Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

A Sampling-based Approach to Nonparametric Dynamic System Identification and Estimation

Songhwai Oh, Jin Kim, Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
{sho,jin,sastry}@eecs.berkeley.edu

Abstract— **We propose a new probabilistic framework for nonparametric identification and estimation of dynamic systems. Under the parametric paradigm, a model of the system and a set of observations are given and the parameter space of the model is searched to optimize an objective function. However, if we are uncertain about the model, the parametric approach can easily overfit data and lead to risky decisions. In nonparametric estimation, the model uncertainty is introduced in a systematic manner to find both the model and associated parameters of the system. In this paper, we consider a dynamic system consisting of a varying number of subsystems with noisy observations. The objective is to identify the subsystems at each time step and estimate the associated parameters such that the observations are explained the best. We develop an efficient algorithm based on Markov chain Monte Carlo methods and apply our approach to multiple target tracking problems. We address the issues with the subsystem initiation and termination and initial state estimation. In simulation our algorithm shows excellent performance for tracking a varying number of maneuvering targets with nonlinear dynamics. In some cases our algorithm outperforms any linear filtering algorithm with perfect associations.**

I. INTRODUCTION

In parametric estimation problems, we are given a set of (noisy) observations from a known model and the goal is to estimate the parameters of the model such that some objective function is maximized (or minimized). For example, in linear regression, we minimize the sum of squared errors. In many practical problems the method of maximum likelihood provides a good solution for parametric estimation problems. However, if there is uncertainty about the model, the parametric estimation methods, such as the maximum likelihood estimation, can easily overfit data and lead to risky decisions. The recognition of the limitations of parametric methods led to the surge of nonparametric methods such as an information criterion [1], Bayesian information criterion [21], minimum description length [10] and Bayes factors [13]. But difficulties in formulating the prior models and testing for different models make them difficult to be applied to complex problems.

Solving complex problems by sampling methods such as Markov chain Monte Carlo (MCMC) has become more tractable due to increased computational power. MCMC-based algorithms now play a significant role in many fields such as physics, statistics, economics, and engineering [4]. In some cases MCMC is the only known general algorithm which finds a good solution in polynomial time [12]. For interested readers we refer to [4], [2], [8]. However, due to measure theoretic issues, standard MCMC is limited to problems with a fixed dimension, i.e., parametric estimation problems, and cannot be readily extended to problems with varying dimensional parameters, i.e., nonparametric estimation problems. Reversible jump MCMC (RJMCMC) solves this issue by introducing reversible jumps between dimensions using deterministic dimension-matching transformations [9]. This method differs from other nonparametric methods, in that it allows the comparison among different models simultaneously while estimating parameters associated with each model. The different models are considered simultaneously based on the evidence at hand so that the models not supported by the evidence are not considered as frequently as the models supported by the evidence. In addition, the method provides numerical values representing confidence in different models and bypasses the model choice structure of the other nonparametric methods, which require prior modeling and testing for the different models [19].

In this paper, we develop a probabilistic framework for nonparametric identification and estimation of dynamic systems. We consider a dynamic system with noisy observations, which consists of a varying number of subsystems. Our goal is to identify the subsystems at each time step and estimate the associated parameters, including the dynamics of each subsystem, such that the observations are explained the best. Each subsystem can be either present or absent at each time step following a Markov transition probability independently from the other subsystems. In addition the initial state of a subsystem is unknown when it appears or reappears. The main difficulty lies in the identification of subsystem initiation and termination times. We take a Bayesian hierarchical modeling approach [18] to formulate a general framework. Under our framework, observations are partitioned into subsystems to which they belong, and thus, it becomes a trivial task to use the existing efficient parametric methods to estimate the parameters of subsystem dynamics.

The multiple target tracking problem is a good example of the dynamic system we have described. Under the most general setup, a varying number of targets are moving around in a region with continuous motions,

0-7803-8335-4/04/$17.00 ©2004 AACC

and the positions of these moving targets are sampled at random intervals. The position measurements are noisy, with detection probability less than one, and there is a noise background of spurious position reports (false alarms). Targets appear at random in space and time. Each target persists independently for a random length of time and then ceases to exist. A track of a target is defined as a path in space-time traveled by the target. The seminal paper by Sittler [22] introduced the major concepts about multiple target tracking and a method to evaluate tracks. He pointed out two major problems in multiple target tracking: data association and state estimation. However, in the worst case, one has to search over all possible tracks, which is equivalent to searching over the collection of all partitions of observations. This combinatorial optimization problem is NP-hard since it is a multidimensional assignment problem [16]. Subsequently, more computationally efficient algorithms have been proposed using heuristics to reduce the size of the search space [23] [17]. The multiple hypothesis tracker (MHT) introduced in [17] uses heuristics such as pruning, gating and N-scan-back logic but at the expense of accuracy.

As opposed to finding the optimal association, there is an alternative suboptimal approach for data association, called the joint-probabilistic data-association filter (JPDAF) [3]. The main limitations of JPDAF are that it assumes a fixed number of targets at all times and it cannot initiate or terminate tracks. On the other hand, probabilistic multi-hypothesis tracking (PMHT) uses probabilistic associations between observations and targets to avoid the maintenance of a hypothesis tree [24]. But PMHT also assumes a fixed number of targets and does not allow track initiation and termination. Our framework can be considered as a full statistical extension of PMHT for tracking an unknown number of targets. With the development of particle filters, multitarget tracking algorithms have been extended to nonlinear dynamics. In [11], PMHT is applied with particle filters, but this method requires suboptimal estimation of the states of targets. JPDAF is used in [20] with particle filters, but the described limitations of JPDAF remain the same. See [6] for more detail information about multitarget tracking algorithms.

This paper provides a probabilistic framework for layered dynamic systems and an efficient RJMCMC based algorithm. We are able to construct an efficient sampler by taking the advantages of the structure of the problem and proposing targeted local moves. We apply this method to multitarget tracking problems, and demonstrate the robustness of the algorithm against outliers. In some cases, when we apply our algorithm to nonlinear dynamics, it outperforms any linear filtering algorithm with perfect associations.

The remainder of this paper is structured as follows. Section II presents a probabilistic framework for nonparametric identification and estimation of dynamic systems, called the dynamic Bayesian model selection. Section III describes an RJMCMC algorithm for solving the nonparametric identification and estimation problems. We also describe an online version of the algorithm. In Section IV, the algorithm is applied to track a varying number of targets moving with nonlinear dynamics, and the performance of the algorithm is analyzed.

II. DYNAMIC BAYESIAN MODEL SELECTION

Let us consider a discrete-time dynamic system $\mathcal{S}$ composed of subsystems $\mathcal{S}_i$ for $i = 1, \ldots, K$ where K is the total number of subsystems. Each subsystem can be either present or absent at any given time. When a subsystem is present over consecutive times, its state evolves according to unique dynamics independently from other subsystems. Let $\mathbb{R}^n$ be the state space of each subsystem. Let $X_t^i \in \mathcal{R} \subset \mathbb{R}^n$ be the state of $\mathcal{S}_i$ and $U_t^i \in \mathbb{R}^p$ be a control input to $\mathcal{S}_i$ for $t = 1, \ldots, T$. $\mathcal{R}$ has a finite volume $V_{\mathcal{R}}$. The dynamic of $\mathcal{S}_i$ is $F^i : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$. Let $F = \{F^i : 1 \le i \le K\}$. The state transition is noisy such that

$$X_{t+1}^i = F^i(X_t^i, U_t^i) + V_t^i,$$

where $V_t^i \in \mathbb{R}^n$ is a noise process. When there is no confusion we will denote the subsystem $\mathcal{S}_i$ as the subsystem i. Let $M_t^i \in \{0, 1\}$ be a Markov chain denoting the status of the subsystem i, such that $M_t^i = 1$ if the subsystem i is present; otherwise $M_t^i = 0$. Let $p^i(j, k)$ for $j, k \in \{0, 1\}$ be the transition probability of M_t^i. Using the independence assumption, we can combine M_t^i into a single Markov chain $M_t \in \{0, 1\}^K$ on a product state space with the transition probability

$$P(M_{t+1} = m_{t+1} | M_t = m_t) = \prod_{i=1}^K p^i(m_t^i, m_{t+1}^i).$$

Let A be the transition matrix for M_t and the i-th row of A be distributed from the Dirichlet distribution $\mathcal{D}(\alpha_i^1, \ldots, \alpha_i^{2^K})$, where $\alpha \in [0, 1]^{2^K \times 2^K}$. Since we do not assume that the transition matrix A is known in advance, a prior model on A allows us to estimate A based on observations.

Let X_t be the state of $\mathcal{S}$ at time t such that $X_t = ({X_t^{i_1}}^T, \ldots, {X_t^{i_{k_t}}}^T)^T \in \mathbb{R}^{n \times k_t}$, where $k_t = \#\{i : M_t^i = 1, 1 \le i \le K\}$ and $M_t^{i_r} = 1$ for $r = 1, \ldots, k_t$. U_t is defined similarly. Let f^i be the conditional density of the next state of the subsystem i given its current state and control. Then the conditional probability of X_{t+1} is given by

$$\begin{aligned} &P(X_{t+1} \in dx_{t+1} | X_t = x_t, U_t = u_t, M_t = m_t, M_{t+1} = m_{t+1}) \\ &= \prod_{i=1}^K \begin{cases} f^i(dx_{t+1}^i | x_t^i, u_t^i) & \text{if } m_{t+1}^i = 1, m_t^i = 1 \\ p_0^i(dx_{t+1}^i) & \text{if } m_{t+1}^i = 1, m_t^i = 0 \\ 1 & \text{otherwise,} \end{cases} \end{aligned}$$

where $p_0^i(x)$ is a priori density of the initial state of the subsystem i.

Let $Y_t^j \in \mathbb{R}^m$ be an observation vector for $j = 1, \ldots, n_t$. The observation model is

$$Y_t^j = H(X_t^i) + V_t'^j,$$

where $V_t'^j \in \mathbb{R}^m$ is a noise process. Let $h(y|x)$ be the density of the observation y given the state x. Now we assume that the observations $Y_t = \{Y_t^1, \ldots, Y_t^{n_t}\}$ are distributed according to the mixture model such that

$$Y_t^j \sim \sum_{r=0}^{k_t} w_t^{i_r} h(\cdot|x_t^{i_r}) \qquad \text{for } j = 1, 2, \ldots, n_t,$$

where $w_t^{i_r}$ is the weight of the subsystem i_r and $\sum_{r=0}^{k_t} w_t^{i_r} = 1$. The weight w_t^i can be interpreted as the chance that a randomly chosen observation is generated from the subsystem i. Let W_t be random variables associated with the weights w_t. We assume W_t are distributed from the Dirichlet distribution $\mathcal{D}(\delta_t^0, \ldots, \delta_t^{k_t})$ and independent over time, i.e., W_t is independent of W_s for $s \neq t$.

As in [18], we define allocation variables Z_t such that $Z_t^j = i_r$ if j-th observation comes from the subsystem i_r and Z_t^j are drawn from the distributions $P(Z_t^j = i_r) = w_t^{i_r}$ for $r = 0, 1, \ldots, k_t$. We associate all observations that are not generated from the known subsystems with clutter and let w_t^0 be its weight. The observations from clutter are uniformly distributed over $\mathcal{R}^m \subset \mathcal{R}$. Let $V_{\mathcal{R}^m}$ be the volume of $\mathcal{R}^m$. We consider these observations as outliers. If the j-th observation is generated from clutter, we set $Z_t^j = 0$. Hence given the allocation variable Z_t, the observations are distributed as

$$P(Y_t^j \in dy | Z_t = z, X_t = x) = \begin{cases} h(dy|x^{z^j}), & z^j \neq 0 \\ dy/V_{\mathcal{R}^m}, & z^j = 0, \end{cases}$$

independently for $j = 1, \ldots, n_t$. We note that this mixture model formulation is similar to the probabilistic associations of PMHT [24].

We assume V_t^i is a white Gaussian process with zero mean and a covariance matrix Σ^i. Σ^i have the scaled inverse χ^2 distribution, i.e., $\Sigma_{(l,l)}^i \sim$ Inv-$\chi^2(\nu_{il}, {\sigma_{0_{il}}}^2)$ for $l = 1, \ldots, n$ [7]. The hyperparameter ν_{il} is called the degree of freedom and $\sigma_{0_{il}}$ is the scale parameter. We also assume $V_t'^j$ is a white Gaussian process with zero mean and a covariance matrix Σ^y, and the hyperparameters are assigned similarly.

Now to make an inference on the described model, we need to estimate the dimensionality k_t and a set of parameters associated with k_t for every t. However, since at each t, explicit associations between parameters at $t-1$, t and $t+1$ are required for the state transitions of subsystems, we consider each instance of the subsystem status as a model. So there are 2^K possible models at each t. The model labeling is explicit in our formulation and we work with the union space $\mathcal{C} = \bigcup_{k \in \mathcal{K}} \mathcal{C}_k$, where $\mathcal{C}_k$ is a parameter subspace associated with the k-th model and $\mathcal{K} = \{1, \ldots, 2^K\}$. In the following sections we show that this highly complex model selection problem is efficiently solved by reversible jump MCMC. Lastly, the joint distribution of all the variables (except those for noise models) can be expressed as

$$\begin{aligned} &P(M_{1:T}, X_{1:T}, U_{1:T}, Y_{1:T}, Z_{1:T}, W_{1:T}, \delta_{1:T}, A, \alpha, F) \\ &= P(F)P(\alpha)P(A|\alpha)P(X_1, M_1)P(U_{1:T}) \\ &\times \textstyle\prod_{t=1}^{T-1} P(M_{t+1}, X_{t+1}|M_t, X_t, U_t, F, A) \\ &\times \textstyle\prod_{t=1}^{T} P(Y_t|Z_t, X_t)P(Z_t|W_t, M_t)P(W_t|\delta_t, M_t)P(\delta_t), \end{aligned}$$

where $x_{1:T} = \{x_1, \ldots, x_T\}$. Hence our main objective is to estimate $P(M_{1:T}|Y_{1:T})$, $P(X_{1:T}|M_{1:T}, Y_{1:T})$, $P(U_{1:T}|M_{1:T}, Y_{1:T})$, $P(A|Y_{1:T})$, $F|Y_{1:T}$ and so on.

III. ALGORITHM

A. MCMC and Reversible Jump MCMC

Many practical problems are involved with high-dimensional, high-complexity probability distributions. In order to make an inference or prediction, one must integrate over these complex distributions, but there is rarely a closed-form analytical expression for the high-dimensional integrals. Markov chain Monte Carlo (MCMC) is a family of stochastic algorithms that uses Markov chains to estimate the integrals which have no closed-form analytical expressions.

In MCMC, an irreducible and aperiodic Markov chain is constructed such that its stationary distribution is the posterior $\pi(\theta)$ where θ is the set of parameters of interests. Then given the current state θ, the sampler proposes a candidate state θ' from a proposal distribution $q(\theta, \theta')$. Then the proposal is accepted with probability

$$\min\left\{1, \tfrac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}\right\}$$

so that the detailed balance of the Markov chain is preserved. Then, by the ergodic theorem, one can estimate $E_{\pi(\theta)}[f(\theta)]$ of a bounded function $f(\cdot)$ by

$$\bar{f} = \tfrac{1}{n-m} \textstyle\sum_{i=m+1}^{n} f(\theta_i)$$

where θ_i is the state of the Markov chain at the i-th iteration [8]. The first m samples are burn-ins and discarded for the estimate calculations. For more detail, see [4], [2], [8].

The standard MCMC is limited to problems with fixed dimension and cannot be easily extended to problems with varying dimension parameters due to the measure theoretic issues. The reversible jump MCMC (RJMCMC) resolves this issue by introducing reversible jumps between dimensions using deterministic dimension-matching transformations [9]. In RJMCMC, different types of moves are considered to traverse across the combined parameter space $\mathcal{C}$. If the MCMC sampler proposes a move type m and a new state θ' from θ, the move is accepted with probability

$$\min\left\{1, \tfrac{\pi(\theta')j_m(\theta')q_{m2}(u')}{\pi(\theta)j_m(\theta)q_{m1}(u)} \cdot \left|\tfrac{\partial g_m(\theta,u)}{\partial(\theta,u)}\right|\right\},$$

where $j_m(\theta)$ is the probability of choosing the move type m when in state θ; $q_{m1}(u)$ and $q_{m2}(u')$ are the proposal densities of u and u', respectively; and g_m is the deterministic dimension-matching bijection such that $g_m(\theta, u) = (\theta', u')$ and $g_m^{-1}(\theta', u') = (\theta, u)$. The Jacobian arises from the change of variables from (θ, u) to (θ', u') [18].

B. DBMS RJMCMC

We now present an RJMCMC algorithm for solving DBMS problems. Let θ be the set of the unknowns of which we are interested in finding the posterior $\pi(\theta)$ given $y_{0:T}$.

In construction of the algorithm we assume that $\delta = \delta_t^0 = \cdots = \delta_t^{k_t}$ for all t and both δ and α are held fixed.

Each MCMC step consists of the following moves: (a) track split, track merge or track update move on randomly selected t, (b) weight update move for W_t, (c) allocation update move for Z_t, (d) transition matrix update move for A, (e) covariance matrices update move, and (f) dynamics update move. The track split and merge moves of (a) are dimension-varying moves while the other moves are within the same dimension. The moves (b) and (c) are the Gibbs moves on W_t and Z_t for t selected in (a). In moves (a)-(c), we propose a new state θ' which differs from θ only at some time t. Since only a small neighborhood of t is affected by this move, the acceptance probability is high and computed efficiently. Since it is required to consider parameters at all times for the moves (d)-(f), we choose the moves (d)-(f) with some small probability p_{all} while the moves (a)-(c) are chosen with probability (w.p.) $1 - p_{all}$. The moves (b)-(f) are routine, and we refer the readers to [18], [7].

Now we describe the dimension varying moves of (a). We choose a track split move w.p. b_{n_0} and a track merge move w.p. d_{n_1} such that $b_{n_0} + d_{n_1} < 1$ where $n_1 = \sum k_t$ and $n_0 = KT - n_1$. A track update move is chosen w.p. $1 - b_{n_0} - d_{n_1}$. For the convenience of our discussion below, we define submove types. We say the move at (i,t) has the submove type: (i) if $M_{t-1}^i = M_{t+1}^i = 0$, (ii) if $M_{t-1}^i = 0$ and $M_{t+1}^i = 1$, (iii) if $M_{t-1}^i = 1$ and $M_{t+1}^i = 0$, and (iv) if $M_{t-1}^i = M_{t+1}^i = 1$. The track update move is similar to the track split move described below, except the dimension stays the same. We choose time t and a subsystem i and update its state at time t. For the sake of space, we omit the description of the track update move. For details about the algorithm, see [15].

1) Track split and merge moves: Let $G_1 = \{(i,t) : M_t^i = 1, 1 \le i \le K, 1 \le t \le T\}$. For a split move, we select (i,t) w.p. $p_b(\theta,(i,t))$, where $p_b(\theta,(i,t)) > 0$ on G_1^C and $p_b(\theta,(i,t)) = 0$ on G_1, and propose a new subsystem at (i,t) by splitting the weight of clutter W_t^0 into $W_t'^0$ and $W_t'^i$. The move is rejected if $W_t^0 = 0$ or $|n_t^0| = 0$ where $n_t^0 = \{j : Z_t^j = 0\}$ since such move is not reversible. For a merge move, we select (i,t) w.p. $p_d(\theta,(i,t))$, where $p_d(\theta,(i,t)) > 0$ on G_1 and $p_d(\theta,(i,t)) = 0$ on G_1^C, and merge the subsystem at (i,t) into clutter.

In a track split move, we take the following steps to propose θ':

1) Propose η_1 from Beta distribution, $\eta_1 \sim Be(2,2)$;
2) Set $W_t'^0 = W_t^0\eta_1$ and $W_t'^i = W_t^0(1-\eta_1)$;
3) Set $M_t'^i = 1$ and $M_t'^r = M_t^r$ for $r \neq i$;
4) Propose η_2 for $X_t'^i$. The proposal of η_2 is different for each submove type and it is described below;
5) Propose the allocation variables $Z_t'^j \in \{0, i\}$ for $j \in n_t^0$ using the posterior of Z_t given the other variables. The remaining variables are unchanged.

There are three different split (merge) moves: placement split (merge), forward split (merge), and backward split (merge). For submove (i), we choose the placement split; for submove (iii) and (iv), the forward split; and for submove (ii), the backward split. In general, the acceptance probability is $\min(1, R)$ with an acceptance ratio $R = \frac{\pi(\theta')}{\pi(\theta)} R_m R_{p1} R_{p2} R_j$ where $R_m = \frac{d_{n_1+1} p_d(\theta',(i,t))}{b_{n_0} p_b(\theta,(i,t))}$, $R_{p1} = 1/(q(\eta_1)q(Z_t'))$, $R_{p2} = 1/q(\eta_2)$, and $R_j = \left|\frac{\partial g}{\partial(\cdot)}\right|$. $q(x)$ denotes the proposal density of x. A similar merge move is applied for each submove type and a merge move is accepted w.p. $\min(1, R^{-1})$ with appropriate substitutions. We also define

$$R_t = \frac{P(y_t|Z_t',X_t')A_{M_{t-1}M_t'}A_{M_t'M_{t+1}}P(Z_t',W_t'|M_t',\delta)}{P(y_t|Z_t,X_t)A_{M_{t-1}M_t}A_{M_tM_{t+1}}P(Z_t,W_t|M_t,\delta)}$$
$$\text{where} \quad \frac{P(Z_t',W_t'|M_t',\delta)}{P(Z_t,W_t|M_t,\delta)} = \frac{(W_t'^0)^{\delta-1+l_0}(W_t'^i)^{\delta-1+l_i}}{(W_t^0)^{\delta-1+l_0+l_i}B((k_t+1)\delta,\delta)}$$

and $l_0 = \#\{j : Z_t'^j = 0, j \in n_t^0\}$ and $l_i = \#\{j : Z_t'^j = i, j \in n_t^0\}$ and $B(\cdot)$ is the Beta function.

Placement Split: Propose a new state $X_t'^i \sim \frac{1}{n_t}\sum_{j=1}^{n_t}\mathcal{N}\left(\cdot|y_t^j, \Sigma^y\right)$. The acceptance ratio becomes $R = R_t \frac{p_0^i(X_t'^i)}{1} R_m R_{p1} \frac{1}{q(X_t'^i)} W_t^0$.

Forward Split: For the forward split, a new state $X_t'^i$ is proposed from $F^i(X_{t-1}^i, U_{t-1}^i) + \eta_2$ where $\eta_2 \sim \mathcal{N}(\cdot|0, \Sigma^i)$. For the submove type (iii), $R = R_t R_m R_{p1} W_t^0$ and for the submove type (iv), $R = R_t \frac{f^i(X_{t+1}^i|X_t'^i)}{p_0^i(X_{t+1}^i)} R_m R_{p1} W_t^0$.

Backward Split: This move is available when $(F_X^i)^{-1}$ exists and is differentiable where F_X^i is a coordinate function of F^i for the state variable X. We sample $\eta_2 \sim \mathcal{N}(\cdot|0,\Sigma^i)$ and set $X_t'^i = (F_X^i)^{-1}(X_{t+1}^i - \eta_2)$. The acceptance ratio becomes $R = R_t R_m R_{p1} \left|\frac{\partial (F_X^i)^{-1}(X_{t+1}^i - \eta_2)}{\partial \eta_2}\right| W_t^0$.

IV. SIMULATION RESULTS

We use a unicycle dynamic model for our simulations. The state vector is $x = (x, y, \theta)^T$ where (x, y) is a position of a vehicle in a plane and θ is a heading of a vehicle. The continuous state equation is $\dot{x} = (\varphi\cos\theta, \varphi\sin\theta, \omega)^T$ where φ is a directional velocity and ω is an angular velocity. φ and ω are control inputs. If φ and ω are constant over the sampling period T_s, its discretized state equation (with noise) is

$$\begin{aligned} x_{t+1} &= x_t + \tfrac{2}{\omega_t}\varphi_t \sin(\tfrac{\omega_t T_s}{2})\cos(\theta_t + \tfrac{\omega_t T_s}{2}) + v_{t,1} \\ y_{t+1} &= y_t + \tfrac{2}{\omega_t}\varphi_t \sin(\tfrac{\omega_t T_s}{2})\cos(\theta_t + \tfrac{\omega_t T_s}{2}) + v_{t,2} \\ \theta_{t+1} &= \theta_t + \omega_t T_s + v_{t,3}, \end{aligned}$$

where $V_t = (v_{t,1}, v_{t,2}, v_{t,3})^T$ are white Gaussian noises.

Now suppose there is more than one target moving with the dynamics described above and the type of target can be classified by its directional velocity. We have $X_{t+1}^i = F^i(X_t^i = (x_t^i, y_t^i, \theta_t^i), \varphi^i, \omega_t^i) + V_t^i$ and each φ^i is constant. Let $\varphi = (\varphi^1, \ldots, \varphi^K)$. But we let ω_t^i be a random variable uniformly distributed over $[-\pi/4, \pi/4]$. So the targets are free to change their heading at any time as they desire and the tracking algorithm is required to distinguish them. We assume that the initial state of a target is uniformly

distributed over $\mathcal{R} = [0, L]^2 \times [-\pi, \pi]$. Hence $p_0^i(x) = \frac{1}{2\pi L^2}$ for each i. We use $b_{n_0} = \frac{n_0}{KT}$ and $d_{n_1} = \frac{.5n_1}{KT}$ as the probabilities for split and merge moves. $p_b(\theta, (i,t))$ is assigned such that (i,t) positioned before or after an established track is weighted three times higher than the remaining $(i,t) \in G_1^C$.

We consider a linear observation model

$$Y_t^j = HX_t^i + V_t'^j \quad \text{where} \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

if j-th observation comes from the target i. The covariance matrices for the noise processes are $\Sigma^i = \text{diag}((.05T_s\varphi^i)^2, (.05T_s\varphi^i)^2, (.2\frac{\pi}{8})^2)$ for each i and $\Sigma^y = \text{diag}((.2T_s)^2, (.2T_s)^2)$. The transition matrix A is held fixed such that $A_{jk} \propto 10^{K-d_1(j,k)}$ where d_1 is the Manhattan distance between two states j and k in their binary representations. So the number of targets varies according to A. The false alarms are uniformly distributed over the region $\mathcal{R}^2 = [0, L]^2$ and the number of false alarms has a Poisson distribution with mean $\lambda V_{\mathcal{R}^2}$. The Dirichlet prior on weights is $\delta = 1$. The detection probability is .9.

Assessing the convergence of MCMC algorithms is usually a difficult task. It is especially challenging in our case since the dimension of the parameter space changes [5]. We can measure the distance between the true tracks and the estimated tracks since we are working in a simulation environment. Let $M_{1:T}^*$ and $X_{1:T}^*$ be the parameters of true tracks. Let $\hat{M}_{1:T}$ be the estimate with maximum posterior and let $\hat{X}_{1:T}$ be the estimate of states given $\hat{M}_{1:T}$. We use two metrics:

$$d_m(M_{1:T}^*, \hat{M}_{1:T}) = \frac{1}{KT} \sum_{t=1}^{T} \sum_{i=1}^{K} \mathbb{I}(M_t^{*i} \neq \hat{M}_t^i)$$

to measure the distance between models and

$$\begin{aligned} &d_x(X_{1:T}^*, M_{1:T}^*, \hat{X}_{1:T}, \hat{M}_{1:T}) \\ &= 1 - \frac{1}{KT}\Big(\sum_{t=1}^{T} \sum_{i=1}^{K} \mathbb{I}(M_t^{*i} = \hat{M}_t^i = 0) \\ &+ \mathbb{I}(M_t^{*i} = \hat{M}_t^i = 1) e^{-\frac{1}{2}(X_t^{*i} - \hat{X}_t^i)^T (3^2\Sigma^i)^{-1} (X_t^{*i} - \hat{X}_t^i)} \Big) \end{aligned}$$

to measure the distance between the models and states simultaneously. Here $\mathbb{I}(\cdot)$ is an indicator function. We used the first two components of X_t^i to evaluate d_x in experiments below.

A. Experiment I - Convergence

We set $K = 3$, $T = 20$, $L = 100$, $T_s = 1$, $\lambda V_{\mathcal{R}^2} = 1$ and $\varphi = (4, 6, 8)$. We generated ten random scenarios then ran each scenario ten times (the first 20,000 samples are used as burn-ins). All observations are assigned to clutter at the initial state of the sampler. Figure 1 shows d_m and d_x averaged over 100 runs against the number of MCMC samples. Any tracks with length less than three time steps are discarded from the estimates. We note that the metric d_x is very conservative and visual inspection shows that any distance less than .1 is almost a perfect match.

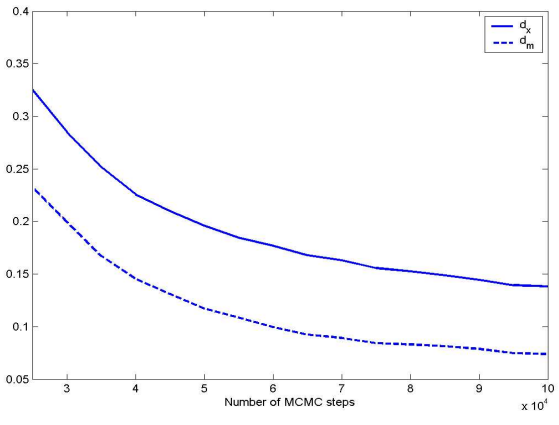

Fig. 1. Experiment I: d_m, d_x vs. number of MCMC samples

B. Experiment II - DBMS vs. Optimal Linear Filter

We have compared the performance of our algorithm against the optimal linear filter using Kalman filters on the same scenarios used in Experiment I. The multiple tracking algorithm such as MHT uses Kalman filters, hence, if the observations are correctly partitioned, i.e., when the data association is correct, MHT will give the optimal estimates according to the linear dynamics. For each scenario, we first partitioned the observations into the original tracks from which the scenario was generated. We then run Kalman filters on each track and compared d_x estimated from Kalman filters against the estimates from DBMS. For the Kalman filters, we use the usual linear model for tracking [14]. The state vector is $x = [x, y, \dot{x}, \dot{y}]^T$ where (x, y) is a position on a plane and $(\dot{x}, \dot{y})$ is a velocity vector. Let δ be the sampling interval. Then the dynamic and measurement models are

$$\begin{aligned} x_{t+1} &= Ax_t + Gw_t \\ y_t &= Cx_t + v_t, \end{aligned}$$

where

$$A = \begin{bmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G = \begin{bmatrix} \frac{\delta^2}{2} & 0 \\ 0 & \frac{\delta^2}{2} \\ \delta & 0 \\ 0 & \delta \end{bmatrix}, \quad \text{and}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Here w_t and v_t are Gaussian noise processes with zero mean and covariance Q and R, respectively. We used $Q = \alpha m_x \times \text{diag}(1, 1)$ and $R = \alpha m_y \times \text{diag}(1, 1)$, where $m_x = \max_{i,j,k}(\Sigma_{j,k}^i)$ and $m_y = \max_{j,k}(\Sigma_{j,k}^y)$. We vary α to check if the estimates are influenced by the changes in covariance.

Table I summarizes these results. Note that the observations are not partitioned for DBMS and the estimates for DBMS are averaged over ten repeated runs. The estimates from the optimal linear filter are not influenced by the

TABLE I

DBMS VS. OPTIMAL LINEAR FILTER

Test Case	DBMS d_x	Optimal Linear Filter d_x ($\alpha = 1$)	d_x ($\alpha = 5$)	d_x ($\alpha = 10$)
1	0.1205	0.1200	0.1233	0.1243
2	**0.1319**	0.1637	0.1660	0.1666
3	**0.1605**	0.2253	0.2277	0.2281
4	0.1206	0.1051	0.1034	0.1032
5	**0.1480**	0.2606	0.2610	0.2612
6	**0.1223**	0.1648	0.1675	0.1680
7	**0.2085**	0.2260	0.2283	0.2289
8	**0.1688**	0.2000	0.2018	0.2022
9	0.1371	0.1225	0.1238	0.1241
10	0.2007	0.1397	0.1437	0.1445

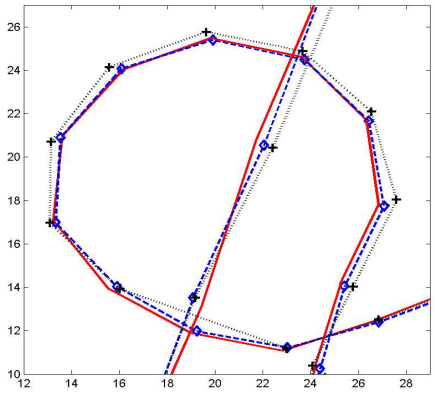

Fig. 2. An example of the inaccuracy of the optimal linear filter (true trajectory - straight line; optimal linear filter estimates - dotted line with crosses; DBMS estimates - dashed line with diamonds)

changes in covariance. In some cases, DBMS outperforms the optimal linear filter (shown in boldface). This means, in these cases, DBMS outperforms any linear filtering algorithm even if the linear filtering algorithm is given perfect information about data association. Since we can directly apply the nonlinear dynamics with DBMS, it does not suffer from the approximation error of the linear filters. In addition, MHT is unlikely to achieve the values listed in Table I, since it is not possible to have perfect associations in all cases due to heuristics required for MHT such as gating, N-scan-back and pruning. Figure 2 shows an example of such estimation error. We note that MHT with 3-scan-back completely fails to track the turning object in Figure 2.

C. Experiment III - False Alarms

We apply different rates of false alarms to assess the robustness of the algorithm against outliers. The setup is the same as Experiment I but we vary $\lambda V_{\mathcal{R}^2}$ from 1 to 10. We first randomly generated ten scenarios and then, for each scenario, different false alarm rate was applied to generate a test case. So we have a total of 100 test cases. We ran each test case 10 times (the first 20,000 samples are used as burn-ins). Figure 4 and 3 show that the algorithm

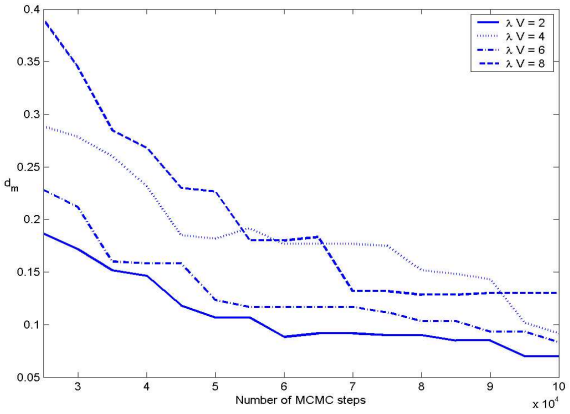

Fig. 3. Experiment III: d_m vs. number of MCMC samples

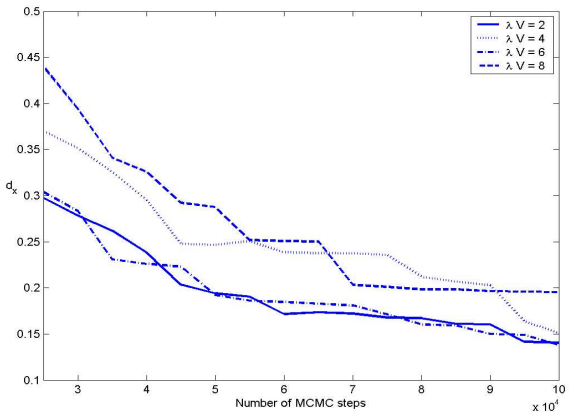

Fig. 4. Experiment III: d_x vs. number of MCMC samples

is robust against false alarms (results for $\lambda V_{\mathcal{R}^2} = 2, 4, 6, 8$ are shown).

D. Experiment IV - Online DBMS RJMCMC

The algorithm described in Section III-B is a batch algorithm. A suboptimal online version of the algorithm can be easily implemented by sliding a window of length T_w. The setting is the same as Experiment I except $T = 500$ and $L = 250$. A single scenario was randomly generated. Three different window sizes $T_w = 10, 20, 30$ are considered and windows are forwarded by a single time step. At each online step, the algorithm is run for a fixed number of MCMC samples. For each window size, we have tried three different numbers of MCMC samples (2000, 4000, and 8000). So there are nine cases and we ran each case 10 times. The average performance of our algorithm (written in Matlab on a PC with a 2.6-GHz Intel processor) is shown in Table II where d_m and d_x are measured over the whole duration T. The performance improves as we increase the window size and number of MCMC samples at the expense of increasing execution time. Note that the average values of d_x and d_m in Table II are smaller than the values reported in earlier

TABLE II

PERFORMANCE OF ONLINE DBMS RJMCMC

T_w	MCMC samples (burn-ins)	d_m	d_x	sec / time step
10	2000 (1000)	0.1208	0.1467	5.1654
	4000 (2000)	0.0739	0.0987	10.2112
	8000 (4000)	0.0444	0.0663	20.2626
20	2000 (1000)	0.1117	0.1379	5.7962
	4000 (2000)	0.0675	0.0928	11.4747
	8000 (4000)	0.0453	0.0666	22.8454
30	2000 (1000)	0.0987	0.1261	6.3159
	4000 (2000)	0.0671	0.0934	12.5211
	8000 (4000)	0.0507	0.0732	25.0049

experiments. This is because for this lengthy scenario there are intervals in which no targets are present and when the algorithm correctly identifies that there are no targets, the resulting d_x and d_m are zeros in those intervals, thus lowering the overall averages.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a dynamic Bayesian model selection framework for layered dynamic systems in which the model selection problems are solved sequentially. An efficient algorithm based on reversible jump Markov chain Monte Carlo is described and extended to online computations. The multitarget tracking problem is formulated as an instance of the dynamic Bayesian model selection problem. We have shown that the algorithm is robust against outliers. In some cases, when we apply DBMS to nonlinear dynamics, DBMS outperforms any linear filtering algorithm with perfect associations. This framework can be easily generalized to other applications such as signal processing and computer vision by using it as a general dynamic pattern recognizer. We are currently working on extending the algorithm for the identification and estimation of hierarchical systems and hybrid systems.

VI. ACKNOWLEDGMENTS

This work was supported by DARPA F30602-00-2-0538 and ARO MURI DAAD19-02-1-0383.

REFERENCES

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd Inter. Symp. on Information Theory*, pages 267–281, 1973.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. In *Machine Learning*, volume 50, pages 5–43, Jan-Feb 2003.

[3] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering Series 179 Academic Press, San Diego, CA, 1988.

[4] I. Beichl and F. Sullivan. The metropolis algorithm. In *Computing in Science and Engineering*, volume 2(1), pages 65–69, 2000.

[5] S. Brooks and P. Giudici. Convergence assessment for reversible jump mcmc simulations. In *Bayesian Statistics*, volume 6. Oxford University Press, 1998.

[6] I. Cox. A review of statistical data association techniques for motion correspondence. In *International Journal of Computer Vision*, volume 10(1), pages 53–66, 1993.

[7] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 1995.

[8] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics Series. Chapman and Hall, 1996.

[9] P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. In *Biometrika*, volume 82, pages 711–732, 1995.

[10] Mark Hansen and Bin Yu. Model selection and minimum description length principle. In *J. Amer. Statist. Assoc.*, volume 96, pages 746–774, 2001.

[11] C. Hue, J.-P. Le Cadre, and P. Perez. Sequential monte carlo methods for multiple target tracking and data fusion. In *IEEE Trans. on Signal Processing*, volume 50(2), pages 309–325, February 2002.

[12] M. Jerrum and A. Sinclair. The markov chain monte carlo method: An approach to approximate counting and integration. In Dorit Hochbaum, editor, *Approximations for NP-hard Problems*. PWS Publishing, Boston, MA, 1996.

[13] R. Kass and A. Raftery. Bayes factors. In *Journal of the American Statistical Association*, volume 90, pages 773–795, 1995.

[14] X. Rong Li and Vesselin P. Jilkov. A survey of maneuvering target tracking. In *In SPIE: Signal and Data Processing of Small Targets*, volume 4048-22, April 2000.

[15] Songhwai Oh. *Multiple Target Tracking for Surveillance*. Technical Report UCB/ERL MO3/54, U.C. Berkeley, 2003.

[16] A.B. Poore. Multidimensional assignment and multitarget tracking. In *Partitioning Data Sets. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 19, pages 169–196, 1995.

[17] D.B. Reid. An algorithm for tracking multiple targets. In *IEEE Transaction on Automatic Control*, volume 24(6), pages 843–854, December 1979.

[18] S. Richardson and P.J. Green. On bayesian analysis of mixtures with unknown number of components. In *Journal of the Royal Statistical Society, B*, volume 59, pages 731–792, 1997.

[19] C.P. Robert, T. Ryden, and D.M. Titterington. Bayesian inference in hidden markov models through reversible jump markov chain monte carlo. In *Journal of the Royal Statistical Society, B*, volume 62, pages 57–75, 2000.

[20] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[21] G. Schwarz. Estimating the dimension of a model. In *The Annals of Statistics*, volume 6, pages 461–464, 1978.

[22] R.W. Sittler. An optimal data association problem on surveillance theory. In *IEEE Trans. on Military Electronics*, volume MIL-8 (Apr), pages 125–139, April 1964.

[23] J.J. Stein and S.S. Blackman. Generalized correlation of multi-target track data. In *IEEE Trans. Aerosp. Electron. Syst.*, volume AES-11(6), pages 1207–1217, November 1975.

[24] R. Streit and T. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proc. SPIE*, volume 2235, pages 394–405, April 1994.