MULTI-MODAL PARTICLE FILTERING FOR HYBRID SYSTEMS WITH AUTONOMOUS MODE TRANSITIONS

Stanislav Funiak [*,1] **Brian C. Williams** [**]

[*] *MIT Space Systems and AI Laboratories, 77 Mass. Ave., Rm. 37-348, Cambridge, MA 02139 USA, stanf@mit.edu*

[**] *MIT Space Systems and AI Laboratories, 77 Mass. Ave., Rm. 33-330, Cambridge, MA 02139 USA, williams@mit.edu*

Abstract: Model-based diagnosis of embedded systems relies on the ability to estimate their hybrid state from noisy observations. This task is especially challenging for systems with many state variables and autonomous transitions. We propose a fair sampling algorithm that combines Rao-Blackwellised particle filters with a multi-modal Gaussian representation. In order to handle autonomous transitions, we let the continuous state estimates contribute to the proposal distribution in the particle filter. The algorithm outperforms purely simulational particle filters and provides unification of particle filters with hybrid hidden Markov model (HMM) observers.

Keywords: state estimation, hybrid modes, Monte Carlo method, analytic approximations, Kalman filters, filtering techniques, diagnostic inference

1. INTRODUCTION

Embedded systems abound in many real-world applications, ranging from space probes (Muscettola *et al.*, 1998) and life support chambers (Hofbaur and Williams, 2002) to walking robots (Pratt *et al.*, 1997). These systems exhibit both continous and discrete behavior and tightly interact with their surrounding environment through continuous dynamics. Therefore, they are well fit for hybrid modeling.

Within the model-based diagnosis community, it is often desirable to estimate the state of hybrid systems from a sequence of noisy observations. This task is crucial for diagnosing subtle faults that exhibit themselves only over a lengthy period of time. A challenge is that real-world systems, such as the BIO-Plex Test Complex at NASA Johnson Space Center (Hofbaur and Williams, 2002) have as many as 10,000,000 modes and 20 continuous variables. At the same time, the systems exhibit non-linear dynamics and autonomous transitions triggered by the continuous dynamics. The sheer size and complexity of these systems make the hybrid state estimation problem very challenging.

In recent years, particle filtering methods have been on the rise (Verma *et al.*, 2001; Dearden and Clancy, 2002; Koutsoukos *et al.*, 2002). Particle filters approximate the posterior distribution with a set of samples that simulate the probabilistic model of the system. Thus, they are applicable to a range of general, non-Gaussian, non-linear models. However, with a few exceptions, these

[1] Supported by NASA under contract NAG2-1388.

methods are purely simulational in the sense that they sample the complete state space. Hence, for large systems, the sample size is too large to be practical.

On the other side of the spectrum stand the multi-modal filtering methods, which represent the belief state as a mixture of Gaussians, e.g. (Li and Bar-Shalom, 1996; Hanlon and Maybeck, 2000; Lerner *et al.*, 2000; Hofbaur and Williams, 2002). Their Gaussian representation and focused search provide an efficient solution to high-dimensional problems. At the same time, non-linearities and merging can introduce significant bias in the estimate.

Clearly, there has been a gap between these two methods: on one end are statistically robust sampling algorithms and on the other, analytical representation algorithms that scale. Freitas (2002) explored this gap in a Rao-Blackwellised particle filtering algorithm. By disallowing autonomous transitions, he was able to decouple the discrete and continuous state and only sample the modes. Yet, autonomous transitions are pervasive in many areas, including rocket propulsion (Koutsoukos *et al.*, 2002) and biological systems (Hofbaur and Williams, 2002). In our previous work, we have addressed autonomous transitions by interpreting multiple-model filtering and AI-based search methods in terms of hybrid HMM-style prediction and update equations. However, we have not addressed the issues of sampling.

The key contribution of this paper is a multi-modal sampling algorithm for hybrid estimation in the presence of autonomous mode transitions. The algorithm samples mode trajectories and, for each trajectory, estimates the continuous state with a Kalman Filter. Hence, it can also be viewed as a specialization of Rao-Blackwellised particle filtering (Murphy and Russell, 2001) with further approximations. The key insight to handling autonomous transitions is that continuous estimates are reused in the importance sampling step of the particle filter. The algorithm is thus substantially more efficient than purely simulational particle filters. It provides an elegant unification of particle filtering with multiple-model filtering and hybrid Markov observers.

2. HYBRID SYSTEM MODELING

2.1 Example: Acrobatic Robot

Consider the following model of an acrobatic robot with two degrees of freedom, swinging on a high bar (see Figure 1). The robot has two links – the torso and the legs – with point masses m_1 and m_2 at their ends. The dynamic model for this system has four continuous variables $\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2$ and

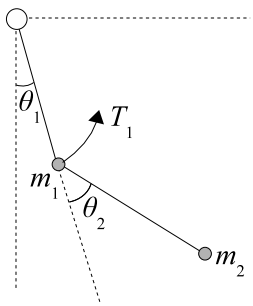

Fig. 1. Acrobatic robot with 2 degrees of freedom.

a discrete mode x_d. If we let $\mathbf{x}_c = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$, its evolution can be expressed as

$$\dot{\mathbf{x}}_c = [\,\dot{\theta}_1, f_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T_1; x_d), \\ \dot{\theta}_2, f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T_1; x_d)] + \mathbf{v}_x \quad (1)$$

where f_1 and f_2 are non-linear functions (Paul, 1982), T_1 is the desired torque, and $\mathbf{v}_x$ is the model uncertainty. The system is underactuated: in order to move, the robot can only apply torque around its center mass.

The system can be in four modes, $m_{0,ok}$, $m_{1,ok}$, $m_{0,failed}$, and $m_{1,failed}$, representing whether or not the robot holds a ball, which increases m_2, and whether or not the actuator has failed. The actuator can fail at all times when torque is exerted with low probability. Furthermore, if the robot is far to the right ($\theta_1 > 0.7$), it captures a ball with probability 0.01 in each time step. If the robot holds a ball and $\theta_1 \leq 0.7$, it will lose the ball with probability 0.01 in each time step. Clearly, capturing a ball is an example of an autonomous mode transition: the transition probabilities depend on the continuous state (see Figure 2).

2.2 Probabilistic Hybrid Automata

Formally, the system can be described as a Probabilistic Hybrid Automaton (PHA), a formalism merging hidden Markov models (HMM) with continuous dynamical system models (Hofbaur and Williams, 2002). It is a tuple $\langle \mathbf{x}, \mathbf{w}, F, T, \mathcal{X}_d, \mathcal{U}_d, T_s \rangle$:

- $\mathbf{x}$ denotes the hybrid state of the automaton, composed of variables $\{x_d\} \cup \mathbf{x}_c$.[2] The discrete variable x_d with finite domain $\mathcal{X}_d$ represents the *operational mode* of the system, while the continuous vector $\mathbf{x}_c \in \mathbb{R}^{n_x}$ denotes the *continuous state*. The initial state probability $p(\mathbf{x}_0)$ is assumed to be known and the conditional state distribution for each mode $p(\mathbf{x}_{c,0}|x_{d,0})$ is Gaussian.

[2] When clear from the context, we use lowercase bold symbols, such as $\mathbf{v}$, to denote a *set* of variables $\{v_1, \ldots, v_l\}$, as well as a *vector* $[v_1, \ldots, v_l]^T$ with components v_i.

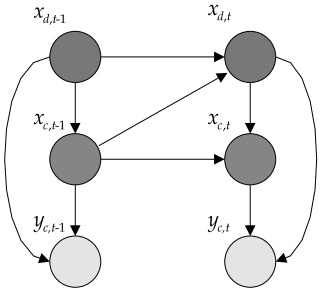

Fig. 2. Conditional dependencies among the state variables $\mathbf{x}_c, x_d$ and the output $\mathbf{y}_c$ expressed as a Dynamic Bayesian Network. The edge from $\mathbf{x}_{c,t-1}$ to $x_{d,t}$ represents the dependence of $x_{d,t}$ on $\mathbf{x}_{c,t-1}$, i.e., autonomous transitions.

- $\mathbf{w} = \mathbf{u}_d \cup \mathbf{u}_c \cup \mathbf{y}_c$ denotes the set of *I/O variables*, consisting of disjoint sets of discrete input variables $\mathbf{u}_d \in \mathcal{U}_d$, continuous input variables $\mathbf{u}_c \in \mathbb{R}^{n_u}$, and continuous output variables $\mathbf{y}_c \in \mathbb{R}^{n_y}$.
- The *discrete transition* function $T : \mathcal{U}_d \times \mathbb{R}^{n_u} \times \mathcal{X}_d \times \mathbb{R}^{n_x} \rightarrow \mathcal{T}$ specifies, for each possible assignment of variables $\mathbf{u}_d$, $\mathbf{u}_c$, x_d, and $\mathbf{x}_c$ at time step $t-1$, a distribution $\tau \in \mathcal{T}$ over the modes at the next time step. T is described by a finite set of pairs $\{\langle \tau, c \rangle\}$ of transition distributions $\tau : \mathcal{X}_d \rightarrow [0; 1]$ and guard conditions c that cover distinct regions of the $\mathcal{U}_d \times \mathbb{R}^{n_u} \times \mathcal{X}_d \times \mathbb{R}^{n_x}$ space.
- $F : \mathcal{X}_d \rightarrow \{\langle \mathbf{f}, \mathbf{g}, \mathbf{v}_x, \mathbf{v}_y \rangle\}$ specifies the *continuous evolution* of the automaton for each mode $x_d \in \mathcal{X}_d$ in terms of the transition function $\mathbf{f}$, observation function $\mathbf{g}$, and mode-dependent zero-mean white Gaussian noise $\mathbf{v}_x(x_d)$ and $\mathbf{v}_y(x_d)$:

$$\mathbf{x}_{c,t} = \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{c,t-1}; x_{d,t}) + \mathbf{v}_x(x_{d,t}) \quad (2)$$

$$\mathbf{y}_{c,t} = \mathbf{g}(\mathbf{x}_{c,t}, \mathbf{u}_{c,t}; x_{d,t}) + \mathbf{v}_y(x_{d,t}) \quad (3)$$

Larger systems can be modeled as a composition of multiple PHAs (Hofbaur and Williams, 2002).

2.3 Hybrid Markov Observer

Multi-modal Gaussian filtering encompasses a wide family of methods for hybrid estimation, including Multiple-Model (MM) estimation, adaptive MM estimation, and more recent AI-based methods. The common premise of these methods is that they represent the belief state as a mixture of Gaussians and they use a bank of Kalman Filters to evolve the continuous state. The algorithms vary in how they choose which mode trajectories to expand (qualitatively or quantitatively) and how they merge continuous state estimates.

The contribution of Hofbaur and Williams (2002) was to interpret these algorithms in terms of HMM-style hybrid belief-state update equations. For example, they define their beam search algorithm in terms of prediction and update equations

$$h_{\bullet t}[\hat{\mathbf{x}}_i] = P_{\mathcal{T}}(m_i | \hat{\mathbf{x}}_{j,t-1}, \mathbf{u}_{d,t-1}) h_{t-1}[\hat{\mathbf{x}}_j] \quad (4)$$

$$h_t[\mathbf{x}_i] = \frac{h_{\bullet t}[\hat{\mathbf{x}}_{i,t}] P_{\mathcal{O}}(\mathbf{y}_{c,t} | \hat{\mathbf{x}}_{i,t})}{\sum_j h_{\bullet t}[\hat{\mathbf{x}}_j] P_{\mathcal{O}}(\mathbf{y}_{c,t} | \mathbf{x}_{j,t})}. \quad (5)$$

In these equations, $h_{\bullet t}[\hat{\mathbf{x}}_i]$ denotes an intermediate hybrid belief-state that is based on transition probabilities only. Hybrid estimation determines the possible transitions for each $\hat{\mathbf{x}}_{j,t-1}$ at the previous time step, thus specifying candidate trajectories to be tracked by the filter bank. Kalman filtering then provides the new hybrid state $\hat{\mathbf{x}}_{i,t}$. Finally, the intermediate belief state is adjusted using the observation function $P_{\mathcal{O}}$.

3. PARTICLE FILTERING: REVIEW

Particle filters belong to the family of Monte Carlo simulation-based methods and are applicable to a range of domains. Given a discrete-time hybrid system (e.g. a PHA), particle filters approximate the posterior of the hybrid state $\mathbf{x}_t$ with a set of sample trajectories $\{\mathbf{x}_{0:t}^{(i)}\}$. The trajectories are traced sequentially and drawn from the posterior distribution $p(\mathbf{x}_{0:t} | \mathbf{y}_{1:t})$.[3] They approximate the posterior in the sense that the posterior probability over a sufficiently large region $\mathcal{A} \subset \mathcal{X}_d \times \mathbb{R}^{n_x}$ is approximated by the number of samples in it:

$$P(\mathbf{x}_t \in \mathcal{A} | \mathbf{y}_{1:t}) \approx \frac{1}{N} |\{i : \mathbf{x}_t^{(i)} \in \mathcal{A}\}| \quad (6)$$

Particle filtering algorithms typically involve three steps, as shown in Figure 3. The algorithm starts by drawing samples from the initial distribution $p(\mathbf{x}_0)$; thus, effectively approximating the posterior at $t = 0$. Then, in each iteration, we evolve the hybrid trajectories by taking *one* random sample $\tilde{\mathbf{x}}_t$ for each trajectory $\mathbf{x}_{0:t-1}^{(i)}$ according to a so-called *proposal distribution* and compute the importance weights (step 2). In its simplest form, the proposal distribution is just the transition distribution $p(\mathbf{x}_t | \mathbf{x}_{t-1})$. This step, called *importance sampling*, is then entirely analogous to the prediction-update sequence in other filtering methods: first, we predict the hybrid state $\tilde{x}_t$ using the estimate at the previous time step and then we adjust the prediction using the newest observation. The final selection step simply multiplies the "good" particles and removes the "bad" ones, so that in the next time step, the high-likelihood particles contribute more to the sampling process. Otherwise, most particles would have zero weight after a few iterations due to the accumulated

[3] We use the notation $v_{k:l}$ to denote the tuple $\langle v_k, v_{k+1}, \ldots, v_l \rangle$.

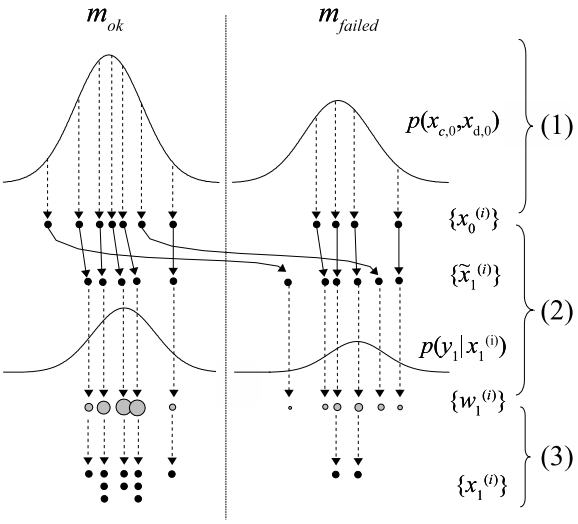

Fig. 3. The three steps of a particle filter: Initialization (1), Importance sampling (2), and Selection (3). The illustrated system has two modes and one continuous variable.

errors in their predictions, and the filter would not converge to the posterior.

Particle filters are conceptually simple, yet offer interesting generalizations of the ideas presented above. First, the particles approximate not only the posterior over the present state $\mathbf{x}_t$, but also over the space of *complete trajectories* traced by the state transitions, albeit with less precision. Furthermore, in principle, the proposal distribution can be an arbitrary function $q(\mathbf{x}_t; \mathbf{x}^{(i)}_{0:t-1}, \mathbf{y}_{1:t})$, not only the transition distribution $p(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1})$. For example, the proposal distribution can directly incorporate present observations and thus, make more precise predictions. It has been shown that as long as the support of q includes the support of $p(\mathbf{x}_t|\mathbf{x}^{(i)}_{0:t-1}, \mathbf{y}_{1:t})$ and we let the weights

$$\tilde{w}_t^{(i)} = \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}^{(i)}_{0:t}, \mathbf{y}_{1:t-1}) p(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}^{(i)}_{0:t-1}, \mathbf{y}_{1:t-1})}{q(\tilde{\mathbf{x}}_t; \tilde{\mathbf{x}}^{(i)}_{0:t-1}, \mathbf{y}_{1:t})}, \tag{7}$$

the particle filter converges to the true posterior in the limit. See (Doucet *et al.*, 2001) for a comprehensive review of particle filters and derivation of the results above.

4. MULTI-MODAL PARTICLE FILTERING

Our algorithm combines particle filtering with a multi-modal Gaussian representation. It still performs particle filtering, but reduces the dimensionality of the sampled space by applying the method of Rao-Blackwellised particle filters (Murphy and Russell, 2001): If we divide the state variables into two sets, the root (sampled) variables $\mathbf{r}$ and leaf variables $\mathbf{s}$, we can express the posterior distribution of $\mathbf{x}$ as

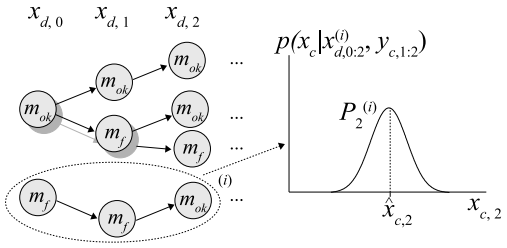

Fig. 4. Representation in the multi-modal particle filter. Note that the trajectory $m_{ok} \rightarrow m_f$ was duplicated in the selection step at $t = 1$.

(1) **Initialization**
- sample $x^{(i)}_{d,0} \sim p(x_{d,0})$, $i = 1, \ldots, N$
- KF initialization: for $i = 1, \ldots, N$, let $\hat{\mathbf{x}}^{(i)}_{c,0} = \mathbb{E}[\mathbf{x}_{c,0}|x^{(i)}_{d,0}]$ and $P_0^{(i)} = \Lambda_{\mathbf{x}_{c,0}|x^{(i)}_{d,0}}$

(2) **Importance sampling step**
- for $i = 1, \ldots, N$, compute the proposal distribution $p(x_{d,t}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1})$
- sample $\tilde{x}^{(i)}_{d,t} \sim p(x_{d,t}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1})$ and set $\tilde{x}^{(i)}_{d,0:t} \leftarrow (x^{(i)}_{d,0:t-1}, \tilde{x}^{(i)}_{d,t})$
- evaluate the importance weights $\tilde{w}^{(i)}_t$
- normalize the importance weights

(3) **Selection step**
- resample (with replacement) N particles from $\{\tilde{x}^{(i)}_{d,0:t}\}$ according to $\tilde{w}^{(i)}_t$ to obtain samples $\{x^{(i)}_{d,0:t}\}$ distributed approximately according to $p(x^{(i)}_{d,0:t}|\mathbf{y}_{c,1:t})$.

(4) **Exact step**
- for $i = 1, \ldots, N$, update $\hat{\mathbf{x}}^{(i)}_{c,t}$ and $P_t^{(i)}$ with an EKF using $\mathbf{f}(\cdot, \mathbf{u}_{c,t-1}; x^{(i)}_{d,t})$, $\mathbf{g}(\cdot, \mathbf{u}_{c,t}; x^{(i)}_{d,t})$, $\Lambda_{\mathbf{v}_x(x^{(i)}_{d,t})}$, and $\Lambda_{\mathbf{v}_y(x^{(i)}_{d,t})}$
- let $t \leftarrow t + 1$ and go to step 2

Fig. 5. Multi-modal particle filter for PHA state estimation.

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= p(\mathbf{s}_t, \mathbf{r}_t|\mathbf{y}_{1:t}) \\ &= \int p(\mathbf{s}_t, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{r}_{0:t-1} \\ &= \int p(\mathbf{s}_t|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}) p(\mathbf{r}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{r}_{0:t-1} \end{aligned} \tag{8}$$

Thus, we expand the posterior in terms of the root trajectory $\mathbf{r}_{0:t}$ and leaf $\mathbf{s}_t$ conditioned on the trajectory (typically, the leaf variables $\mathbf{s}$ would be descendants of the roots $\mathbf{r}$ in a DBN structure, hence the root-leaf analogy). The key to this formulation is that if we can compute $p(\mathbf{s}_t|\mathbf{r}_{0:t}, \mathbf{y}_{1:t})$ analytically, we only need to sample the root variables $\mathbf{r}$. In this manner, fewer particles are needed to cover the posterior.

In the spirit of the beam search algorithm of Hofbaur and Williams (2002), we sample the mode trajectories and, conditioned on these trajectories, compute the posterior of the continuous variables analytically (see Figure 4). This partition reveals

significant structure: given a sampled trajectory $x^{(i)}_{d,0:t}$, the transition and observation distributions are known. The distribution $p(\mathbf{x}_{c,t}|x^{(i)}_{d,0:t}, \mathbf{y}_{1:t})$ is then approximately Gaussian [4] and can be efficiently updated with a Kalman Filter.

The resulting multi-modal particle filtering algorithm is shown in Figure 5. Each particle now holds not only a sample trajectory $x^{(i)}_{d,0:t}$ drawn approximately from the posterior distribution of $x_{d,0:t}$ but also the estimated mean $\hat{\mathbf{x}}^{(i)}_{c,t}$ and covariance matrix $P^{(i)}_t$ of $p(\mathbf{x}_{c,t}|x^{(i)}_{d,0:t}, \mathbf{y}_{c,1:t})$. There is also an additional Kalman Filtering step that updates $\hat{\mathbf{x}}^{(i)}_{c,t}$ and $P^{(i)}_t$ for each particle i based on the transition and observation equations associated with the latest mode $x^{(i)}_{d,t}$.

As before, the algorithm starts by initializing the particles, albeit only the modes are sampled. The algorithm then proceeds to expand each trajectory i probabilistically by taking one random sample from the corresponding proposal distribution. After we compute the importance weights, we resample the trajectories according to their weights, so as to direct their future expansion into relevant regions of the trajectory space.

The proposal distribution $p(x_{d,t}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1})$ is similar in its form to the transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in Markov processes. However, it is conditioned on a complete trajectory and all previous observations, rather than simply on the previous state. This is because $\{x_{d,t}\}$ alone is *not* an HMM process: due to the autonomous transitions, knowing $x_{d,t-1}$ alone does not tell us what the distribution of $x_{d,t}$ is. The distribution of $x_{d,t}$ is known only when conditioned on the mode *and* continuous state in the previous time step (see Figure 2).

This observation suggests that in order to compute the proposal distribution, we need to expand it in terms of the previous estimates:

$$
\begin{aligned}
&p(x_{d,t}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1}) \\
&= \int_{\mathbf{x}_{c,t-1}} p(x_{d,t}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1}, \mathbf{x}_{c,t-1}) \cdot \\
&\qquad\qquad p(\mathbf{x}_{c,t-1}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1}) d\mathbf{x}_{c,t-1} \\
&= \int_{\mathbf{x}_{c,t-1}} p(x_{d,t}|x^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}) \cdot \\
&\qquad\qquad p(\mathbf{x}_{c,t-1}|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1}) d\mathbf{x}_{c,t-1} \\
&= \sum_{\langle \tau, c \rangle \in T} \tau(x_{d,t}) P(c(\mathbf{x}_{c,t-1})|x^{(i)}_{d,0:t-1}, \mathbf{y}_{c,1:t-1}) \quad (9)
\end{aligned}
$$

[4] More precisely, $p(\mathbf{x}_{c,t}|x^{(i)}_{d,0:t}, \mathbf{y}_{1:t})$ is Gaussian if and only if all $\mathbf{f}$ and $\mathbf{g}$ along the trajectory are linear and there are no autonomous transitions.

Here, the first equality follows from the total probability theorem. The second equality comes from the independence assumption made in the model (Figure 2). The third, final equality holds because the mode distribution τ is the same for all values of $\mathbf{x}_{c,t-1}$ that satisfy its associated guard condition c. Only the guard conditions that apply to inputs $\mathbf{u}_{d,t-1}$ and $\mathbf{u}_{c,t-1}$ are considered.

The final term in equation 9 is the probability that a guard condition is satisfied, given the trajectory and observations to time $t-1$. But then, $\mathbf{x}_{t-1}$ is distributed (approximately) as $\mathcal{N}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, P^{(i)}_{t-1})$, so the final term is simply an integral over a Gaussian multivariate distribution with mean $\hat{\mathbf{x}}^{(i)}_{c,t-1}$ and covariance $P^{(i)}_{t-1}$. Depending on the form of the guard condition, this probability can be evaluated more or less efficiently. For (hyper)rectangular regions, efficient approximations exist (Joe, 1995), and convex (possibly unbounded) regions can be reduced to rectangular ones with a linear transform. For other cases, one can always fall back to Monte Carlo simulations.

Given our choice of the proposal distribution, the weight $\tilde{w}^{(i)}_t$ in equation 7 simplifies to

$$\tilde{w}^{(i)}_t = p(\mathbf{y}_{c,t}|\mathbf{y}_{c,1:t-1}, x^{(i)}_{d,0:t}) \quad (10)$$

Unfortunately, equation 10 is rather hard to evaluate efficiently. Even if we expand the weight in terms of the continuous state as

$$\int_{\mathbf{x}_{c,t}} p(\mathbf{y}_{c,t}|\mathbf{x}_{c,t}, x^{(i)}_{d,t}) p(\mathbf{x}_{c,t}|x^{(i)}_{d,0:t}, \mathbf{y}_{c,1:t-1}) d\mathbf{x}_{c,t}, \quad (11)$$

the second integrand term is still non-standard in the presence of autonomous transitions.

We address this issue by ignoring the autonomous transitions for the purpose of computing importance weights. In this case, the integrand is a product of two Gaussians (assuming linearized transitions and observations), and the weight can be computed using the measurement residual

$$\mathbf{r}_t = \mathbf{y}_{c,t} - \mathbf{g}(\mathbf{f}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{u}_{c,t-1}; x^{(i)}_{d,t}), \mathbf{u}_{c,t}; x^{(i)}_{d,t}) \quad (12)$$

from the prediction step of an EKF:

$$\tilde{w}^{(i)}_t \approx \mathcal{N}(\mathbf{r}^{(i)}_t, \mathbf{S}^{(i)}_t) \quad (13)$$

An open research problem is to approximate $\tilde{w}^{(i)}_t$ better in the presence of autonomous transitions.

Note that the expressions derived for the proposal distribution and the importance weight (equations 9 and 13, respectively) are precisely the probabilistic hybrid transition function $P_{\mathcal{T}}$ and the hybrid observation function $P_{\mathcal{O}}$ in equations 4-5. By leveraging Rao-Blackwellisation, the algorithm elegantly unifies with HMM-based hybrid observers and multiple-model estimation. In particular, advances in one method, e.g. a better approximation

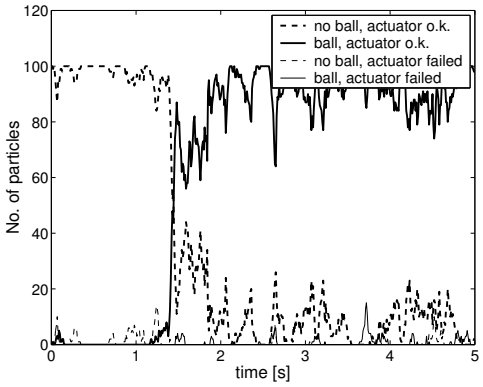

Fig. 6. The distribution of particles across the four modes for the robot ball capture scenario.

to $\tilde{w}_t^{(i)}$ in the particle filter or a more accurate representation of the belief state in search-based algorithms, may lead to improvements in the other methods.

5. DISCUSSION

In our experiments, we simulated the motion of the acrobatic robot with a differential equation solver for three scenarios and generated noisy observations seen by the filter. Our implementation only maintains the latest mode for each trajectory, since the latest mode, along with the continuous estimate, provide sufficient statistics for the sampling and Kalman filtering steps.

Figure 6 shows a typical execution of the algorithm with 100 particles when $T_s = 0.01$s. Initially, the robot is straight at 45 degrees from the vertical. After swinging for one round, the robot receives a ball and keeps it, which is correctly detected by the filter.

For comparison, we also implemented a simple bootstrap particle filter. However, the filter did not converge even with 5000 particles. In the near future, we hope to conduct a more rigorous comparison with state-of-the-art particle filters and the Gaussian beam filter used within hybrid mode estimation (Hofbaur and Williams, 2002).

This paper demonstrated an efficient sampling algorithm for hybrid models with autonomous transitions. By sampling from a proposal distribution that can be efficiently expressed using the previous continuous state estimates, the algorithm overcomes coupling between the discrete and continuous state. It provides natural unification of Rao-Blackwellised particle filtering with multiple-model filtering and methods based on HMM-style hybrid prediction and update equations.

Acknowledgements

To A. Hofmann of the MIT LEG Laboratory.

REFERENCES

Dearden, R. and D. Clancy (2002). Particle filters for real-time fault detection in planetary rovers. In: *Proceedings of the 13th International Workshop on Principles of Diagnosis (DX02)*. pp. 1–6.

Doucet, A., N. Freitas and N. J. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Freitas, N. (2002). Rao-Blackwellised particle filtering for fault diagnosis. *IEEE Aerospace*.

Hanlon, P. and P. Maybeck (2000). Multiple-model adaptive estimation using a residual correlation kalman filter bank. *IEEE Transactions on Aerospace and Electronic Systems* **36**(2), 393–406.

Hofbaur, M. W. and B. C. Williams (2002). Mode estimation of probabilistic hybrid systems. In: *HSCC 2002* (C.J. Tomlin and M.R. Greenstreet, Eds.). Vol. 2289 of *Lecture Notes in C. S.*. pp. 253–266. Springer Verlag.

Joe, H. (1995). Approximations to multivariate normal rectangle probabilities based on conditional expectations. *Journal of the American Statistical Association* **90**(431), 957–964.

Koutsoukos, X., J. Kurien and F. Zhao (2002). Monitoring and diagnosis of hybrid systems using particle filtering methods. In: *MTNS 2002*.

Lerner, U., R. Parr, D. Koller and G. Biswas (2000). Bayesian fault detection and diagnosis in dynamic systems. In: *Proc. of the 17th National Conference on A. I.*. pp. 531–537.

Li, X.R. and Y. Bar-Shalom (1996). Multiple-model estimation with variable structure. *IEEE Transactions on Automatic Control* **41**, 478–493.

Murphy, K. and S. Russell (2001). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In: *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. Freitas and N. Gordon, Eds.). Chap. 24, pp. 499–515. Springer-Verlag.

Muscettola, N., P. Nayak, B. Pell and B. C. Williams (1998). The new millennium remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence* **103**(1-2), 5–48.

Paul, R. P. (1982). *Robot Manipulators*. MIT Press.

Pratt, J., P. Dilworth and G. Pratt (1997). Virtual model control of a bipedal walking robot. In: *Proc. of the IEEE International Conference on Robotics and Automations (ICRA '97)*.

Verma, V., J. Langford and R. Simmons (2001). Non-parametric fault identification for space rovers. In: *International Symposium on Artificial Intelligence and Robotics in Space (iSAIRAS)*.