2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Optimal Sensor Scheduling for Hybrid Estimation

Weiyi Liu and Inseok Hwang

Abstract— **In this paper, we consider a sensor scheduling problem for a class of hybrid systems named as the Stochastic Linear Hybrid System (SLHS). We propose an algorithm which selects one (or a group of) sensor at each time from a set of sensors. Then, the hybrid estimation algorithm computes the estimates of the continuous state and the discrete state of the SLHS based on the observation from the selected sensors. As the sensor scheduling algorithm is designed such that a Bayesian decision risk is minimized, the true discrete state can be better identified. At the same time, the continuous state estimation performance of the proposed algorithm is better than that of other hybrid estimation algorithms using only predetermined sensors. Finally, our algorithm is validated though an illustrative target tracking example.**

I. INTRODUCTION

The problem of sensor scheduling involves utilizing multiple sensing agents to estimate the true state of a system. This problem arises in many applications [1] [2]. By properly switching between different sensors and merging/exchanging the information between them, sensor scheduling often gives more information about a system and yields better estimation accuracy but increases the complexity of the whole system. Generally speaking, sensor scheduling can be regarded as a optimal control problem which involves deriving an optimal control logic for sensor selection such that some cost (e.g. the estimation error) is minimized. The seminal work in this area can be found in [3] and [4]. Recently, R. M. Murray's group and B. Sinopoli's group presented interesting results (see [5] [6] and the references therein). Also, He et. al. have applied the Monte Carlo method to the sensor scheduling problem [7]. Other relative research can be found in [8], [9], and [10].

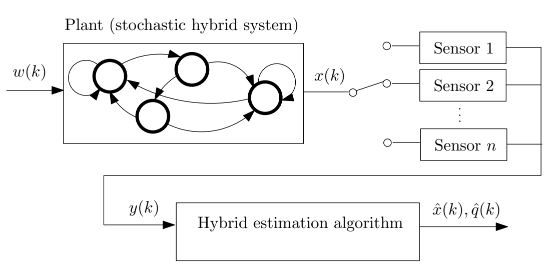

Fig. 1. Configuration of the stochastic hybrid system, the sensor network and the hybrid estimator.

We can model a system with switching sensors as a hybrid system, a class of system with interacting continuous dynamics and discrete dynamics. A switching between different sensors can be regarded as a transition between different discrete states of a hybrid system. In [11] and [12], the authors solved the sensor scheduling problem using the hybrid systems' approaches, but the nominal system itself is a Linear-Time-Invariant (LTI) system. In this paper, we consider the case in which the nominal system is a hybrid system with multiple switching modes, each observed by multiple sensing agents (See also Figure 1). Past research has shown that the state estimation itself is challenging for hybrid systems with synchronized sensors. If we do not know the discrete state transition history, the evolution of a hybrid system involves the exponentially increasing number of hypotheses over time [13]. This makes "optimal estimation" of the hybrid system computationally prohibitive.

In this paper, we derive the optimal sensor scheduling algorithm to estimate the discrete state and the continuous state of the SLHS. The proposed algorithm is divided into two parts: first, the optimal switching logic (switching history) of the sensors is determined through a sensor scheduling algorithm; second, the hybrid estimation is carried out based on the optimal sensor switching history. Due to the complexity of the whole system involving both switching system dynamics and switching observation model, the computation of the optimal sensor switching logic is intractable if it is designed such that the estimation error is minimized directly. To avoid this difficulty, we choose the "best" switching logic in the sense that the Bayesian decision risk in the estimation algorithm is minimized. The Bayesian decision risk is parameterized as the common area (or volume) under the different likelihood functions of the different hypotheses. Although there is no closed-form solution, its upper bound can be easily computed. In the case of two hypotheses testing, the area is bounded by the *Battacharyya Bound* [14]. The Bayesian decision risk bound of multiple hypotheses can be found in [15]. The proposed optimal sensor switching logic is designed such that the Bayesian decision risk bound is minimized at each estimation step. Numerical simulations show that the proposed sensor scheduling algorithm gives better estimation accuracy than the estimation algorithms just using predetermined sensor(s).

The rest of this paper is organized as follows: Section II introduces the SLHS model and the switching sensor model. In Section III, an illustrative example is presented to show the relation between the Bayesian decision risk computation and hybrid estimation. Also, we introduce the definition of the Battacharyya bound. In Section IV, the optimal sensor

W. Liu is a graduate student in the School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave., West Lafayette, IN, USA liu61@purdue.edu

I. Hwang is an Associate Professor in the School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Ave., West Lafayette, IN, USA ihwang@purdue.edu

switching logic and the corresponding hybrid estimation algorithm are derived. In Section V, the performance of the proposed algorithm is validated through an illustrative example. Conclusions are given in Section VI.

II. PROBLEM FORMULATION

In this section, we review the Stochastic Linear Hybrid System (SLHS) [16] and a switching sensor model is presented for hybrid estimation.

A. Stochastic Linear Hybrid System

We consider the following *Stochastic Linear Hybrid System (SLHS)* model. The discrete-time continuous dynamics of the SLHS is given by:

$$x(k) = A_{q(k)}x(k-1) + B_{q(k)}u(k) + F_{q(k)}w(k) \qquad (1)$$

where $x(k) \in X = \mathbb{R}^n$ is the state vector; $u \in \mathbb{R}^{m_u}$ is the known input vector; $q(k) \in \mathscr{Q} = \{1, 2, \ldots, n_d\}$ is the discrete state at time k; $\mathscr{Q}$ is a finite set of all the discrete states; A_q, B_q and F_q are the system matrices with appropriate dimensions, corresponding to each discrete state $q \in \mathscr{Q}$, and $w(k) \in \mathbb{R}^{m_w}$ is the white Gaussian process noise with $w(k) \sim \mathscr{N}(0, W(k))$ where W is a covariance matrix.

There are two types of discrete state transitions in the SLHS:

1) *Markov-jump transition model*: the discrete state transition history is a realization of a homogeneous Markov Chain. The finite state space of the Markov Chain is the discrete state space $\mathscr{Q}$. Suppose at each time k, the probability vector is given by $\pi(k) = \left[\pi_1(k) \cdots \pi_{n_d}(k)\right]^T$, with $\pi_i(k)$ denotes the probability of the event that the system's true discrete state at time k is i. Then, at the next time step $k+1$, the probability vector is:

$$\pi(k+1) = \Gamma\pi(k) \qquad (2)$$

where a constant matrix Γ is the Markov transition matrix with $\sum_j \Gamma_{ij} = 1$ (we use Γ_{ij} to denote the scalar component in the i-th row and j-th column of the Markov transition matrix Γ).

2) *State-dependent transition model*: the discrete state transition is governed by:

$$q(k+1) = \gamma(q(k), x(k), \theta) \qquad (3)$$

where $\theta \in \Theta = \mathbb{R}^l$ and $\gamma : \mathscr{Q} \times X \times \Theta \to \mathscr{Q}$ is the *discrete-state transition function* defined as:

$$\gamma(i, x, \theta) = j \quad \text{if } [x^T \ \ \theta^T]^T \in G(i, j)$$

We call $G(i, j)$ as a *guard condition*. For each combination of (i, j), the guard condition $G(i, j)$ is a subset of the space $\Omega = X \times \Theta$.

In this paper, we consider a specific kind of the guard condition $\{G(i, j) | i, j \in \mathscr{Q}\}$ named as the *stochastic linear guard condition*:

$$G(i,j) = \left\{ \begin{bmatrix} x \\ \theta \end{bmatrix} \middle| x \in X, \theta \in \Theta, L_{ij} \begin{bmatrix} x \\ \theta \end{bmatrix} + b_{ij} \leq 0 \right\} \qquad (4)$$

where $\theta \in \Theta = \mathbb{R}^l$ and $\theta \sim \mathscr{N}(\bar{\theta}, \Sigma_\theta)$ is a l-dimensional Gaussian random vector with mean $\bar{\theta}$ and covariance Σ_θ representing uncertainties in the guard condition; L_{ij} is a $\nu \times (n+l)$ matrix, b_{ij} is a ν-dimensional constant vector, and ν is the dimension of the vector inequality. Here, a vector inequality $y \leq 0$ means that each scalar element of y is non-positive.

B. Observation Model

In the observation model, several sensing agents are working cooperatively to provide the observation information. Suppose the cardinal number of the sensor set is M. We make the following assumption:

Assumption 1: At each time k, only one sensor is operating from a set of M sensors.

Assumption 1 is a common assumption in the research area of sensor scheduling. It can be extended to a more general case: if we allow several sensors working together at the same time, we can stack the observation vector from each sensor to get a bigger observation vector, and treat this vector as an observation coming from a new fictional sensor. Thus, Assumption 1 still holds.

The observation model of the i-th sensor is given by:

$$z_i(k) = C_i x(k) + v_i(k) \qquad (5)$$

where $i \in \mathscr{M}$ and $\mathscr{M} := \{1, \ldots, M\}$ is the set of M sensors; $z_i(k) \in \mathbb{R}^p$ is the measurement (output) of the i-th sensor; C_i are the observation matrices with appropriate dimension; and $v(k) \in \mathbb{R}^p$ is the white Gaussian observation noise with $v(k) \sim \mathscr{N}(0, R_i(k))$. We use $\mathscr{M}^N$ to denote the set of all ordered sequence of sensor schedules up to time N. Thus, an element $\sigma^N = \{\sigma_0^N, \sigma_1^N, \ldots, \sigma_{k-1}^N\} \in \mathscr{M}^N$ is a N-horizon sensor schedule. Under a given sensor schedule σ^N, the measurement sequence is given by:

$$z(k) = z_{\sigma^N}(k) = C_{\sigma_k^N} x(k) + v_{\sigma_k^N}(k), \forall k \in \{0, 1, \ldots, N-1\} \qquad (6)$$

III. BAYESIAN DECISION RISK FOR HYBRID ESTIMATION

Most hybrid estimation algorithms [13] involve a Bayes decision procedure of computing the likelihood of each discrete state based on the new measurement at each time step. Generally speaking, the purpose of sensor scheduling is to minimize the Bayesian decision risk in this procedure.

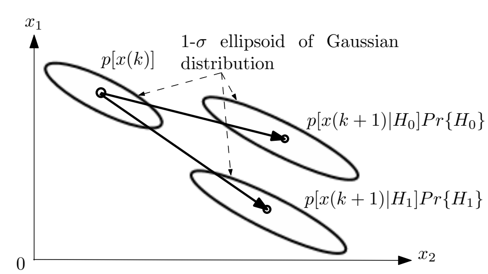

Fig. 2. Constinuous evolution of a simple stochastic linear hybrid system.

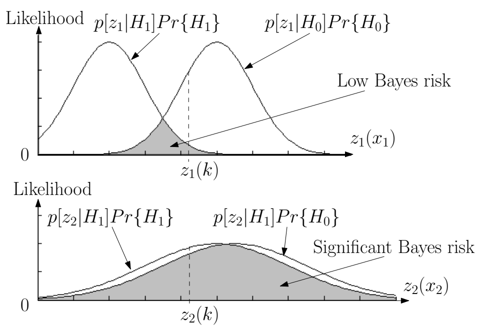

Fig. 3. Bayesian decision risk of the two sensors.

A. An Illustrative Example

We consider a stochastic hybrid system whose continuous state consists of two elements $x = [x_1 \ x_2]^T$ and its one-step evolution is shown in Figure 2. Suppose the system has two discrete states $\mathscr{Q} = \{1,2\}$ and at time k, the continuous state probability distribution function (pdf) is given by $p[x(k)]$ [1]. Without loss of generality, we assume the distributions are Gaussian and the system is a linear hybrid system. At time $k+1$, before the arrival of measurement z_i, we propagate the system under the two hypotheses:

$$H_0 : q(k+1) = 1 \text{ and } H_1 : q(k+1) = 2$$

The pdf of $p[x(k+1)|H_0]Pr\{H_0\}$ and $p[x(k+1)|H_1]Pr\{H_1\}$ are shown in Figure 2 using their $1-\sigma$ ellipsoid of the Gaussian distribution. Suppose we are to choose only one sensor to provide the measurement at each time. The sensor models are given by:

$$z_1(k) = [1 \quad 0]x(k) + v_1(k) = x_1(k) + v_1(k) \tag{7}$$

$$z_2(k) = [0 \quad 1]x(k) + v_2(k) = x_2(k) + v_2(k) \tag{8}$$

where v_1 and v_2 are the observation noise. From (7) and (8), we can compute the likelihoods of the two hypotheses H_0 and H_1 under the condition that Sensor 1 or 2 is currently in use, which is shown in Figure 3. In Figure 3, the upper figure shows the likelihood functions of the two hypotheses when Sensor 1 is activated, and the lower figure shows the likelihood functions when Sensor 2 is activated. In Figure 3, the difference between the likelihoods in the upper figure is much bigger than that in the lower figure, i.e., Sensor 1 can clearly identify the true discrete state under this scenario. The probability of making a wrong decision in this scenario can be regarded as the Bayesian decision risk. The shaded area is a measure of the Bayesian decision risk of the two hypotheses scenario. Obviously, the probability of the Bayes decision error is low if the shaded area is small.

B. Bound of the Bayesian Decision Risk

Given an observation sequence $\mathbb{Z}$ and a sensor schedule σ, the Bayes decision involves multiple decision regions $\mathscr{R}_i$ where H_i is the most likely hypothesis:

$$\mathscr{R}_i = \{\mathbb{Z} | p[\mathbb{Z}|H_i,\sigma] > p[\mathbb{Z}|H_j,\sigma] \quad \forall j \neq i\} \tag{9}$$

[1]Throughout this paper, we use $Pr\{\bullet\}$ to denote the probability of an event and $p[\bullet]$ for the probability distribution function (pdf).

Thus, the Bayesian decision risk can be computed by:

$$\begin{aligned} Pr\{error\} &= \sum_i \sum_{j \neq i} Pr\{\mathbb{Z} \in \mathscr{R}_j, H_i|\sigma\} \\ &= \sum_i \sum_{j \neq i} Pr\{\mathbb{Z} \in \mathscr{R}_j | H_i, \sigma\} Pr\{H_i\} \\ &= \sum_i \sum_{j \neq i} \int_{\mathscr{R}_j} p[\mathbb{Z}|H_i,\sigma] Pr\{H_i\} d\mathbb{Z} \end{aligned} \tag{10}$$

In (10), the probability $Pr\{error\}$ (Bayesian decision risk) can be regarded as the objective function to be minimized for the sensor scheduling algorithm. However, Equation (10) is computationally intractable, yet it is possible to bound the Bayesian decision risk in a closed form. If there are only two hypotheses H_0 and H_1, the *Battacharyya Bound* can be applied, which is given by:

$$Pr\{error\} \leq Pr\{H_0\}^{\frac{1}{2}} Pr\{H_1\}^{\frac{1}{2}} \int \sqrt{p[\mathbb{Z}|H_0] p[\mathbb{Z}|H_1]} d\mathbb{Z} \tag{11}$$

If the likelihood functions of the both hypotheses are Gaussian: $p[\mathbb{Z}|H_0] \sim \mathscr{N}(\mu_0, \Sigma_0)$ and $p[\mathbb{Z}|H_1] \sim \mathscr{N}(\mu_1, \Sigma_1)$, integral (11) can be evaluated analytically:

$$Pr\{error\} \leq Pr\{H_0\}^{\frac{1}{2}} Pr\{H_1\}^{\frac{1}{2}} \exp\{-k\} \tag{12}$$

where

$$k = \frac{1}{4}[\mu_1 - \mu_0]^T [\Sigma_0 + \Sigma_1]^{-1} [\mu_1 - \mu_0] + \frac{1}{2} \ln \frac{|(\Sigma_0 + \Sigma_1)/2|}{\sqrt{|\Sigma_0||\Sigma_1|}}$$

Note that in (12) the upper bound of the probability $Pr\{error\}$ can be regarded as a cost that the sensor scheduling algorithm is to minimize. If the Bayes decision involves multiple hypotheses, each being a Gaussian distribution: $p[\mathbb{Z}|H_i] = \mathscr{N}(\mu_i, \Sigma_i)$, the Bayes decision risk upper bound can be computed by [15]:

$$Pr\{error\} \leq \sum_i \sum_{j > i} Pr\{H_i\}^{\frac{1}{2}} Pr\{H_j\}^{\frac{1}{2}} \exp\{-k(i,j)\} \tag{13}$$

where

$$k(i,j) = \frac{1}{4}[\mu_j - \mu_i]^T [\Sigma_i + \Sigma_j]^{-1} [\mu_j - \mu_i] + \frac{1}{2} \ln \frac{|(\Sigma_i + \Sigma_j)/2|}{\sqrt{|\Sigma_i||\Sigma_j|}}$$

To summarize, we have derived a relationship between the upper bound of the Bayesian decision risk and the sensor selection problem for hybrid estimation.

IV. SENSOR SCHEDULING AND HYBRID ESTIMATION ALGORITHM

In this section, we present a sensor scheduling algorithm combined with a hybrid estimation algorithm. Our algorithm uses the "mixing" step which is similar to the IMM algorithm [17] to keep the exponentially growing computational complexity constant. First, we compute mode transition probabilities using the discrete state transition model. Then we compute the initial conditions for a bank of Kalman Filters (KF), each matched to a discrete state of the hybrid system. Based on the initial conditions, the sensor scheduling algorithm decides which sensor should be used at the next time step, so that the Bayesian decision risk is minimized. Finally, the estimate of the continuous state is given by a weighted sum of the output of each KF, and the discrete state estimate is given by the discrete state with the highest probability among all discrete states. Each step of our algorithm is described as follows (see also Figure 4):

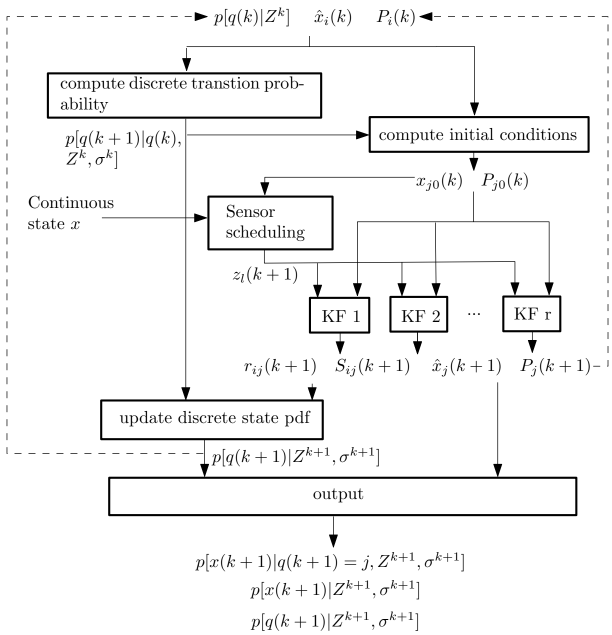

Fig. 4. The proposed sensor scheduling and hybrid estimation algorithm.

- **Step 1. Mixing (merging) probabilities:** The mixing probabilities $Pr\{q(k)=i|q(k+1)=j, Z^k, \sigma^k\}$ for all $i, j \in \mathscr{Q}$ are defined as:

$$\mu_{ji}(k) := Pr\{q(k)=i|q(k+1)=j, Z^k, \sigma^k\} \tag{14}$$

By the Bayes' Theorem,

$$Pr\{q(k)=i|q(k+1)=j, Z^k, \sigma^k\} = \frac{1}{c_j} Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\} p\left[q(k)|Z^k, \sigma^k\right] \tag{15}$$

where c_j is a normalizing constant. To evaluate (14) and (15), we use the following approach to compute the discrete state transition probability $Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\}$:

1) *Markov-jump transition*: the Markov transition matrix provides the *a priori* knowledge directly. The discrete state transition probability $Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\}$ in (15) can be written as:

$$Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\} = \Gamma_{ij} = const.$$

2) *State-dependent transition*: we recall that $\theta \in \Theta = \mathbb{R}^l \sim \mathscr{N}(\bar{\theta}, \Sigma_\theta)$ has a multivariate Gaussian distribution (if $\Theta \neq \emptyset$). With the linear guard condition given in (4), we compute the discrete state transition probability $Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\}$ in (15) as [16]:

$$Pr\{q(k+1)=j|q(k)=i, Z^k, \sigma^k\} = \Phi_\nu\left(L_{ij}\begin{bmatrix}\hat{x}_i(k)\\ \theta\end{bmatrix} + b_{ij}, L_{ij}\begin{bmatrix}P_i(k) & 0\\ 0 & \Sigma_\theta\end{bmatrix}L_{ij}^T\right)$$

where $\Phi_\nu(\bar{y}, \Sigma_y)$ is the ν-dimensional Gaussian cumulative density function (cdf) with mean $\bar{y}$ and covariance Σ_y:

$$\Phi_\nu(\bar{y}, \Sigma_y) := Pr\{y \leq 0\} = \int_{-\infty}^{0}\int_{-\infty}^{0}\cdots\int_{-\infty}^{0} \mathscr{N}_\nu(y; \bar{y}, \Sigma) dy_1 dy_2 \cdots dy_\nu$$

- **Step 2. Initial conditions for each KF:** At each time k, we approximate the initial condition of each KF by a single Gaussian distribution. The initial conditions (mean $\hat{x}_{j0}(k)$ and covariance $P_{j0}(k)$) for the j-th KF are given by:

$$\hat{x}_{j0}(k) = \sum_{i=1}^{n_d} \mu_{ji}(k)\hat{x}_i(k) \tag{16}$$

$$P_{j0}(k) = \sum_{i=1}^{n_d} \mu_{ji}(k)\{P_i(k) + [\hat{x}_i(k) - \hat{x}_{j0}(k)][\hat{x}_i(k) - \hat{x}_{j0}(k)]^T\} \tag{17}$$

- **Step 3. Sensor scheduling:** For mode j, compute the prior distribution (likelihood function) of the observation $z_i(k+1)$ (recall that the subscript i means the i-th sensor is used for observation):

$$p[z_i(k+1)|q(k+1)=j, Z^k, \sigma^k] = \mathscr{N}(r_{ij}(k+1), S_{ij}(k+1))$$

where

$$r_{ij}(k+1) = C_i A_j \hat{x}_{j0}(k) + C_i B_j u(k+1)$$
$$S_{ij}(k+1) = C_i A_j P_{j0}(k) A_j^T C_i^T + C_i Q(k) C_i^T + R_i(k)$$

Based on the likelihood function $p[z_i(k+1)|q(k+1)=j, Z^k, \sigma^k]$, compute the Battacharyya Bound of the Bayesian decision risk of the i-th sensor using (12) or (13). The sensor schedule σ^k is augmented to σ^{k+1} such that the upper bound in (12) or (13) is minimized at time $k+1$.
- **Step 4. Mode-matched filtering:** Suppose at time $k+1$, the l-th sensor in the sensor set $\mathscr{M}$ is chosen in Step 3. After the arrival of the new measurement $z_l(k)$, each KF computes the *posterior* mean and covariance $\hat{x}_j(k+1)$, $P_j(k+1)$ conditioned on $q(k)=j$ for $\forall j \in \mathscr{Q}$.
- **Step 5. Discrete-state probability update:** For each KF, the likelihood function is

$$\Lambda_j(k+1) := p[z(k+1)|q(k+1)=j, Z^k, \sigma^{k+1}] = \mathscr{N}_p(r_{ij}(k+1); 0, S_{ij}(k+1)) \tag{18}$$

By Bayes' Theorem, the discrete state probability $\alpha_j(k+1|k+1) := Pr\{q(k+1)=j|Z^{k+1}, \sigma^{k+1}\}$ is given by

$$\alpha_j(k+1|k+1) = \frac{1}{\delta_j} Pr\{z_l(k+1)|q(k+1)=j, Z^k, \sigma^{k+1}\} Pr\{q(k+1)=j|Z^k, \sigma^{k+1}\} \tag{19}$$

where δ_j is a normalizing constant. Substituting (18) into (19) and using the total probability theorem on the term $Pr\{q(k+1)=j|Z^k\sigma^{k+1}\}$ in (19), we get

$$\alpha_j(k+1|k+1) = \frac{1}{\delta_j}\Lambda_j(k+1) \times \sum_{i=1}^{n_d}[Pr\{q(k+1)=j|q(k)=i;Z^k,\sigma^{k+1}\} \times Pr\{q(k)=i|Z^k,\sigma^{k+1}\}] \quad (20)$$

- **Step 6. Output:** By the total probability theorem, the continuous state pdf at time $k+1$ is given by

$$p[x(k+1)|Z^{k+1},\sigma^{k+1}] = \sum_{j=1}^{n_d}\{p[x(k+1)|q(k+1)=j,Z^{k+1},\sigma^{k+1}] \times p[q(k+1)=j|Z^{k+1},\sigma^{k+1}]\} \quad (21)$$

We approximate the sum of the r terms in (21) via moment matching by a single Gaussian pdf [18]:

$$p[x(k+1)|Z^{k+1},\sigma^{k+1}] \approx \mathcal{N}_n(x;\hat{x}(k+1),P(k+1))$$

where

$$\hat{x}(k+1) = \sum_{j=1}^{n_d}\alpha_j(k+1|k+1)\hat{x}_j(k+1|k+1)$$

$$P(k+1) = \sum_{j=1}^{n_d}\left\{P_j(k+1|k+1) + \alpha_j(k+1|k+1)[\hat{x}_j(k+1|k+1)-\hat{x}(k)][\hat{x}_j(k+1|k+1)-\hat{x}(k)]^T\right\}$$

The discrete state probability at time $k+1$ is given by

$$Pr\{q(k+1)=j|Z^{k+1},\Sigma^{k+1}\} = \alpha_j(k+1|k+1)$$

and its estimate is:

$$\hat{q}(k+1) = \arg\max_j Pr\left\{q(k+1)=j|Z^{k+1},\sigma^{k+1}\right\}$$

V. SIMULATIONS

In this section, we validate the performance of our algorithm through a target tracking scenario.

A. Target Dynamics

For the purpose of illustration, we consider the dynamics of a target which has three discrete states (modes): Left Turn (LT), Right Turn (RT) and Constant Velocity (CV). In the LT mode and the RT mode, the target performs a coordinated turn with a constant turning rate while the target keeps its velocity constant in the CV mode. We assume that the transition between different modes to be governed by a time-homogeneous Markov Chain whose evolution is given by (2). Let $\pi(k) = [\pi_1(k)\ \pi_2(k)\ \pi_3(k)]^T$, where $\pi_1(k)$, $\pi_2(k)$ and $\pi_3(k)$ are the probabilities that the true discrete state is LT, RT or CV, respectively. The Markov transition matrix is parameterized as:

$$\Gamma = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \quad (22)$$

The continuous state of the target is represented by the state vector: $x = [\xi\ \dot{\xi}\ \eta\ \dot{\eta}]^T \in X = \mathbb{R}^4$ in the $\xi - \eta$ frame. The continuous dynamics is governed by stochastic difference equations, each corresponding to one discrete state. The dynamics corresponding to the CV mode is given by [16]:

$$\begin{bmatrix} \xi(k+1) \\ \dot{\xi}(k+1) \\ \eta(k+1) \\ \dot{\eta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi(k) \\ \dot{\xi}(k) \\ \eta(k) \\ \dot{\eta}(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix} \begin{bmatrix} w_{\xi CV}(k) \\ w_{\eta CV}(k) \end{bmatrix} \quad (23)$$

where $w_{\xi CV}$ and $w_{\eta CV}$ are independent white Gaussian noise with $E[w_{\xi CV}^2(k)] = 1^2$, $E[w_{\eta CV}^2(k)] = 1^2$ and $E[w_{\xi CV}(k)w_{\eta CV}(k)] = 0$; $T_s = 1sec$ is the sampling time. The continuous dynamics corresponding to the LT and the RT modes is given by [16]:

$$\begin{bmatrix} \xi(k+1) \\ \dot{\xi}(k+1) \\ \eta(k+1) \\ \dot{\eta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \frac{sin(\omega T_s)}{\omega} & 0 & -\frac{1-cos(\omega T_s)}{\omega} \\ 0 & cos(\omega T_s) & 0 & -sin(\omega T_s) \\ 0 & \frac{1-cos(\omega T_s)}{\omega} & 1 & \frac{sin(\omega T_s)}{\omega} \\ 0 & sin(\omega T_s) & 0 & cos(\omega T_s) \end{bmatrix} \begin{bmatrix} \xi(k) \\ \dot{\xi}(k) \\ \eta(k) \\ \dot{\eta}(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix} \begin{bmatrix} w_{\xi L/RT}(k) \\ w_{\eta L/RT}(k) \end{bmatrix} \quad (24)$$

where $\omega = 10deg/sec$ for the LT mode and $\omega = -10deg/sec$ for the RT mode; $w_{\xi L/RT}$ and $w_{\eta L/RT}$ are independent white Gaussian noise with $E[w_{\xi L/RT}^2(k)] = 1^2$, $E[w_{\eta L/RT}^2(k)] = 1^2$ and $E[w_{\xi L/RT}(k)\ \ w_{\eta L/RT}(k)] = 0$.

We assume that there are two sensing agents providing the observation information. At each time k, only one sensor is turned on. The observation model for Sensor 1 is given by:

$$z_1(k) = C_1x(k) + v_1(k) = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0.1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} v_{11}(k) \\ v_{12}(k) \end{bmatrix} \quad (25)$$

and the observation model for Sensor 2 is given by:

$$z_2(k) = C_2x(k) + v_2(k) = \begin{bmatrix} 0.1 & 0 & 0 & 1 \\ 0 & 0.1 & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} v_{21}(k) \\ v_{22}(k) \end{bmatrix} \quad (26)$$

In (25) and (26) v_{11}, v_{12}, v_{21} and v_{22} are mutually independent white Gaussian noise with $E[v_{11}^2(k)] = E[v_{12}^2(k)] = E[v_{21}^2(k)] = E[v_{22}^2(k)] = 1^2$

We simulate the target motion for 110 seconds. Figure 5 shows the actual target trajectory and compares it with the position estimation results computed by different observation setups: two sensors working cooperatively, using Sensor 1 only, and using Sensor 2 only. From Figure 5, we can see that the proposed hybrid estimation algorithm combined with the sensor scheduling algorithm gives the best position estimates. If a predetermined sensor is applied, the Bayesian decision

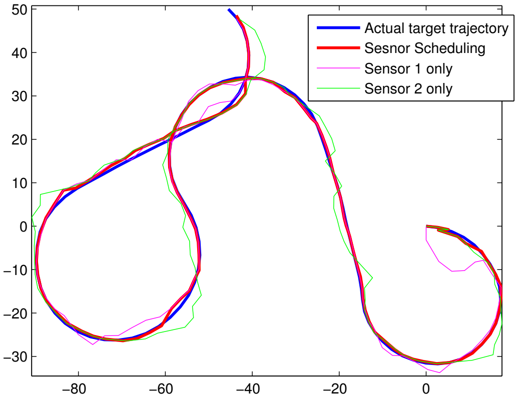

Fig. 5. Target trajectory and position estimation results.

risk for the sensor is high, which leads to big position estimation errors. However, if the two sensors working cooperatively, the Bayesian decision risk is always reduced to a low level.

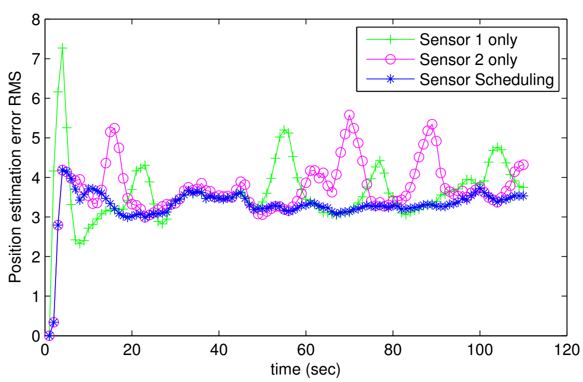

Fig. 6. Result of Monte Carlo simulation: Root-Mean-Square(RMS) error of position estimates.

Figure 6 compares the RMS position estimation errors obtained via a 100 run Monte Carlo simulation. In Figure 6, the RMS errors of the sensor scheduling algorithm stays below the error of the hybrid estimation algorithms that utilize one predetermined sensor. The simulation shows that the sensor scheduling algorithm selects the "best" sensor such that the estimation error is minimized at each step. Table I compares the discrete state estimation errors and summarizes the continuous state estimation errors given by the three algorithms. From the simulation results, we can see that the hybrid estimation algorithm with scheduled multiple sensors is the best among the three algorithms.

VI. CONCLUSIONS

In this paper, we have considered the problem of optimal sensor scheduling for hybrid state estimation. We have proposed the Bayesian decision risk for the hybrid estimation with multiple sensors. The optimal sensor scheduling is designed such that the Bayesian decision risk (or its upper bound) is minimized. The performance of the proposed algorithm has been validated through a target tracking scenario.

TABLE I

PERFORMANCE COMPARISON OF PERFORMANCE: STATISTICS OF 100 SIMULATION RUNS

Algorithm		Sensor scheduling	Sensor 1 only	Sensor 2 only
RMS position error (m)	Peak	4.3354	7.5244	5.8382
	Overall average	3.5233	4.4327	4.2359
Average no. of discrete state estimation error		17	35	33

VII. ACKNOWLEDGEMENT

The authors would like to acknowledge that this work is supported by NSF CAREER Award CNS-0746299 and to thank Dr. Helen Gill for her support.

REFERENCES

[1] V. Krishnamurthy and D. V. Djonin. Optimal threshold policies for multivariate POMDPs in radar resource management. *IEEE transactions on Signal Processing*, 57(10):3954–3969, October 2009.

[2] B. Doshi B. Henrick L. Benmohamed, P. Chimento and I. Wang. Sensor network design for underwater surveillance. *MILCOM*, 0:1–7, 2006.

[3] J. Peschon L. Meier III and R. Dressler. Optimal control of measurement subsystems. *IEEE transactions on Automatic Control*, 12:528–536, 1967.

[4] K. Herring. and J. Melsa. Optimum measurements for estimation. *IEEE Transactions on Automatic Control*, 19(3):264 – 266, 1974.

[5] B. Sinopli L. Shi, M. Epstein and R. M. Murray. Effective sensor scheduling schemes in a sensor network by employing feedback in the communication loop. In *Proceedings of the 2007 IEEE Multi-conference on Systems and Control*, Singapore, October 2007.

[6] B. Hassibi V. Gupta, T. H. Chung and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42:251–260, 2006.

[7] Y. He and E.K.P. Chong. Sensor scheduling for target tracking: A monte carlo sampling approach. *Digital Signal Processing*, 16(5):533 – 545, 2006.

[8] Y. Oshman. Optimal sensor selection strategy for discrete-time state estimator. *IEEE Transactions on Aerospace and Electronic Systems*.

[9] E.K.P. Chong Y. Li, L.W. Krakow and K.N. Groom. Approximate stochastic dynamic programming for sensor scheduling to track multiple targets. *Digital Signal Processing*, 19:978–989, 2009.

[10] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[11] A. Abate J. Hu M. P. Vitus, W. Zhang and C. Tomlin. On efficient sensor scheduling for linear dynamical systems. In *Proc. American Control Conference*, Baltimore, MD, USA, June 2010.

[12] A. Bemporad D. Bernardini, D. Munoz de la Pena and E. Frazzoli. Simultaneous optimal control and discrete stochastic sensor selection. In *Hybrid System Computation and Control (HSCC) Conference*, pages 61–75. Springer-Verlag, 2009.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[14] P. Hart R. Duda and D. Stork. *Pattern Classification*. Wiley Interscience, New York, 2000.

[15] S. Rajamanoharan L. Blackmore and B.C. Williams. Active estimation for jump markov systems. *IEEE Transactions on Automatic Control*, 53(10):2223–2236, 2008.

[16] C. E. Seah and I. Hwang. Stochastic linear hybrid systems: Modeling, estimation, and application in air traffic control. *IEEE transactions on Control Systems Technology*, 17(3):563–575, 2009.

[17] X. R. Li and Y. Bar-Shalom. Design of an interacting multiple model algorithm for air traffic control tracking. *IEEE transactions on Control Systems Technology*, 1(3):186–194, 1993.

[18] Y. Bar-Shalom and T. F. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.