead algorithm is

presented and new Sensor Adaptive Interacting Multiple Model (SA-IMM), Sensor Adaptive particle filter, and Sensor Adaptive Probabilistic Data Association (SA-PDA) trackers are described. Section V presents two numerical examples.

II. PROBLEM FORMULATION

A. Target and Sensor Model

Let $t = 1, 2, \dots$ denote discrete time. The target/sensor (signal) model consists of the following ingredients. To simplify the presentation we consider only single target tracking and we will not model target existence behavior.

- a) **Target Mode:** In order to model target maneuvers the underlying target trajectory model jumps between a finite set of known modes representing various trajectory models such as constant speed and heading, constant heading with linear acceleration, coordinated turn, etc. This mode jumping can be modeled by a finite state Markov chain r_t . Let r_t denote a discrete-time, time-homogeneous, s -state, first-order Markov chain with state space $\{e_1, \dots, e_s\}$, where e_i denotes the unit s dimensional vector with 1 in the i th entry. Denote the transition probabilities as $p_{m,n} = \mathbf{P}\{r_{t+1} = n | r_t = m\}$ for any $m, n \in S$, where $S = \{1, 2, \dots, s\}$. The transition probability matrix $\mathcal{P} = [p_{m,n}]$ is, thus, an $s \times s$ stochastic matrix with elements satisfying $p_{m,n} \geq 0$ and $\sum_{n=1}^s p_{m,n} = 1$, for each $m \in S$. Denote the initial probability distribution by $p_m = \mathbf{P}\{r_1 = m\}$, for $m \in S$, such that $p_m \geq 0$, $\forall m \in S$ and $\sum_{m=1}^s p_m = 1$.
- b) **Target Kinematic State:** Target trajectories of interest are modeled by a process $\{x_t\}$, where typically, x_t is the target kinematic state (position, velocity) in Cartesian coordinates. Because the target model jumps according to the modes r_t , the target state x_t can be modeled as the following Jump Markov Linear System (JMLS):

$$x_{t+1} = A(r_{t+1})x_t + B(r_{t+1})w_t \quad (2)$$

where $x_t \in \mathbb{R}^{n_x}$ is the system state, and $w_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, I_{n_w}) \in \mathbb{R}^{n_w}$. $A(\cdot)$ and $B(\cdot)$ are functions of the Markov chain state r_t , i.e., $(A(\cdot), B(\cdot)) \subset \{(A(m), B(m)); m \in S\}$, and they evolve according to the realization of the finite state Markov chain r_t . We assume that $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$, where $P_0 > 0$, and let x_0 and w_t be mutually independent for all t .

- c) **Sensor Choice and Observation Process:** Assume that at time t there are L_t possible sensors available that can be used to give measurements of the target's state x_t . However, at each time instant, only one of these L_t possible sensors can be chosen. The variable $u_t \in \{1, \dots, L_t\}$ denotes the sensor selected at time t from among the L_t possible sensors. u_t will be chosen to minimize a cost function as described later. For notational simplicity, we will ignore the time dependence t of L_t .

Let the vector $y_t(u_t) \in \mathbb{R}^{n_y}$ denote the observation vector at time t if sensor $u_t \in \{1, \dots, L\}$ was chosen. In tracking problems, as well as the "true" observation $y_t(u_t)$, several false measurements are also obtained by sensor u_t . These false measurements are typically modeled as detections which are uniformly distributed throughout the observation span of a particular sensor. For example, a range measuring sensor might detect a real target present at a particular range with a certain detection probability and will report this range with an additive measurement error. The sensor will also report other detections (e.g., clutter, noise) at other ranges. These false measurements are assumed to be uniformly distributed over the full measurement range of the sensor and the probability of seeing a false range measurement at any particular range is governed by a Poisson arrival process. In this paper, we concentrate mostly on the no false measurements case and briefly consider false measurements in Section IV-C.

In general, the real target observation probability density $p(y_t|x_t, r_t, u_t)$ of a given sensor at time t could be non-Gaussian. However, to simplify our presentation, in this paper, we will use the following linear Gaussian sensor model

$$y_t(u_t) = C(r_t, u_t)x_t + D(r_t, u_t)v_t \quad (3)$$

where $v_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, I_{n_v}) \in \mathbb{R}^{n_v}$ are i.i.d. Gaussian sequences, and

$$D(i, j)D'(i, j) > 0, \quad \forall i \in S, \quad j \in \{1, 2, \dots, L\} \quad (4)$$

i.e., positive definite. We assume that x_0 , w_t and v_t are mutually independent for all t .

Equations (2) and (3) denote a controlled partially observed jump Markov linear system. The model parameters

$$\lambda = \{p_m, p_{mn}, A(m), B(m), C(m, u_t), D(m, u_t), \hat{x}_0, P_0; m, n \in S\}$$

are assumed known. This is a reasonable assumption in most tracking problems.

Note that if the realization of the Markov chain sequence r_t is known (or if $s = 1$) and there is only one sensor, then (2) and (3) reduce to a linear Gaussian state space model and the Kalman filter yields optimal (conditional mean) estimates of the the target state. On the other hand, if $L = 1$, $A(r_t) = I$, $B(r_t) = 0$, then y_t in (3) specializes to a standard Hidden Markov Model—i.e., the finite state Markov chain r_t observed in i.i.d. noise.

B. Sensor Adaptive Target Tracking Problem

Let

$$Y_t = \{u_1, u_2, \dots, u_t, y_1(u_1), y_2(u_2), \dots, y_t(u_t)\} \quad (5)$$

so that Y_t represents the information available at time t on which to base state estimates and sensor selection decisions. The sensor adaptive tracking problem proceeds in three stages for each $t = 0, 1, \dots, N - 1$, where N is a fixed positive integer.

- 1) **Selection:** Based on Y_t , we generate $u_{t+1} = \mu_{t+1}(Y_t)$, which determines which sensor is to be used at the next measurement time.
- 2) **Observation:** We then observe $y_{t+1}(u_{t+1})$, where u_{t+1} is the sensor selected in the previous stage.
- 3) **Estimation:** After observing $y_{t+1}(u_{t+1})$, we generate our best estimate \hat{x}_{t+1} of the state of the Markov chain x_{t+1} as $\hat{x}_{t+1} = E\{x_{t+1}|Y_{t+1}\}$. Note that the state estimate \hat{x}_{t+1} is dependent on the sequence of sensors selected from time 1 to $t + 1$, i.e., u_1, \dots, u_{t+1} (since it depends on Y_{t+1}).

We define the sensor sequence

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

and say that sequences are admissible if $u_{t+1} = \mu_{t+1}(Y_t)$, where

$$\mu_{t+1} : Y_t \rightarrow \{1, \dots, M\}$$

is a sequence of functions.

We assume the following cost is associated with estimation errors and with the particular sensor sequence chosen. If based on the observation at time t , the decision is made to choose $u_{t+1} = l$ (i.e., to choose the l th sensor at time $t + 1$, where $l \in \{1, \dots, L\}$), then the instantaneous cost incurred at time t is

$$\alpha_t(l) \|x_t - \hat{x}_t\|^2 + \beta_t(l) \|r_t - \hat{r}_t\|^2 + \gamma_t(x_t, r_t, l). \quad (6)$$

Here, $\alpha_t(l)$ and $\beta_t(l)$, $l = 1, 2, \dots, L$ are known nonnegative scalar weights. $\hat{x}_t = E\{x_t|Y_t\}$ and $\hat{r}_t = E\{r_t|Y_t\}$ denote the optimal (conditional mean) estimates of x_t and r_t given the information sequence Y_t —note that these depend on the sequence μ . $\|x_t - \hat{x}_t\|^2$ denotes the mean square state estimation error in the target trajectory at time t due to the choice of sensor sequence u_1, \dots, u_t . Thus, the first two terms denote the square of the tracking error estimate generated by sensor l . Finally,

$\gamma_t(x_t, r_t, l) \geq 0$ denotes the instantaneous cost of using the sensor $u_{t+1} = l$ when the target's mode is r_t and the target's state is x_t .

Our aim is to find the sensor sequence that minimizes the total accumulated cost J_μ over the set of admissible sensor selection rules, where

$$J_\mu = E \left\{ \sum_{t=1}^N \alpha_t(u_{t+1}) \|x_t - \hat{x}_t\|^2 + \beta_t(u_{t+1}) \|r_t - \hat{r}_t\|^2 + \sum_{t=0}^{N-1} \gamma_t(x_t, r_t, u_{t+1}) \right\} \quad (7)$$

where $u_{t+1} = \mu_{t+1}(Y_t)$. Note that the above expectation is with respect to (x_0, r_0) (which has prior distribution $\pi_0^{x_0, r_0}$), $v_1, \dots, v_N, w_0, \dots, w_{N-1}, r_0, \dots, r_{N-1}$; see, for example, [3, pp. 211].

The terms $\alpha_t(l)$ and $\beta_t(l)$ allow different sensors $l \in \{1, 2, \dots, L\}$ to be weighted differently, and the time index in α_t allows us to weight the state estimate errors over time. The above problem is a finite horizon partially observed stochastic control problem. Infinite horizon problems with discounted cost can be formulated similarly.

C. Information State Formulation

Define the information state $\pi_t^{x_t, r_t}$ as the conditional density of x_t, r_t given the history Y_t , i.e.,

$$\pi_t^{x_t, r_t}(x, i) \triangleq p(x_t = x, r_t = e_i | Y_t) \quad \text{where } x \in \mathbb{R}^{n_x}, i \in S = \{1, \dots, s\}. \quad (8)$$

To simplify notation in the sequel define the probability density functions

$$\begin{aligned} \pi_t^{x_t}(x) &\triangleq p(x_t = x | Y_t) = \sum_{i=1}^s \pi_t^{x_t, r_t}(x, i) \\ \pi_t^{r_t}(i) &\triangleq \mathbf{P}(r_t = e_i | Y_t) = \int_{\mathbb{R}^{n_x}} \pi_t^{x_t, r_t}(x, i) dx \\ \pi_t^{r_t} &\triangleq \begin{bmatrix} \mathbf{P}(r_t = e_1 | Y_t) \\ \vdots \\ \mathbf{P}(r_t = e_s | Y_t) \end{bmatrix}. \end{aligned}$$

Using Bayes' rule, it can be shown that the information state (filtered density) evolves as

$$\begin{aligned} \pi_{t+1}^{x_{t+1}, r_{t+1}} &= \frac{1}{K_t} p(y_{t+1} | r_{t+1}, x_{t+1}, u_{t+1}) \\ &\times \int_{\mathbb{R}^{n_x}} \sum_{m=1}^s p(x_{t+1}, r_{t+1} | x_t = x, r_t = m) \\ &\times \pi_t^{x_t, r_t}(x, m) dx \end{aligned} \quad (9)$$

where K_t denotes a normalization term. For ease of notation, we express the above filtered density update (9) as

$$\pi_{t+1}^{x_{t+1}, r_{t+1}} = T_t(\pi_t^{x_t, r_t}, y_{t+1}, u_{t+1}). \quad (10)$$

Defining $Q(r_t) = C(r_t)C'(r_t)$ and $\Sigma(r_t, u_t) = D(r_t, u_t)D'(r_t, u_t)$ for the JMLS (2) and (3), the various terms in (9) are

$$\begin{aligned} p(y_{t+1} | r_{t+1}, x_{t+1}, u_{t+1}) &\sim \mathcal{N}(C(r_{t+1}, u_{t+1}) \\ &\quad \Sigma(r_{t+1}, u_{t+1})) \\ p(x_{t+1}, r_{t+1} | x_t = x, r_t = m) &= p_{m, r_{t+1}} \mathcal{N}(A(r_{t+1})x \\ &\quad Q(r_{t+1})). \end{aligned} \quad (11)$$

The information state is a sufficient statistic for Y_t . Thus, the admissible policies (sensor selection rules) satisfy $\mu_{t+1} : \pi^{x_t, r_t} \rightarrow \{1, \dots, L\}$.

In terms of the information state, the cost function (7) can be re-expressed as

$$\begin{aligned} J_\mu &= E \left\{ \sum_{t=1}^N \alpha_t(u_{t+1}) \left[\langle x_t^2, \pi_t^{x_t} \rangle - \langle x_t, \pi_t^{x_t} \rangle^2 \right] \right. \\ &\quad + \beta_t(u_{t+1}) (1 - \pi_t^{r_t} \pi_t^{r_t}) \\ &\quad \left. + \sum_{t=0}^{N-1} \int_{\mathbb{R}^{n_x}} \sum_{i=1}^s \gamma_t(x, i, u_{t+1}) \pi_t^{x_t, r_t}(x, i) dx \right\} \\ &\triangleq E \left\{ \sum_{t=0}^N \epsilon_t(u_{t+1}, \pi_t^{x_t, r_t}) \right\} \end{aligned} \quad (12)$$

where E is as in (7), $\langle x_t^2, \pi_t^{x_t} \rangle = \int_{\mathbb{R}^{n_x}} x' x \pi_t^{x_t}(x) dx$ and with $x_t = [x_t(1), \dots, x_t(n_x)]'$ and

$$\begin{aligned} \langle x_t, \pi_t^{x_t} \rangle &= \left[\int_{\mathbb{R}} x_t(1) \pi_t^{x_t}(x_t(1)) dx_t(1), \right. \\ &\quad \left. \dots, \int_{\mathbb{R}} x_t(n_x) \pi_t^{x_t}(x_t(n_x)) dx_t(1) \right]' \\ &\in \mathbb{R}^{n_x} \end{aligned}$$

In addition, $\langle x_t, \pi_t^{x_t} \rangle^2 = \langle x_t, \pi_t^{x_t} \rangle' \langle x_t, \pi_t^{x_t} \rangle$.

For the JMLS (2) and (3), the information state $\pi_t^{x_t, r_t}$ is a Gaussian mixture with s^t components. Computing $\pi_t^{x_t, r_t}$ and, hence, also $E\{x_t | Y_t\}$ or $E\{r_t | Y_t\}$ involve prohibitive computational cost that is exponential in t . Thus, in general, the solution to the sensor adaptive tracking problem for a JMLS target model is not practical without some sort of approximation.

III. OPTIMAL ALGORITHMS

A. Stochastic Dynamic Programming

Based on the above formulation, the solution to the optimal sensor selection problem is obtained using the Dynamic Programming algorithm [3], which proceeds backward in time from $t = N$ to $t = 0$:

$$V_N(\pi^{x_N, r_N}) = \epsilon_N(\pi^{x_N, r_N}) \quad (13)$$

$$\begin{aligned} V_t(\pi^{x_t, r_t}) &= \min_{u \in \{1, \dots, L\}} \left[\epsilon_t(u, \pi_t^{x_t, r_t}) \right. \\ &\quad \left. + E_{y_{t+1}} \{V_{t+1}(T(\pi_t^{x_t, r_t}, y_{t+1}, u))\} \right]. \end{aligned} \quad (14)$$

The optimal cost starting from the initial condition π_0 is given by $V_0(\pi_0)$ and if $u_{t+1}^* = \mu_{t+1}^*(\pi_t)$ minimizes the right hand side of (14) for each t and each π_t , the selection policy

$$\mu^* = \{\mu_1^*, \mu_2^*, \dots, \mu_N^*\}$$

is optimal.

The optimality of the above DP algorithm (14) follows from [11, Th. 3.2.1]. In particular, the measurable selection [11, cond. 3.3.2] is trivially satisfied (trivial because the control space is finite) as long as γ_t in (7) is lower semi-continuous.

In general, the above DP recursion (14) does not translate directly into practical solution methodologies. The fundamental problem with (14) is that at each iteration t , one needs to compute $V_t(\pi^{x_t, r_t})$ over an uncountable set of states π^{x_t, r_t} . However, in two special cases, $V_t(\pi^{x_t, r_t})$ has a finite dimensional characterization.

B. Linear Quadratic Gaussian Sensor Selection

If the sequence $\{r_t\}$ is known (or $s = 1$) and $\gamma_t(x_t, r_t, u_{t+1}) = c_t(u_{t+1})$, then computing the optimal policy μ is equivalent to the following nonlinear deterministic control problem [21]. Minimize

$$J_\mu = \sum_{t=1}^N \text{tr}[P_{t|t}] + \sum_{t=0}^{N-1} r_t(u_{t+1}) \quad (15)$$

subject to the constraint that

$$\begin{aligned} P_{t|t} &= P_{t|t-1} - P_{t|t-1}C'(r_t, u_t) \\ &\quad \times \left[C(r_t, u_t)P_{t|t-1}C'(r_t, u_t) \right. \\ &\quad \left. + D(r_t, u_t)D'(r_t, u_t) \right]^{-1} \\ &\quad \times C(r_t, u_t)P_{t|t-1} \end{aligned} \quad (16)$$

$$\begin{aligned} P_{t|t-1} &= B(r_t)B'(r_t) + A(r_t)P_{t-1|t-1}A'(r_t) \\ t &= 1, \dots, N. \end{aligned} \quad (17)$$

The above equations are merely the covariance update equation of a Kalman filter. Thus, in the linear Gaussian quadratic case, the optimal sensor selection can be computed off-line independently of the data. This is not surprising; since the Kalman filter covariance is independent of the observation sequence.

C. HMM Sensor Selection

If $A(r_t) = I$ and $B = 0$, then (2), (3) reduces to a partially observed Markov decision process (POMDP). Suppose further that v_t is finite state i.i.d noise so that the measured output y_t belongs to a finite number of symbols, say $\{O_1, \dots, O_M\}$. Then, in the case $\alpha = \beta = 0$, the optimal sensor sequence can be exactly determined by a finite dimensional algorithm. This finite dimensional characterization was first derived in [27], where it was shown that the value function $V_t(\pi)$ is piecewise linear and convex. Several linear programming based algorithms have been presented in the operations research literature for computing these piecewise linear segments. These include Sondik's algorithm [27], Monahan's algorithm, Cheng's algorithm [20], and the Witness algorithm [6]. See the website

<http://www.cs.brown.edu/research/ai/pomdp/index.html> for an excellent tutorial exposition with graphics of these various algorithms.

In the case when α is nonzero, near-optimal policies can be developed as shown in [14]. As described in [14], the quadratic cost $\alpha_t(u)\|x_t - \hat{x}_t\|^2$ can be approximated by piecewise linear segments, and the optimal control policy determined for this piecewise linear cost using the POMDP algorithms described in the previous paragraph.

IV. SUBOPTIMAL ALGORITHMS

In practice, the algorithms presented above are computationally too demanding for real-time on-line implementation. As mentioned above optimal state estimation for a JMLS is an NP hard problem, and moreover, it is difficult to include false measurements. In this section, we describe a one-step ahead sub-optimal solution to the sensor selection problem with multiple jumping models. The idea behind the one-step ahead algorithm is to compute the expected posterior density for the target state for each sensor based on the current posterior density and the known measurement models, then make a measurement using the sensor that gave the best predicted cost. The idea is illustrated below for the no false measurement case. The false measurement case is briefly described in Section IV-C.

A. One-Step Ahead Algorithm

To simplify our notation, we will assume that the weights α and β in (7) are constants. The one-step ahead algorithm proceeds recursively as follows: Assume that $\pi_{t-1}^{x_{t-1}, r_{t-1}}$ (or a suitable approximate value) has been computed at time $t-1$.

Step 1: Minimize at time t the instantaneous cost defined in (6), which can be re-expressed in terms of the information state as

$$\begin{aligned} J_{\mu_t} &= \alpha \left[\langle x^2, \pi_t^{x_t} \rangle - \langle x, \pi_t^{x_t} \rangle^2 \right] \\ &\quad + \beta \left(1 - \pi_t^{r_t'} \pi_t^{r_t} \right) + \langle \gamma'_t(\cdot, \cdot, u_t), \pi_t^{x_t, r_t} \rangle \end{aligned} \quad (18)$$

subject to $\pi_t^{x_t, r_t} = T(\pi_{t-1}^{x_{t-1}, r_{t-1}}, y_t, u_t)$, where $T(\cdot)$ is defined in (9).

Using the dynamic programming recursion (14), the above cost function is straightforwardly minimized by

$$\begin{aligned} V_t(\pi^{x_t, r_t}) &= \alpha \left[\langle x^2, \pi_t^{x_t} \rangle - \langle x, \pi_t^{x_t} \rangle^2 \right] \\ &\quad + \beta \left(1 - \pi_t^{r_t'} \pi_t^{r_t} \right) \end{aligned} \quad (19)$$

$$\begin{aligned} u_t^* &= \arg \min_u \left[\langle \gamma'_t(\cdot, \cdot, u_t), \pi_{t-1}^{x_{t-1}, r_{t-1}} \rangle \right. \\ &\quad \left. + E_{y_t} \{ V_t(T(\pi_{t-1}^{x_{t-1}, r_{t-1}}, y_t, u)) \} \right]. \end{aligned} \quad (20)$$

Step 2: Set $\pi_t^{x_t, r_t} = T(\pi_{t-1}^{x_{t-1}, r_{t-1}}, y_t, u_t^*)$.

Step 3: $t \rightarrow t + 1$.

For a JMLS, the information state $\pi_t^{x_t, r_t}$ is a Gaussian mixture with s^t components. Therefore, in principle (although the computational complexity grows exponentially with t), the

above one-step ahead algorithm provides an algorithm for the sensor adaptive tracking problem. Below we discuss how an approximation of $\pi_t^{x_t, r_t}$ can be efficiently computed using the IMM algorithm—the resulting algorithm will be termed the Sensor Adaptive IMM Algorithm (SA-IMM). First, however, we briefly consider the linear Gaussian and HMM cases where π_t is easily computable.

Linear Quadratic Sensor Selection: Consider the same setup as in Section III-B. Given $P_{t-1|t-1}$, it follows from (20) that the one-step ahead algorithm at time t for selecting the optimal sensor is

$$u_t^* = \arg \min_{u \in \{1, \dots, L\}} [P_{t|t}(u) + \gamma_{t-1}(u)]$$

where $P_{t|t}(u_t)$ evolves according to (17). Unlike the optimal algorithm of Section III-B, the one-step ahead algorithm only requires L Kalman covariances to be evaluated at each time.

HMM Sensor Scheduling: Assume the same conditions as Section III-C but with β nonzero. The information state evolves according to the well known HMM filter [8] :

$$\pi^{r_t} = T(\pi_{t-1}^{r_{t-1}, y_{t-1}, u_{t-1}}) = \frac{\phi(y_t(u_t)) \mathcal{P}' \pi_{t-1}^{r_{t-1}}}{\mathbf{1}' \phi(y_t(u_t)) \mathcal{P}' \pi_{t-1}^{r_{t-1}}}$$

where \mathcal{P} denotes the transition probability matrix of r_t (see Section II-A),

$$\phi(y_t(u_t)) = \text{diag}[p(y_t|r_t = e_1, u_t), \dots, p(y_t|r_t = e_s, u_t)]$$

and $\mathbf{1}$ denotes the column vector of ones. Given $\pi_{t-1}^{r_{t-1}}$, the one-step ahead adaptive sensor solution for HMMs becomes

$$u_t^* = \arg \min_{u \in \{1, \dots, L\}} \gamma'_{t-1}(\cdot, u) \pi_{t-1}^{r_{t-1}} + \beta \sum_{m=1}^M \left[1 - \frac{\pi_{t-1}^{r_{t-1}} \mathcal{P} \phi^2(y_t(u_t)) \mathcal{P}' \pi_{t-1}^{r_{t-1}}}{(\mathbf{1}' \phi(y_t(u_t)) \mathcal{P}' \pi_{t-1}^{r_{t-1}})^2} \right].$$

B. Sensor Adaptive IMM (SA-IMM) Algorithm

The IMM algorithm computes the approximate information state for a JMLS in a computational efficient manner by approximating the information state π^{x_t, r_t} (s^t component Gaussian mixture) by an s -component Gaussian mixture at each time instant t . It is worthwhile mentioning that there are several other

suboptimal algorithms for approximating the information state of a JMLS—in particular, particle filter-based sequential sampling methods appear to be extremely promising; see [7] for details.

Given an approximate computation of the state $\pi^{x_{t-1}, r_{t-1}}$, the SA-IMM algorithm operates recursively via the steps below.

Step 1—IMM Algorithm: Compute an approximation of the updated information state $T(\pi^{x_{t-1}, r_{t-1}}, y_t, u_t)$ as follows. We have $\pi^{x_t, r_t} = \pi^{r_t} p(x_t|r_t, Y_t)$. Furthermore π^{r_t} and $p(x_t|r_t, Y_t)$ can be expressed recursively according to the following two equations:

i) By elementary application of Bayes' rule we have (21)–(24), shown at the bottom of the page.

ii) Modal update

$$\pi^{r_t}(j) = \sum_{i=1}^s p_{i,j} \pi^{r_{t-1}}(i) \times \int_{\mathbb{R}^{n_x}} p(y_t|x_t, r_t = e_j, u_t) p(x_t|r_t = e_j, Y_{t-1}) dx_t.$$

Consider the term $p(x_t|r_t = e_j, Y_{t-1})$ enclosed in square brackets on the last line of (24). The IMM algorithm approximates this by a single Gaussian density with the same mean and variance for each $j = 1, \dots, s$. Note that the densities $p(y_t|x_t, r_t, u_t)$, $p(x_t|x_{t-1}, r_t)$ are Gaussian as described in (11). Thus, with the IMM approximation, π^{x_t, r_t} is a Gaussian mixture with s components.

Step 2—Sensor Adaptive Step: Compute u_t^* according to (20).

Step 3—IMM State Update: Update the information state π^{x_t, r_t} using the new observation $y_t(u_t^*)$ according to the IMM algorithm. Each of the components $p(x_t|r_t = e_j, Y_t)$, $j = 1, \dots, s$ can be computed via a Kalman filter—hence, the SA-IMM algorithm involves a bank of s parallel Kalman filters; see [2] for details.

$$p(x_t|r_t, Y_t) = p(y_t|x_t, r_t, u_t) p(x_t|r_t, Y_{t-1}) \quad (21)$$

$$= p(y_t|x_t, r_t, u_t) \int_{\mathbb{R}^{n_x}} p(x_t|x_{t-1}, r_t) p(x_{t-1}|r_t, Y_{t-1}) dx_{t-1} \quad (22)$$

$$= p(y_t|x_t, r_t, u_t) \int_{\mathbb{R}^{n_x}} p(x_t|x_{t-1}, r_t) \quad (23)$$

$$\times \left[\frac{\sum_{i=1}^s p_{i,r_t} p(x_{t-1}|r_{t-1} = e_i, Y_{t-1}) p(r_{t-1} = e_i|Y_{t-1})}{\sum_{i=1}^s a_{i,r_t} p(r_{t-1} = i|Y_{t-1})} \right] dx_{t-1} \quad (24)$$

C. Sensor Adaptive Particle Filter Tracker

The main idea here is to replace the IMM state estimator above with a sequential Monte Carlo (particle filter) state estimator. The algorithm is similar to the Sensor Adaptive IMM algorithm presented above, except that the information state in Steps 1 and 3 is updated via a sequential Monte Carlo (particle filtering) algorithm. There are several classes of particle filtering algorithms that can be used, and [7] presents several algorithms for jump Markov linear systems together with variance reduction methods.

D. Sensor Adaptive IPDA (SA-IPDA)

When false measurements (clutter) are present the usual procedure is to identify which measurements are “close” enough to the predicted measurement for a particular target track then use this set of “validated” measurements in the track update procedure. The nearest-neighbor rule simply selects the validated measurement closest to the predicted measurement position whereas the PDA and IPDA algorithms use all of the validated measurements [2], [5], [22]. The track is updated with each measurement in the validated set and the updated estimates are combined in a weighted averaging process to form the new estimate. The weighting is proportional to the likelihood of each measurement scenario. The IPDA algorithm differs from the standard (Probabilistic Data Association (PDA) algorithm in that IPDA recursively computes the probability of target existence along with the track estimate. Thus, IPDA recursively computes the probability that the trail of detections currently being followed by the tracker originate from a real target rather than being a random collection of detections arising from clutter which happen to appear target like.

In the SA-IPDA algorithm we must decide on which sensor to use next, based only on the current state estimate and known statistical properties of each sensor and the clutter. Let ρ_t be the false measurements at time t and Ω_t the sequence of all real and false measurements and sensor choices up to and including time t . The SA-IPDA algorithm computes the expected posterior density if sensor l was used by first computing the posterior density if real measurement y_t and clutter ρ_t were obtained and then averaging it over y_t and ρ_t , with x_t , Ω_{t-1} and u_t fixed, to yield

$$\begin{aligned} \bar{p}(x_t|\Omega_{t-1}, u_t = l) &\triangleq E\{p(x_t|y_t, \rho_t, u_t = l, \Omega_{t-1})|x_t, \Omega_{t-1}, u_t = l\} \\ &= \int \int \frac{p(y_t, \rho_t|x_t, u_t = l)p(x_t|\Omega_{t-1})}{p(y_t, \rho_t|\Omega_{t-1}, u_t = l)} \\ &\quad \times p(y_t, \rho_t|x_t, u_t = l) dy_t d\rho_t \\ &= p(x_t|\Omega_{t-1}) \int \int \frac{p(y_t, \rho_t|x_t, u_t = l)^2}{p(y_t, \rho_t|\Omega_{t-1}, u_t = l)} dy_t d\rho_t. \end{aligned} \quad (25)$$

The sensor l^* , which minimizes a specific cost related to this expected density after normalization, is chosen, and a standard PDA or IPDA update follows.

Instead of minimizing a cost of the expected posterior density, another approach is to minimize the expected cost of the actual posterior density. For instance, we could compute the variance

of the posterior density given y_t , ρ_t , $u_t = l$, and Ω_{t-1} , average it over y_t and ρ_t to yield

$$E\{\text{var}\{x_t|y_t, \rho_t, \Omega_{t-1}, u_t = l\}|\Omega_{t-1}, u_t = l\}$$

and select the minimizing sensor. In the linear Gaussian case with uniformly distributed false measurements, this is equivalent to computing a modified matrix Riccati equation update for each sensor, similar to that used in [12].

E. Networked Sensors with One-Way Communication Links

Consider again the scenario in which communication between a number of sensors and a tracker occurs over finite bandwidth links, as discussed in Section III-A. In the sensor selection approach, the tracker uses its information state at each time instant to dynamically select the next batch of virtual sensors. If the feedback channel from the tracker to each sensor is absent, then the nature of the problem changes significantly since instead of passively implementing the selections of the tracker, each sensor must make its own bit allocation decisions, based on local information. Conversely, the task of the tracker is now simply to filter the data it receives from the sensors, without providing any feedback. Thus, this is no longer a centralized stochastic control problem.

In order to describe the essential characteristics of this problem, we focus here on the single link case without target mode jumping. Let the target and sensor be governed by (2) and (3), with the mode r_t and control variable u_t suppressed. Suppose that at each time t , only one symbol ψ_t from the finite set $\mathbb{Z}_M \triangleq \{0, 1, \dots, M-1\}$ can be transmitted from the sensor to the tracker, where $R \triangleq \log_2 M$ is the corresponding bit rate. Due to the finite rate, each symbol arrives at the tracker at time $t+1$. The coder and estimator are represented by the equations

$$\begin{aligned} \psi_t &= \gamma_t(Y_t, \Psi_{t-1}) \in \mathbb{Z}_M \\ \hat{x}_t &= \eta_t(\Psi_{t-1}) \in \mathbb{R}^{n_x}, \quad t = 1, 2, \dots \end{aligned}$$

where $\Psi_{t-1} \triangleq (\psi_1, \dots, \psi_{t-1})$. The aim is to find a *coder-estimator* $(\gamma, \eta) \triangleq (\{\gamma_t\}, \{\eta_t\})$, which minimizes the finite horizon cost

$$J \triangleq E \left\{ \sum_{t=1}^N \alpha_t \|x_t - \hat{x}_t\|^2 \right\}.$$

Note that since the information available to the tracker is a subset of that available to the coder, stochastic dynamic programming cannot be applied to minimize this. Nevertheless, several important structural properties may be stated.

First, it can be shown that the minimum cost may be decomposed as

$$\begin{aligned} \min_{\gamma, \eta} J &= E \left\{ \sum_{t=1}^N \alpha_t \|x_t - \hat{x}_{t|t-1}\|^2 \right\} \\ &\quad + \min_{\gamma, \eta} E \left\{ \sum_{t=1}^N \alpha_t \|\hat{x}_{t|t-1} - \hat{x}_t\|^2 \right\} \end{aligned} \quad (26)$$

where $\hat{x}_{t|t-1} \triangleq E\{x_t|Y_{t-1}\}$. As such, the problem of coding and estimating $\{x_t\}$ under a quadratic cost is exactly equivalent

to coding and estimating $\{\hat{x}_{t|t-1}\}$. In other words, as the coding step is where information is lost, the optimal coder first applies a Kalman filter to extract the best possible state estimates and then encodes them rather than the raw measurements. Note that the first term on the right-hand side of (26) is just the cost that would be obtained without a data rate constraint. As expected, it is a lower bound for the rate-constrained cost.

Second, for a given estimator η , the optimal coder is

$$\begin{aligned} \psi_t &= \arg \min_{\psi \in \mathbb{Z}_M} f_{t+1}(\hat{x}_{t+1|t}, (\Psi_{t-1}, \psi)) \\ \forall t &\in [1, 2, \dots, N-1] \end{aligned}$$

where the functions $f_t : \mathbb{R}^{n_x} \times \mathbb{Z}_M^t \rightarrow [0, \infty)$ are defined by the downward recursion

$$\begin{aligned} f_t(\hat{x}_{t|t-1}, \Psi_{t-1}) &\triangleq \alpha_t \|\hat{x}_{t|t-1} - \eta_t(\Psi_{t-1})\|^2 \\ &+ E \left\{ \min_{\psi \in \mathbb{Z}_M} f_{t+1}(\hat{x}_{t+1|t}, (\Psi_{t-1}, \psi)) \middle| \hat{x}_{t|t-1}, \Psi_{t-1} \right\} \quad (27) \end{aligned}$$

with $f_{N+1} \triangleq 0$ and $\min_{\gamma, \eta} J = \min_{\eta} f_1$. Hence, the symbol transmitted by an optimal coder at time t depends only on the current Kalman filter output $\hat{x}_{t+1|t}$ and the symbols previously transmitted. This is not surprising, since by virtue of the Gaussian assumption $\{\hat{x}_{t+1|t}\}$ is Markovian, and thus, $\hat{x}_{t+1|t}$ is a sufficient statistic for the future evolution of the filter process. Past filter outputs affect the current choice of symbol only through the symbols already transmitted. See [23] for more details.

In general, it is impossible to derive explicit formulae for optimal coder-estimators, even for the case of a one-dimensional target model. The Generalized Lloyd Algorithm [10] could be applied, or alternatively, f_1 could be approximated numerically for fixed horizon, data rate, and target and sensor models, but the entire procedure would have to be repeated if any of these changed. Nevertheless, the structural properties outlined above provide useful guidelines for the construction of suboptimal coder-estimators, as discussed in [23].

Finally, consider the following question: For given target and sensor models, what is the minimum data rate for which there exists a coder-estimator that produces bounded mean square estimation errors? In the special case of an autoregressive, linear time-invariant target model with no process or measurement noise, it has been shown that for there to exist a coder-estimator that takes the mean square error to zero it is necessary and sufficient that

$$R > \log_2 |\lambda|$$

where λ is the model pole with largest magnitude [24]. We conjecture that the same condition is necessary and sufficient for bounded mean square errors in the stochastic case.

F. Sensor Management Issues

The aim here is to show that the above algorithms can be easily modified to incorporate constraints on the total usage of particular sensors. Such constraints are often used in sensor resource management.

Consider a N horizon problem where sensor 1 can be used at most N_1 times, where $N_1 \leq N$. 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 (12) or in the suboptimal case the one-step ahead cost (6).

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 i th position. We will use process z_t to denote the number of times sensor 1 is used. Let $z_t = f_i$ if sensor 1 has been used $i - 1$ times up to time t . Then, the process $z_t \in S_1$ can be modeled as follows: If sensor 1 is used (i.e., $u_t = 1$) and $z_t = f_i$, then z_{t+1} jumps to state f_{i+1} . If any other sensor is used then $z_{t+1} = z_t = f_i$. Thus, z_t is a deterministic Markov chain with dynamics given by

$$z_t = Q(u_t)z_{t-1}, \quad z_0 = f_1 \quad (28)$$

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

$$Q(u_t = 1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

and $Q(u_t) = I_{(N_1+1) \times (N_1+1)}$ if $u_t \neq 1$.

The action space U_{t, z_t} is defined as follows:

$$U_{t, z_t} = \begin{cases} \{2, \dots, L\}, & \text{if } z_t = f_{N_1+1} \\ \{1, \dots, L\}, & \text{if } z_t \neq f_{N_1+1}. \end{cases}$$

The one-step ahead scheduling policy is given by $V_N(\pi_N, z_N) = \epsilon_N(\pi^{x_N, r_N})$, and for $t = N-1, N-2, \dots, 0$

$$\begin{aligned} V_t(\pi^{x_t, r_t}, z) &= \min_{u \in U_{k+1, z_{k+1}}} \left[\epsilon_t(u, \pi_t^{x_t, r_t}) \right. \\ &\left. + E_{y_{t+1}} \{V_{t+1}(T(\pi_t^{x_t, r_t}, y_{t+1}, u), Q'z)\} \right]. \quad (29) \end{aligned}$$

The above dynamic programming recursion can be recast into a form similar to that for a Jump Markov linear system by the following coordinate change: Consider the augmented Markov chain (r_t, z_t) . This has transition probability matrix $\bar{A} = \mathcal{P} \otimes Q$, where \mathcal{P} denotes the transition probability matrix of r_t , and \otimes denotes tensor (Kronecker product). Because z_t is a fully observed Markov chain, the information state of (X_t, z_t) is $\pi_t^{r_t} \otimes z_t$. This augmented information state is identical to that of a jump Markov linear system (with larger state space) and can be computed via the IMM algorithm in Section IV-B. Thus, the one-step ahead algorithm described above can be used for a practical suboptimal solution.

V. NUMERICAL EXAMPLES

In this section, we present a realistic simulation to illustrate some of the basic features of the sensor adaptive methods described above. The simulations presented here are confined to scheduled multisensor tracking of single and multiple targets.

Three two-dimensional (range-azimuth) pulse compression radar sensors with characteristics described below are employed to track single and multiple targets in a fusion system with limited data rate. These three radar sensors are arranged as shown in Fig. 3. Sensors 1 and 3 both have 1° antenna beam-width, whereas sensor 2 has an antenna beam-width of 0.2° . The azimuth measurement accuracy of these radar sensors is a function

of the antenna beam-width and the receive signal-to-noise ratio, which in turn is a function (via the radar equation) of many factors including the range of the target from the sensor, the radar transmit power, the receiver matched filter and integration gain, and the receiver front-end noise, etc. All three radar sensors have a compressed receiver pulse length of 50 ns and effective peak transmit power of 50 kW. The range measurement accuracy is a function of the receive signal-to-noise ratio (via the radar equation as for azimuth) and the compressed pulse length. Sensors 1 is located at (x, y) coordinates (120 km, 120 km), sensor 2 is located at (x, y) coordinates (100 km, 50 km), and sensor 3 is located at (x, y) coordinates (150 km, 100 km). The tracking computer update rate is 1 s in all cases, and for simplicity, it is assumed that all sensors are synchronized.

In the simulation examples below, we consider the state vector $x_t \in \mathbb{R}^4$ in (2) consisting of the x position, x velocity, y position, and y velocity of the target. In terms of the observation (3), for each sensor $l \in \{1, 2, 3\}$, the range and bearings measurements (polar coordinates) translate to cartesian coordinates (see [2]), where we have the equation at the bottom of the page, where $d_{k,l}$, $\theta_{k,l}$ denote, respectively, the range and bearing of the target with respect to sensor l .

Scenario 1—Single Maneuvering target and Multiple Sensors: The first scenario is depicted in Fig. 3. A target is in the field of view of three sensors. We simulated the target from time $t = 1, \dots, 60$ s. The target evolves according to the dynamics (2), with

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \sqrt{0.1} & 0 & 0 & 0 \\ 0 & \sqrt{0.1} & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad (30)$$

where $T = 1$ s. The target starts at (x, y) coordinates (100 km, 200 km) with (x, y) speed components 0 m/s and -2000 m/s. After 25 s, the target executes a sharp maneuver for 3 s, after which, its (x, y) speed components are 500 m/s and -1600 m/s; see Fig. 3. During these maneuver period, B in (2) is scaled by $\sqrt{10}$ to reflect additional uncertainty during the maneuver.

The three radar sensors communicate via a low data-rate network to a central tracking computer. The data-rate is such that data from only one sensor is able to be sent to the tracking computer during each update interval. The tracking computer knows the location of the sensors and their measurement statistics. The tracking computer must decide which sensor data it will request during the next update interval and the decision must be based only on the sensor data the tracking computer has seen up to (but not including) the current update interval.

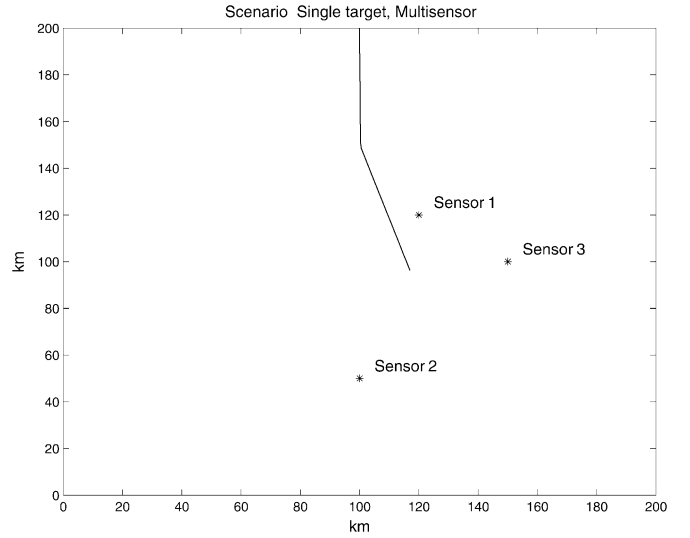


Fig. 3. Scenario 1: Single target and three sensors.

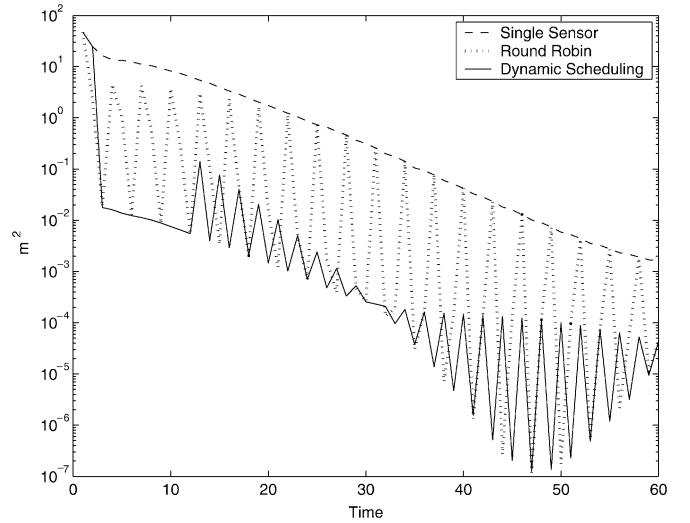


Fig. 4. Comparison of tracking error variance of three algorithms: Best sensor, Round robin scheduling, and SA-IMM scheduling.

The SA-IMM algorithm discussed in Section IV-B is used to select the next sensor and the data from the chosen sensor is communicated to the tracking computer and used to update the track estimate. The tracking error for a single realization is given in Fig. 4, which shows the variance of the x -coordinate of the tracking error for the “best” single sensor, a round-robin scheduling policy, and the adaptive scheduling policy given in this paper.

$$C(l) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D(l)D'(l) = \begin{bmatrix} d_{k,l}^2 \sigma_\theta^2 \sin^2 \theta_{k,l} + \sigma_d^2 \cos^2 \theta_{k,l} & (\sigma_d^2 - d_{k,l}^2 \sigma_\theta^2) \sin \theta_{k,l} \cos \theta_{k,l} \\ (\sigma_d^2 - d_{k,l}^2 \sigma_\theta^2) \sin \theta_{k,l} \cos \theta_{k,l} & d_{k,l}^2 \sigma_\theta^2 \cos^2 \theta_{k,l} + \sigma_d^2 \sin^2 \theta_{k,l} \end{bmatrix}$$

TABLE I
COMPARISON OF PERFORMANCE OF THREE SCHEDULING POLICIES FOR RANDOMLY LOCATED SENSOR 3. THE LOCATION OF THE SENSOR IS SPECIFIED BY (31)

Sensor 3 location parameter a (km)	Average Variance (m^2)		
	Single	Round Robin	Dynamic Sched
0	108×10^{-4}	48×10^{-4}	0.527×10^{-4}
1.5	123×10^{-4}	49×10^{-4}	0.532×10^{-4}
15	568×10^{-4}	113×10^{-4}	15×10^{-4}
150	12.47	1.02	0.45

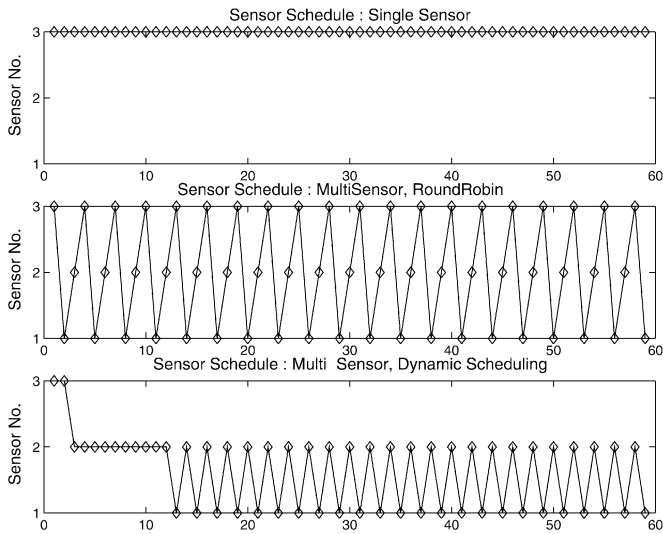


Fig. 5. Sensor choice of 3 algorithms: Best sensor, Round robin scheduling, and SA-IMM scheduling.

To further compare the performance of the three scheduling policies, we illustrate their performance as the location of Sensor 3 is randomly changed. We randomly assigned the x and y coordinates of the location of sensor 3 as

$$(x, y) = (150, 100) + (n_x, n_y) \quad (31)$$

where n_x and n_y are two independent uniformly distributed random variables with distribution $U[-a, a]$, and a is a parameter that is varied below. Table I gives average variance estimates of the x -coordinate of the tracking error for the three scheduling policies. For each value of a , the variance is averaged from time 45 to 60 and then averaged over 200 independent realizations.

The performance advantage gained from data adaptive scheduling is clear. The x coordinate of the tracking error in the SA-IMM case is very small and shows up as zero on the plot. The trace and/or determinant of the tracking error covariance matrix display the same characteristics with SA-IMM significantly outperforming round-robin and best single sensor. Fig. 5 shows the selected sensor sequence for these three sensor scheduling policies. The sensor sequence used by the SA-IMM will depend on the scenario and noise realizations and is guaranteed to always perform better than the best single sensor or round-robin policies.

Scenario 2—Multiple Targets and Sensors: A second, more complex, multitarget example is illustrated in Fig. 6, where the same three pulse compression radar sensors in the scenario above are now used to track three targets. Target 2 starts at (x, y) coordinates (100 km, 200 km) with (x, y) speed

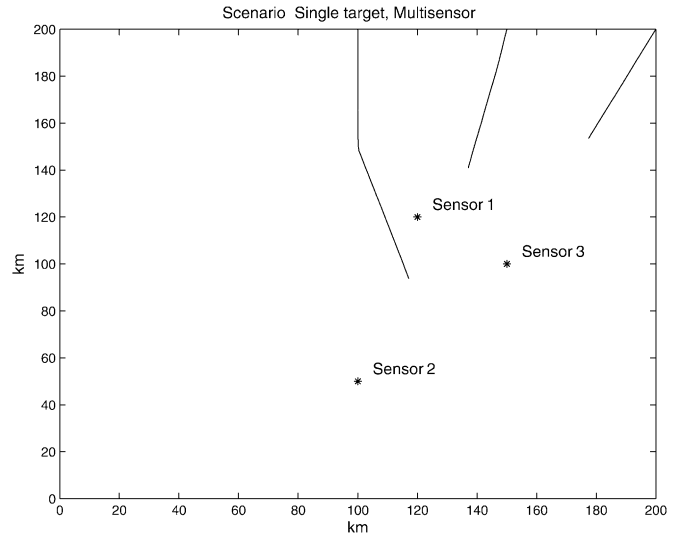


Fig. 6. Scenario 2: Three targets and three sensors.

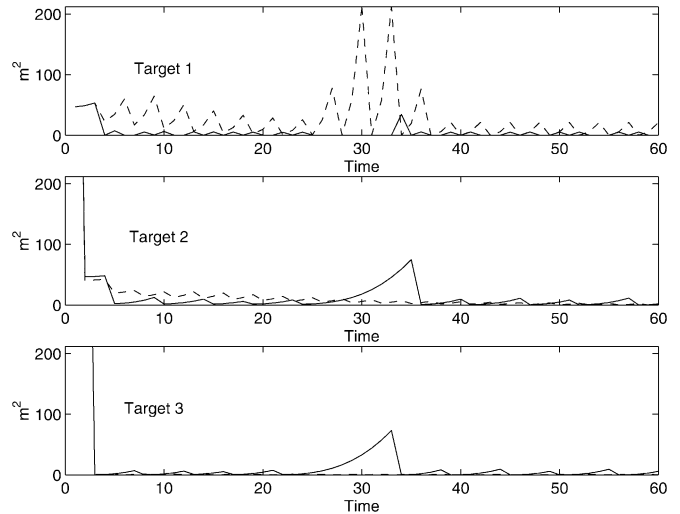


Fig. 7. Comparison of tracking error variance of Round Robin scheduling (dashed line) versus SA-IMM scheduling (solid line) for the three targets.

components (0 m/s, -2000 m/s). After 25 s, this target executes a sharp maneuver for 3 s, after which its (x, y) speed components are 500 m/s and -1600 m/s. Target 1 starts at (x, y) coordinates (200 km, 200 km) with (x, y) speed components (-400 and -800 m/s). Target 3 starts at (x, y) coordinates (150 km, 200 km) with (x, y) speed components (-200 and -1000 m/s). The scenario lasts 60 s.

Again, the low bandwidth requirement means that only measurement data from one sensor on one target is able to be communicated to the tracking computer during each tracking interval. The same three scheduling policies are compared in Fig. 7. The single best sensor policy performs poorly relative to the multitarget round robin and multitarget SA-IMM and is not plotted. multitarget round robin simply cycles through all sensors and all tracks. multitarget SA-IMM selects the most advantageous sensor and target. Fig. 7 shows the variance of the x -coordinate of the tracking error on all three tracks for these two policies. As with the single target case, the data adaptive sensor scheduling policy of SA-IMM significantly outperforms

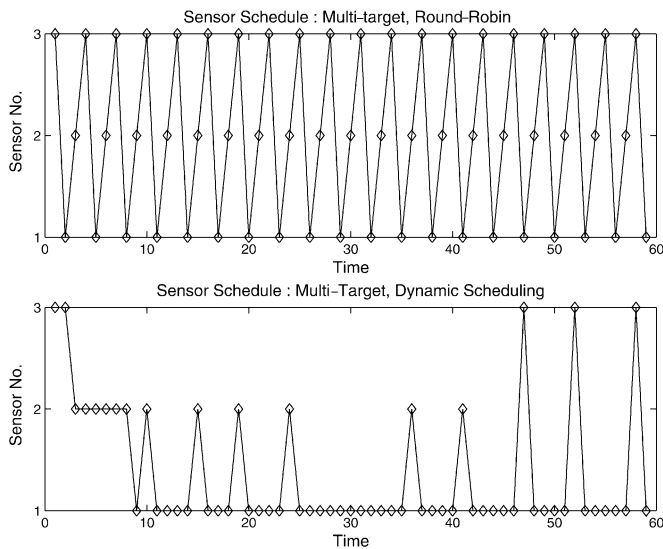


Fig. 8. Sensor choice of two algorithms: Round Robin and SA-IMM.

the multitarget round robin policy. The sensor sequence used is shown in Fig. 8. The selected track sequence is not plotted. In the multitarget round robin case the track sequence cycles in the same manner as the sensor sequence whereas in SA-IMM the chosen track and sensor are data dependent.

For further numerical examples on multiarmed bandit scheduling and HMM sensor scheduling, see [14] and [16].

VI. CONCLUSIONS AND EXTENSIONS

This paper has presented optimal and suboptimal algorithms for sensor and data rate control for tracking maneuvering targets. The maneuvering target was modeled as a jump Markov linear system. Determining the optimal control strategy was formulated as a stochastic dynamic programming problem. Special cases (linear quadratic and Hidden Markov Model sensor selection) where the dynamic programming recursion has a finite dimensional characterization were described. Three novel computationally feasible suboptimal algorithms were presented. In particular the sensor adaptive IMM and Sensor adaptive IPDA algorithms proposed in this paper are natural generalizations of the well known IMM and IPDA algorithms that incorporate sensor control. Finally, a brief outline of information theoretic aspects of the sensor and data rate control problem was presented.

As an extension of this paper, it is of interest to examine the role of simulation based reinforcement learning algorithms as opposed to stochastic dynamic programming in sensor and data rate control. Our recent papers [1], [18] show how stochastic gradient based reinforcement learning algorithms can be used for optimally controlling constrained Markov decision processes and for suboptimal control of partially observed Markov decision processes. Another possibility is to use a discrete policy search algorithm to determine the optimal sensor schedule—the resulting problem is a stochastic discrete optimization problem. In recent work [30], we have derived discrete stochastic approximation algorithms and analyzed their tracking properties.

REFERENCES

- [1] F. V. Abad and V. Krishnamurthy, (2003) Self Learning Control of Constrained Markov Decision Processes—A Valued Gradient Approach. GERAD-HEC, Montreal, QC, Canada. [Online]. Available: <http://www.gerad.ca/fichiers/cahiers/G-2003-51.pdf>
- [2] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Boston, MA: Artech House, 1993.
- [3] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. 1 and 2.
- [4] D. R. Billetter, *Multifunction Array Radar*: Artech House, 1989.
- [5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA: Artech House, 1999.
- [6] A. R. Cassandra, "Exact and Approximate Algorithms for Partially Observed Markov Decision Process," Ph.D. dissertation, Brown Univ., Providence, RI, 1998.
- [7] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 613–624, Mar. 2001.
- [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 R. J. Evans, "Image-enhanced multiple model tracking," *Automatica*, vol. 35, no. 11, pp. 1769–1786, 1999.
- [10] A. Gersho and R. M. Gray, *Vector Quantization and Data Compression*. Boston, MA: Kluwer, 1993.
- [11] O. Hernández-Lerma and J. B. Laserra, *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. New York: Springer-Verlag, 1996.
- [12] D. Kershaw and R. Evans, "Waveform selective probabilistic data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 4, pp. 1180–1188, Oct. 1997.
- [13] D. E. Kershaw and R. J. Evans, "Optimal waveform selection for tracking systems," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1536–1551, Sep. 1994.
- [14] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, Jun. 2002.
- [15] V. Krishnamurthy and R. J. Elliott, "Filters for estimating Markov modulated poisson processes and image based tracking," *Automatica*, vol. 33, no. 5, pp. 821–833, May 1997.
- [16] V. Krishnamurthy and R. J. Evans, "Hidden Markov model multiarmed bandits: A methodology for beam scheduling in multitarget tracking," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 2893–2908, Dec. 2001.
- [17] V. Krishnamurthy, F. Vazquez Abad, and K. Martin, "Adaptive nonlinear filters for narrowband interference suppression in spread spectrum CDMA systems," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 742–753, May 1999.
- [18] V. Krishnamurthy and F. V. A. K. Martin, "Implementation of gradient estimation to a constrained Markov decision problem," in *Proc. IEEE Conf. Decision Contr.*, Maui, HI, 2003.
- [19] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2139–2156, Aug. 1999.
- [20] W. S. Lovejoy, "A survey of algorithmic methods for partially observed Markov decision processes," *Ann. Operations Res.*, vol. 28, pp. 47–66, 1991.
- [21] L. Meier, J. Perschon, and R. M. Dressler, "Optimal control of measurement systems," *IEEE Trans. Autom. Control*, vol. AC-12, no. 10, pp. 528–536, Oct. 1967.
- [22] D. Musicki, R. Evans, and S. Stankovic, "Integrated probabilistic data association," *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1237–1241, Jun. 1994.
- [23] G. N. Nair and R. J. Evans, "A finite-dimensional coder-estimator for rate-constrained state estimation," in *Proc. 14th IFAC World Congr.*, Beijing, China, Jul. 1999, pp. 19–24.
- [24] —, "Stabilization with data-rate-limited feedback: Tightest attainable bounds," *Syst. Contr. Lett.*, vol. 41, no. 1, pp. 49–56, Aug. 2000.
- [25] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.
- [26] L. J. Sciacca and R. J. Evans, "Cooperative sensor networks with bandwidth constraints," in *Proc. SPIE AeroSense Conf.*, Orlando, FL, Apr. 2002.
- [27] R. D. Smallwood and E. J. Sondik, "Optimal control of partially observable Markov processes over a finite horizon," *Operations Res.*, vol. 21, pp. 1071–1088, 1973.
- [28] P. Whittle, "multiarmed bandits and the Gittins index," *J. R. Statist. Soc. B*, vol. 42, no. 2, pp. 143–149, 1980.

- [29] X. Xie and R. J. Evans, "Multiple-target tracking and multiple frequency line tracking using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 39, no. 12, pp. 2659–2676, Dec. 1991.
- [30] G. Yin, V. Krishnamurthy, and C. Ion, "Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization," *SIAM J. Optimiz.*, vol. 14, no. 4, pp. 1187–1215, 2004.



Robin Evans (F'04) was born in Melbourne, Australia, in 1947. After receiving the B.E. degree in electrical engineering at the University of Melbourne, Parkville, Australia, in 1969, he worked as a radar systems engineering officer with the Royal Australian Air Force. He received the Ph.D. degree in 1975 from the University of Newcastle, Callaghan, Australia.

He did postdoctoral studies at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, and the Control

and Management Department, Cambridge University, Cambridge, U.K. In 1977, he joined the University of Newcastle, where he was Head of the Department of Electrical and Computer Engineering from 1986 to 1991 and Co-Director of the Center on Industrial Control Systems from 1988 to 1991. In 1992, he moved to the University of Melbourne, where he was Head of the Department of Electrical and Electronic engineering until 1996. He is currently Research Leader for the Cooperative Center for Sensor Signal and Information Processing and Director of the Center for Networked Decision Systems. His research has ranged across many areas, including theory and applications in industrial control, radar systems, signal processing, and telecommunications.

Dr. Evans is a Fellow of the Australian Academy of Science and of the Australian Academy of Technological Sciences and Engineering.



Vikram Krishnamurthy (F'05) was born in 1966. He received the bachelor's degree from the University of Auckland, Auckland, New Zealand, in 1988, and Ph.D. degree from the Australian National University, Canberra, in 1992.

He is currently a Professor and Canada Research Chair with the Department of Electrical Engineering, University of British Columbia, Vancouver, BC, Canada. Prior to 2002, he was a chaired professor at the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville,

Australia, where he also served as Deputy head of department. His current research interests include ion channels and biological nanotubes, networked sensor scheduling and control, statistical signal processing, nonlinear filtering, and wireless telecommunications.

Dr. Krishnamurthy is currently an associate editor for the *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, the *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, and *Systems and Control Letters*. He was guest editor of the March 2005 special issue of the *IEEE TRANSACTIONS ON NANOBIOSCIENCE* on bio-nanotubes. He currently serves on the Signal Processing Theory and Methods (SPTM) Technical Committee of the IEEE Signal Processing Society and the International Federation of Automatic Control (IFAC) Technical Committee on Modeling, Identification, and Signal Processing. He has served on the technical program committee of several conferences in signal processing, telecommunications, and control.



Girish Nair was born in Petaling Jaya, Malaysia. He received the B.E. (Elec.) degree in 1994, the B.Sc. (mathematics) degree in 1995, and the Ph.D. (electrical engineering) degree in 2000, all from the University of Melbourne, Parkville, Australia.

He is currently a senior lecturer with the Department of Electrical and Electronic Engineering at the University of Melbourne, with research interests in the intersection of communications, information theory, and control.

Dr. Nair is the recipient of several prizes, including the Best Theory Paper award at the UKACC International Conference on Control, Cambridge, U.K., in 2000 and the L. R. East Medal in electrical engineering in 1994.

Len Sciaccia, photograph and biography not available at time of publication.