Combining Stochastic and Greedy Search in Hybrid Estimation *

Lars Blackmore
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
larsb@mit.edu

Stanislav Funiak
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
sfuniak@cs.cmu.edu

Brian Williams
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
williams@mit.edu

Abstract

Techniques for robot monitoring and diagnosis have been developed that perform state estimation using probabilistic hybrid discrete/continuous models. Exact inference in hybrid dynamic systems is, in general, intractable. Approximate algorithms are based on either 1) greedy search, as in the case of k-best enumeration or 2) stochastic search, as in the case of Rao-Blackwellised Particle Filtering (RBPF).

In this paper we propose a new method for hybrid state estimation. The key insight is that stochastic and greedy search methods, taken together, are often particularly effective in practice. The new method combines the stochastic methods of RBPF with the greedy search of k-best in order to create a method that is effective for a wider range of estimation problems than the individual methods alone. We demonstrate this robustness on a simulated acrobatic robot, and show that this benefit comes at only a small performance penalty.

Introduction

Robotic systems must be able to estimate their operational state, which can evolve both continuously and discretely, and is only partially observable. Probabilistic hybrid models such as Probabilistic Hybrid Automata (Hofbaur & Williams 2002a) and hybrid dynamic Bayesian networks (Lerner *et al.* 2000) are convenient modeling tools for robotic applications. Tasks such as robot monitoring and diagnosis can often be framed as state estimation in hybrid models, by representing the health of the system with discrete variables and its dynamics with continuous ones.

In the general case inference in hybrid models is NP-hard (Lerner & Parr 2001); nevertheless, approximate inference is often feasible. The two predominant approaches to approximate inference in dynamic hybrid models are greedy (k-best) enumeration (Lerner *et al.* 2000; Hofbaur & Williams 2002a) and Rao-Blackwellised Particle Filtering (RBPF) (Doucet *et al.* 2000; Morales-Menendez, de Freitas, & Poole 2002; Funiak & Williams 2003). Both these approaches are based on the idea of representing the belief state by a mixture of Gaussians for a subset of the trajectories traced by the discrete state. The former approach enumerates the trajectories in best-first order, while the latter evolves them through sampling.

While prior work (Hutter & Dearden 2003) has compared the performance of RBPF to other particle filters, there has been little empirical comparison of RBPF and k-best methods. Such an analysis is crucial to understanding the tradeoffs between the two methods, and to developing a new approach that combines the strengths of both. In this paper we carry out the comparison and show that both approaches have limitations, depending on whether the posterior distribution is concentrated in few discrete mode trajectories, or is relatively flat across many different trajectories. These results demonstrate the need for a new algorithm that is robust to changes in the variance of the approximated posterior distribution.

The key insight behind the new algorithm is that, in many AI methods, a combination of stochastic and greedy search methods can be effective in practice. This is analogous to the 'exploration vs. exploitation' tradeoff, which has been used with great success in Constraint Satisfaction Problems (CSP), for example. The new algorithm uses Rao-Blackwellised particle filtering to generate, stochastically, additional candidates for k-best enumeration that would not have been tracked by a purely greedy approach. The algorithm maintains a set of particles, which are updated using RBPF, and a set of mode trajectories with the highest posterior probability, generated by both RBPF and search-based successor enumeration. This algorithm makes use of the efficient properties of both k-best enumeration and RBPF, while being probabilistically sound.

We demonstrate, using a simulated acrobatic robot, that the mixed algorithm is effective for both a concentrated and flat posterior distribution, showing a dramatic increase in robustness for a relatively small performance penalty.

Hybrid State Estimation

We consider the problem of estimating (filtering) the hidden state of dynamic systems, modeled as Concurrent Probabilistic Hybrid Automata (CPHA). CPHA model the system as a set of n concurrently operating Probabilistic Hybrid Automata (PHA) that interact through continuous inputs and outputs. The hidden state of each automaton k consists of a discrete random process X_d^k and a set of continuous random processes $\mathbf{X}_c^k$. The state of the CPHA is observed indirectly

*This work is supported by NASA Ames Award NCC2-1235, and NASA Award NNA04CK91A

through a continuous observation process $\mathbf{Y}_c$. Each discrete state assignment (mode) $X_d^k = x_d^k$ is associated with a set of algebraic equations $\mathcal{F}_{AE}(x_d^k)$ and difference equations $\mathcal{F}_{DE}(x_d^k)$ that govern the continuous evolution of the automaton. Given a joint assignment $\mathbf{X}_d = \mathbf{x}_d$ to the discrete state of all automata $1, \ldots, n$, the transition and observation distributions for the continuous state of the overall CPHA, $\mathbf{X}_c \triangleq \cup_k \mathbf{X}_c^k$, are obtained by taking the union of equations for the individual automata (Hofbaur & Williams 2002a),

$$\bigcup_{k=1}^{n} \mathcal{F}_{AE}(x_d^k) \cup \mathcal{F}_{DE}(x_d^k). \tag{1}$$

Under the assumption of Gaussian noise, this representation leads to transition and observation distributions that take on the form of a conditional non-linear Gaussian model

$$\begin{aligned} p(\mathbf{x}_{c,t}|\mathbf{x}_{d,t}, \mathbf{x}_{c,t-1}) &= \mathcal{N}(\mathbf{f}(\mathbf{x}_{c,t-1}; \mathbf{x}_{d,t}), \Sigma_{x,\mathbf{x}_{d,t}}) \\ p(\mathbf{y}_{c,t}|\mathbf{x}_{d,t}, \mathbf{x}_{c,t}) &= \mathcal{N}(\mathbf{g}(\mathbf{x}_{c,t}; \mathbf{x}_{d,t}), \Sigma_{y,\mathbf{x}_{d,t}}). \end{aligned} \tag{2}$$

The means of the transition and observation distributions are given by the functions $\mathbf{f}(\cdot; \mathbf{x}_{d,t})$, $\mathbf{g}(\cdot; \mathbf{x}_{d,t})$, and the covariances are given by $\Sigma_{x,\mathbf{x}_{d,t}}$, $\Sigma_{y,\mathbf{x}_{d,t}}$. These are obtained by symbolically solving the set of equations corresponding to mode $\mathbf{x}_{d,t}$ (Eq. 1).

The discrete evolution of the system is described by a set of probabilistic transitions and the associated guards over the continuous variables, see Fig. 2. Given this description, the discrete transition distribution for the CPHA, $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1})$, decomposes as a product of the transition distributions for the individual automata:

$$p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1}) = \prod_{k=1}^{n} p(x_{d,t}^k|x_{d,t-1}^k, \mathbf{x}_{c,t-1}). \tag{3}$$

The structure in the model can be exploited to compute the transition model for the desired discrete assignments $\mathbf{x}_{d,t}$ on-line (Hofbaur & Williams 2002a) and aids in further decomposition of the model (Hofbaur & Williams 2002b).

The goal of hybrid state estimation is to determine the posterior distribution of the hidden (discrete and continuous) state given all the observations so far, $p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t}|\mathbf{y}_{1:t})$. This distribution can be expressed as a sum of posterior distributions for all trajectories that end in state $\mathbf{x}_{d,t}$:

$$p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t}|\mathbf{y}_{1:t}) = \sum_{\mathbf{x}_{d,1:t-1}} p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t}|\mathbf{y}_{1:t}). \tag{4}$$

Each summand can be further expanded as a product of the posterior probability of the discrete mode trajectory $\mathbf{x}_{d,1:t}$ and the posterior distribution of the continuous state, conditioned on this mode trajectory:

$$p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{d,1:t}|\mathbf{y}_{1:t})p(\mathbf{x}_{c,t}|\mathbf{x}_{d,1:t}, \mathbf{y}_{1:t}). \tag{5}$$

This decomposition leads to a natural representation of the belief state as a mixture of Gaussians, one for each reachable mode trajectory $\mathbf{x}_{d,1:t}$. Given $\mathbf{x}_{d,1:t}$, the second term can be approximated as a Gaussian, using a combination of a Kalman Filter and numerical integration techniques, such as Gaussian Quadrature and Exact Monomials (Lerner 2002).

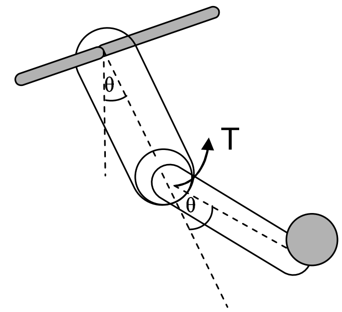

Figure 1: Schematic diagram of an acrobatic robot.

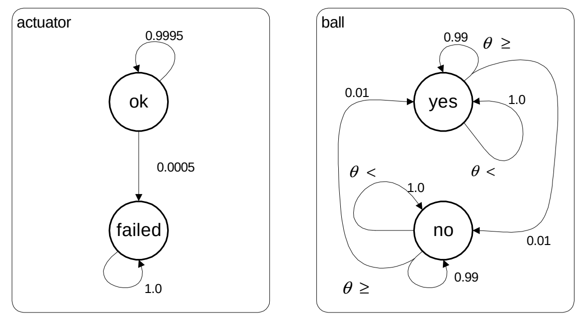

Figure 2: Discrete transition model for an acrobatic robot. If the actuator has failed, it exerts no torque. When the robot catches the ball, `ball=yes`, and the mass of the lower link increases.

The weight of each mixture component is then computed using the belief state update (Funiak & Williams 2003)

$$p(\mathbf{x}_{d,1:t}|\mathbf{y}_{1:t}) = b(\mathbf{x}_{d,1:t}) \propto P_O \cdot P_T \cdot b(\mathbf{x}_{d,1:t-1}). \tag{6}$$

In this equation, $P_T \triangleq p(\mathbf{x}_{d,t}|\mathbf{x}_{d,1:t-1}, \mathbf{y}_{1:t-1})$ is the probability of transitioning to a state $\mathbf{x}_{d,t}$, given the past mode trajectory and past observations, and $P_O \triangleq p(\mathbf{y}_t|\mathbf{x}_{d,1:t}, \mathbf{y}_{1:t-1})$ is the measurement update. Both P_T and P_O can be calculated approximately using efficient methods described in (Hofbaur & Williams 2002a).

Failure Detection with k-best, RBPF

It is intractable to compute the probability in Eq. 5 for all possible discrete mode trajectories, since their number increases exponentially with time. A common approach is to track only a subset of mode trajectories $\{\mathbf{x}_{d,1:t}^{(i)}\}$, $i = 1, \ldots, K$ and the corresponding continuous estimates $p(\mathbf{x}_{c,t}|\mathbf{x}_{d,1:t}^{(i)}, \mathbf{y}_{1:t})$, and recursively evolve this set at each time step. This approach is taken in both Rao-Blackwellised particle filtering (RBPF) and k-best enumeration.

RBPF works by evolving the trajectories probabilistically. At each time step, the filter evolves the samples according to a *proposal distribution*, which is typically a tractable distribution, such as P_T or a product of P_T and an approximation to P_O. Each sample i is then assigned a weight $w_t^{(i)}$ that adjusts for the differences in the proposal and the value $P_T \cdot P_O$.

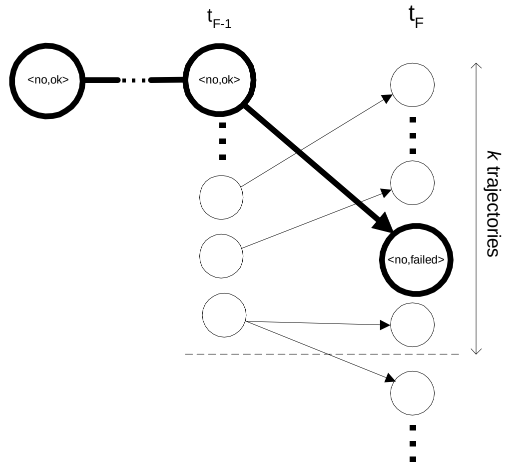

Figure 3: Failure detection with k-best enumeration. In this figure, the trajectories are shown ordered in terms of their posterior probability, with the most likely at the top. Trajectories below the dashed line are discarded. The true mode trajectory is shown in bold.

Conditioned on the newly evolved state $\mathbf{x}_{d,t}^{(i)}$, we update the continuous state with a Kalman Filter, as discussed in the previous section. The samples are periodically resampled according to their weights, in order to multiply the fitting samples and discard the unlikely ones.

The k-best filter (Hofbaur & Williams 2002a), on the other hand, attempts to capture most of the probability space by greedily expanding the trajectories with the highest posterior probability $b(\mathbf{x}_{d,1:t})$. Given a set of trajectories, the filter enumerates a set of successor trajectories in decreasing order of posterior probability, exact up to the approximations in P_T and P_O and the normalization factor in the belief state update. For CPHA, this enumeration can be done efficiently, by expanding the trajectories componentwise and upper-bounding the remaining probability, with a combination of A* and branch-and-bound search (Hofbaur & Williams 2002a).

To illustrate the algorithms, consider the model of an acrobatic robot in Figs. 1 and 2. The hidden continuous state consists of four variables, representing the angles and angular velocities at two joints. Only the angle of the middle joint is observed, with additive Gaussian noise. The hidden discrete state consists of two variables, `actuator` and `ball`, representing the health state of the robot's actuator and whether or not it carries a ball. These two discrete variables together determine the equations governing the evolution of the system, assuming that the weight of the ball is known.

Suppose that the robot does not carry a ball, and its actuator is functional, up to time step t_F, when a failure occurs. As illustrated in Fig. 3, the k-best enumeration algorithm maintains the nominal trajectory, and will detect the failure, as long as a trajectory with the fault transition is among the set of leading trajectories at (or near) the time when the actual fault occurs.[1] By contrast, with the RBPF, mode transitions are sampled stochastically (Fig. 4). For the true trajectory to be tracked, the mode transition into the failure state must be sampled. Since this transition has a low prior many particles will be needed, in order to detect the fault.

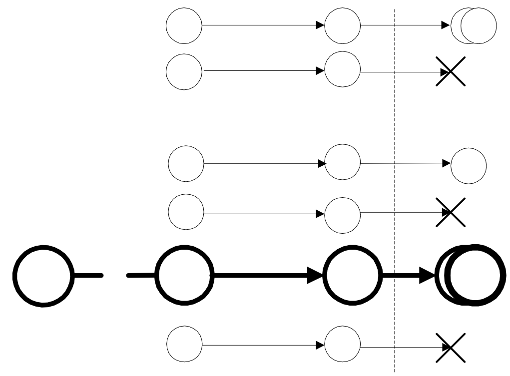

Figure 4: Failure detection with RBPF. The question mark indicates that the generation of the successor corresponding to the true trajectory is generated stochastically.

Now, consider a modified model, in which the probability of catching a ball is 0.5, rather than 0.01. In addition, let the mass of the ball be small, so that the effect of catching a ball, on the observations, is relatively small. In this case, the posterior distribution over the trajectories will be flat, as there will be many trajectories in the belief state that oscillate between the robot having and not having a ball. Initially, these trajectories will have higher posterior probabilities than the ground-truth trajectory, because they have much higher priors than the failure, and it takes several time steps before the evidence for the failure builds up. The RBPF, on the other hand, will occasionally generate a fault sample even if there are many alternative trajectories that have a higher posterior probability. The fault trajectory may thus be present in the set of trajectories *despite having a lower posterior probability than other candidate trajectories*. This behavior makes the RBPF more robust to local maxima.

To summarize, the advantage of k-best enumeration is that it stores the posterior probability of each trajectory exactly, rather than approximating the probability by a number of repeated particles. This observation has been noted in the past (Lerner 2002). However, the ability of k-best enumeration to track the true trajectory is *critically dependent on the number of alternative trajectories with high posteriors* in relation to the number of tracked trajectories k. In particular, if the posterior distribution over the trajectories is flat, k-best enumeration will not consider the right trajectory, and the RBPF performs better. This observation motivates the development of our algorithm that combines k-best enumeration and RBPF in a greedy and stochastic search.

[1]It is insufficient to consider the fault transition several time steps later, since at that time, continuous state tracking is no longer accurate.

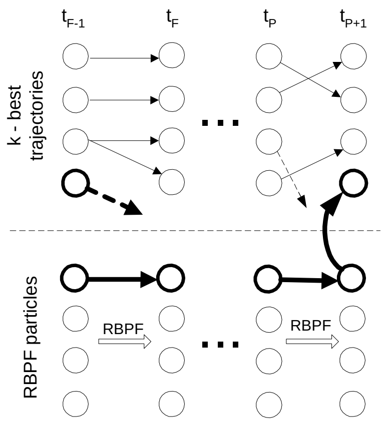

Figure 5: Failure detection with our mixed method. At time t_F the true trajectory (bold line) is no longer among the trajectories with the k highest posteriors. However, the true trajectory is sampled stochastically and becomes a member of the RBPF particle set. At time t_{P+1}, the posterior of the true trajectory becomes large enough for it to be included in the leading k trajectories.

Hybrid Estimation using a Mixed Stochastic/Greedy Method

Our proposed algorithm combines the greedy and stochastic approaches, of k-best enumeration and RBPF respectively, to explore the discrete mode trajectories of the system (Fig. 5). The algorithm maintains two sets of trajectories: one set of stochastically generated trajectories, updated with RBPF, and a separate set of leading trajectories, enumerated according to their posterior probability. The key idea is to generate successors to the leading k trajectories through *both* previously deterministically generated successors to the current k leading trajectories and the candidates generated by the RBPF. In this manner, the true trajectory that was discarded by simple k-best enumeration on the basis of having a lower posterior probability than other trajectories can still be tracked in the RBPF particle set and included in the deterministic set at a later time.

The combination of greedy and stochastic search introduces two challenges. First, in order to enable the comparison of the two sets of trajectories, based on the posterior probability, each particle needs to be augmented with $b(\mathbf{x}_{d,1:t})$, the posterior probability of a mode sequence $\mathbf{x}_{d,1:t}$. This can be updated with Eq. 6, by reusing the values P_T and P_O, computed in the importance sampling step of the RBPF.

Second, trajectories generated by both greedy and stochastic search must be combined to give the belief state maintained by the mixed algorithm. Both techniques maintain a number of trajectories and a Gaussian distribution over the continuous state, conditioned on each trajectory $\mathbf{x}_{d,1:t}$. The belief state representation in the new algorithm is a mixture of Gaussians, obtained by summation over the k trajectories with the highest $b(\mathbf{x}_{d,1:t})$ from both RBPF and greedy successor enumeration.

```
1:  rbpf_candidates ← do_rbpf_update()
2:  priority_queue.push(rbpf_candidates)
3:  priority_queue.push(kbest_trajectories)
4:  while size(new_kbest < k) do
5:     candidate ← priority_queue.pop()
6:     if (is_goal(candidate)) then
7:        candidate ← pop_until_unique(candidate)
8:        new_kbest.push(candidate)
9:     else
10:       new_candidates ← expand_to_successors(candidate)
11:       priority_queue.push(new_candidates)
12:    end if
13: end while
14: do_normalization(new_kbest)
```

Figure 6: Belief state update using greedy and stochastic search.

The pseudocode for the resulting algorithm is shown in Fig. 6. Lines 4 through 13 of the algorithm implement an A* search for the k successors with the highest posterior. This is similar to the method for k-best enumeration described by (Hofbaur & Williams 2002a). The two key additions are as follows:

1. *Addition of RBPF particles to search queue:* In line 1, the RBPF update step is carried out as described by (Funiak & Williams 2003), and the resulting particles are added to the search queue as candidates in line 2. To ensure soundness of the A* search, candidates are added with the unnormalized observation function used by (Maybeck & Stevens 1991).
2. *Checking for uniqueness:* Many identical candidates are generated by RBPF and added to the search queue. The set new_kbest, however, holds *unique* trajectories. Line 7 ensures that unique trajectories only are added to the set of k best.

In the code shown, line 7 ensures uniqueness by removing duplicate candidates from the priority queue until a unique candidate is found. This relies on the fact that identical candidates will be neighbors in the priority queue. Alternatively, candidates can be checked for uniqueness when they are pushed onto the queue.

Experimental Results

We carried out experimental analysis with a range of estimation scenarios; shown here are two examples that illustrate the key results. We used the following two metrics:

1. The fraction of diagnostic faults, defined as $\frac{\#\ wrong\ diagnoses}{\#\ time\ steps}$. Wrong diagnoses are defined as estimates of the discrete state at the fringe that do not correspond to the same discrete state as the ground truth.
2. The mean square estimation error (MSE) of the continuous estimate corresponding to the MAP diagnosis. This is defined as $((\hat{\mathbf{x}}_{c,t} - \mathbf{x}_{c,t})^T(\hat{\mathbf{x}}_{c,t} - \mathbf{x}_{c,t}))^{1/2}$, where $\hat{\mathbf{x}}_{c,t}$ is the continuous estimate corresponding to the MAP diagnosis, and $\mathbf{x}_{c,t}$ is the continuous state ground truth. This measure is averaged over all time steps and experiments.

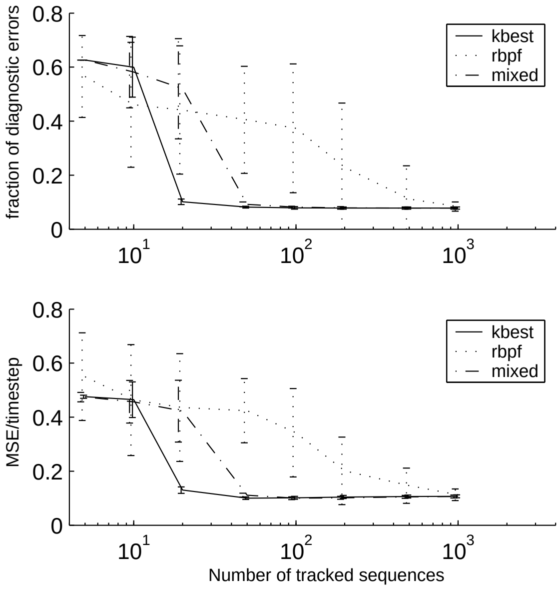

Figure 7: Performance for the actuator failure scenario with concentrated posterior.

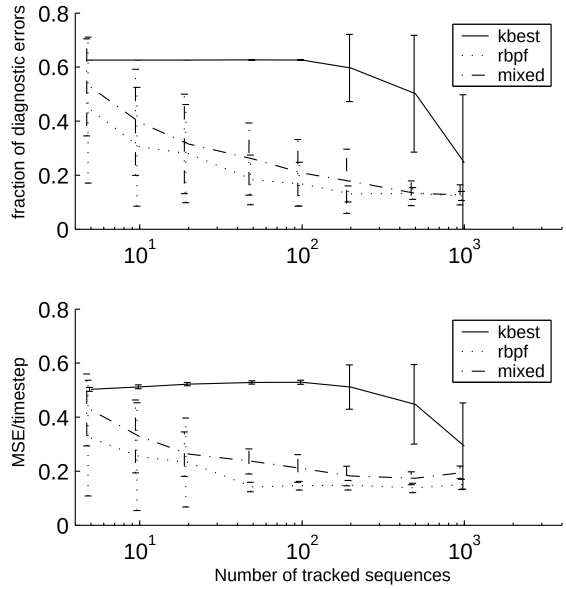

Figure 8: Performance for the failure scenario with the flat posterior.

Each algorithm was run on 20 random observation sequences with fixed mode assignments. The results given here show the mean and standard deviation (shown as error bars) of the performance metrics for these runs. For the mixed method, half of the trajectories were evolved using the RBPF.

Concentrated Posterior

First, we examined the original acrobatic robot model, shown in Fig. 1. In this model, the probability of a ball transition is 0.01 and the probability of an actuator failure is 0.0005. The mass of the ball is 4kg. Since all transitions have low priors, the true posterior distribution is concentrated in a relatively small number of discrete mode trajectories.

As shown in Fig. 7, the k-best algorithm clearly outperforms the RBPF in both performance metrics, except for very small and very large k. There is also a qualitative difference in the convergence of each algorithm with increasing k. Whereas the performance of the RBPF improves gradually as k increases, the k-best algorithm shows a large shift in performance between $k = 10$ and $k = 20$. The mixed method follows the behavior of the k-best algorithm, with twice as many particles (since half of the trajectories are evolved stochastically).

Flat Posterior

Second, we examined a modified model, in which the posterior distribution was spread out among many distinct trajectories. Specifically, we increased the ball transition probabilities from 0.01 to 0.5 and increased the probability of a failure to 0.01. In addition, the mass of the ball was reduced to 1kg. Hence, whenever $\theta_1 \geq 0.55$, many trajectories with relatively high priors exist, and because the effect of a transition on observations is initially small, a large number of distinct trajectories will have high posterior probability.

Fig. 8 shows that in this case, the RBPF clearly outperforms the k-best method, both in terms of diagnostic errors and mean square estimation error. The large number of sequences with high posterior prevents the strict enumeration of the k-best method from considering the initially less likely failure sequence, except with very large k. The RBPF, by contrast, samples the actuator failure transition fairly. If a particle samples the transition close to where it occurred in reality, the observation likelihood of that particle grows until it dominates the trajectory space, giving the correct diagnosis. The behavior of the mixed method is similar to that of the RBPF.

Discussion of Results

In the experimental results an interesting pattern emerges: the k-best algorithm undergoes a phase shift in performance, depending on whether or not k is large enough for trajectories similar to the ground truth to be tracked. The critical value of k depends on the concentration of the posterior distribution. If the posterior distribution is concentrated in a very small number of distinct sequences, the k-best method will perform well even for small k. For a flat posterior, the critical value of k is so large when detecting rare events that many trajectories will need to be tracked, to reliably detect the fault. In these cases, the RBPF will perform better for small k, since the distribution is sampled fairly. The mixed method combines the benefits of the two methods. While its performance is marginally worse than the RBPF or k-best in their best-case scenario, it is much more robust to the choice of model parameters than the RBPF and k-best individually.

Conclusion

This paper presented a new algorithm for approximate estimation in probabilistic hybrid models. Our algorithm combines greedy and stochastic search by tracking two sets of trajectories with k-best and Rao-Blackwellised particle filters, and uses both sets to generate the set of leading trajectories at the next time step. The algorithm is more robust than k-best and RBPF taken individually, with only a minor performance penalty.

References

Doucet, A.; de Freitas, J.; Murphy, K.; and Russell, S. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

Funiak, S., and Williams, B. 2003. Multi-modal particle filtering for hybrid systems with autonomous mode transitions. In *Proceedings of SafeProcess*.

Hofbaur, M., and Williams, B. C. 2002a. Mode estimation of probabilistic hybrid systems. In *Intl. Conf. on Hybrid Systems: Computation and Control*.

Hofbaur, M. W., and Williams, B. C. 2002b. Hybrid diagnosis with unknown behavioral modes. In *Proc. 13th Int. Workshop on Principles of Diagnosis*.

Hutter, F., and Dearden, R. 2003. The Gaussian particle filter for diagnosis of non-linear systems. In *Proc. 14th Int. Conf. on Principles of Diagnosis*, 65–70.

Lerner, U., and Parr, R. 2001. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence*, 310–318.

Lerner, U.; Parr, R.; Koller, D.; and Biswas, G. 2000. Bayesian fault detection and diagnosis in dynamic systems. In *Proc. of the 17th National Conference on AI*.

Lerner, U. 2002. *Hybrid Bayesian Networks for Reasoning About Complex Systems*. Ph.D. Dissertation, Stanford University.

Maybeck, P., and Stevens, R. 1991. Reconfigurable flight control via multiple model adaptive control methods. *IEEE Transactions on Aerospace and Electronic Systems* 27(3):470–480.

Morales-Menendez, R.; de Freitas, N.; and Poole, D. 2002. Real-time monitoring of complex industrial processes with particle filters. In *Proceedings of Neural Information Processing Systems (NIPS)*.