Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-032 April 28, 2006

A Combined Stochastic and Greedy Hybrid Estimation Capability for Concurrent Hybrid Models with Autonomous Mode Transitions

Lars Blackmore, Stanislav Funiak, and Brian Williams

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

A Combined Stochastic and Greedy Hybrid Estimation Capability for Concurrent Hybrid Models with Autonomous Mode Transitions

Lars Blackmore LARSB@MIT.EDU
Massachusetts Institute of Technology
77 Mass. Ave, Cambridge, MA 02139

Stanislav Funiak SFUNIAK@CS.CMU.EDU
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213

Brian Williams WILLIAMS@MIT.EDU
Massachusetts Institute of Technology
77 Mass. Ave, Cambridge, MA 02139

Abstract

Robotic and embedded systems have become increasingly pervasive in applications ranging from space probes and life support systems to robot assistants. In order to act robustly in the physical world, robotic systems must be able to detect changes in operational mode, such as faults, whose symptoms manifest themselves only in the continuous state. In such systems, the state is observed indirectly, and must therefore be estimated in a robust, memory-efficient manner from noisy observations.

Probabilistic hybrid discrete/continuous models, such as Concurrent Probabilistic Hybrid Automata (CPHA) are convenient modeling tools for such systems. In CPHA, the hidden state is represented with discrete and continuous state variables that evolve probabilistically. In this paper, we present a novel method for estimating the hybrid state of CPHA that achieves robustness by balancing greedy and stochastic search. The key insight is that stochastic and greedy search methods, taken together, are often particularly effective in practice.

To accomplish this, we first develop an efficient stochastic sampling approach for CPHA based on Rao-Blackwellised Particle Filtering. We then propose a strategy for mixing stochastic and greedy search. The resulting method is able to handle three particularly challenging aspects of real-world systems, namely that they 1) exhibit autonomous mode transitions, 2) consist of a large collection of concurrently operating components, and 3) are non-linear. Autonomous mode transitions, that is, discrete transitions that depend on the continuous state, are particularly challenging to address, since they couple the discrete and continuous state evolution tightly. In this paper we extend the class of autonomous mode transitions that can be handled to arbitrary piecewise polynomial transition distributions.

We perform an empirical comparison of the greedy and stochastic approaches to hybrid estimation, and then demonstrate the robustness of the mixed method incorporated with our HME (Hybrid Mode Estimation) capability. We show that this robustness comes at only a small performance penalty.

1. Introduction

Robotic and embedded systems have become increasingly pervasive in a variety of applications. Space missions, such as Mars Science Laboratory (MSL, 2005) and the Jupiter Icy Moons Orbiter (JIMO, 2005), have increasingly ambitious science goals, such as operating for longer periods of time and with increasing levels of onboard autonomy. Manned missions in space and in polar environments will rely on life support systems, such as the Advanced Life Support System developed at the NASA Johnson Space Center(Hanford, 2002), to provide a renewable supply of oxygen, water, and food. Here on Earth, robotic assistants, such as CMU's Pearl (Montemerlo, Pineau, Roy, Thrun, & Verma, 2002) and iRobot's Roomba (iRobot, 2005), directly benefit people in ways ranging from providing health care to routine services and rescue operations.

In order to act robustly in the physical world, robotic systems must handle the uncertainty and partial observability inherent in most real-world situations. Robotic systems often face unpredictable, harsh physical environments and must continue performing their tasks (perhaps at a reduced rate), even when some of their subsystems fail. For example, in land rover missions, such as MSL, the robot needs to detect when one or more of its wheel motors fail, which could jeopardize the safety of the mission. The rover can detect the failure from a drift in its trajectory and then compensate for the failure, either by adjusting the torque to its other wheels or by replanning its path to the desired goal.

In our previous work we have developed methods for estimating the state of systems that evolve in a discrete manner, where system behavior is modeled by Concurrent Probabilistic Constraint Automata (CPCA), one automaton per component. The Livingstone model-based diagnosis system(Williams & Nayak, 1996) flew on the Deep Space One probe, which had approximately 4^{80} modes of normal and faulty operation. Monitoring and diagnosis was performed in real-time, by approximating the belief state through enumeration of the k most likely mode trajectories at each step. Best-first enumeration of the concurrent automata mode transitions was enabled by exploiting conditional independence between transitions. Recent advances in model-based diagnosis (Williams, Chung, & Gupta, 2001)(Sachenbacher & Williams, 2004)(Martin, Williams, & Ingham, 2005)(Mikaelian, Williams, & Sachenbacher, 2005) have further increased the capabilities of estimation for discrete models through model compilation, structural decomposition and symbolic encoding methods that accelerate search, and richer encodings for mixed software/hardware systems.

In many situations, however, a purely discrete model is insufficient. Probabilistic hybrid models represent the system with both discrete and continuous state variables that evolve probabilistically according to a known distribution. The discrete state variables typically represent a *behavioral mode* of the system, while the continuous variables represent its *continuous dynamics*. Probabilistic hybrid models can be used to provide an appropriate level of modeling abstraction when purely discrete, qualitative models are too coarse, while purely continuous, quantitative models are too fine-grained.

In this paper, we investigate the problem of estimating the state of systems with probabilistic hybrid models. Given a sequence of control inputs and noisy observations, our goal is to estimate the discrete and continuous state of the hybrid model. Probabilistic hybrid models are particularly useful for fault diagnosis, the problem of determining the health state of a system. With hybrid models, fault diagnosis can be framed as a state estimation

problem, by representing the nominal and fault modes with discrete variables and the state of the system dynamics with continuous variables.

Our previous work (Hofbaur & Williams, 2002a) developed the HME (Hybrid Mode Estimation) system, extending our work on discrete model-based diagnosis for Concurrent Probabilistic Constraint Automata, to reason about probabilistic hybrid models, known as Concurrent Probabilistic Hybrid Automata (Hofbaur & Williams, 2002a). As with CPCA, CPHA represent the system as a collection of concurrently operating automata, one automaton for each component in the system. Each mode in an automaton has an associated set of stochastic difference and algebraic equations, which are solved to obtain a complete dynamical model of the system. An alternative representation, developed in parallel by the hybrid Bayes Network community, is the hybrid Dynamic Bayes Network (DBN), where the system is represented as a directed acyclic graph, in which vertices represent the random variables in the system and edges capture the conditional dependencies among them.

We present here a novel method for hybrid estimation with CPHA that achieves robustness by balancing greedy and stochastic search. 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) (Gomes, Selman, & Kautz, 1998) and reinforcement learning (Sutton & Barto, 1998), for example. Previous methods have used either greedy search (for example, Hofbaur & Williams, 2004) or stochastic sampling (for example, Verma, Langford, & Simmons, 2001). In this paper we show empirically that these methods can have limited performance depending on whether the belief state is concentrated in a few mode sequences, or is relatively flat. The mixed greedy/stochastic method, by contrast, is robust to changes in the variance of the posterior distribution.

Developing this method requires three main technical contributions. First, we introduce an efficient stochastic sampling approach for CPHA based on Rao-Blackwellised Particle Filtering. Second, we perform an empirical study of the greedy and stochastic CPHA methods on a simulated acrobatic robot example. Third, based on the comparative insights gained we propose a mixed exploration/exploitation strategy, and demonstrate its superiority over the separate approaches.

We first consider existing methods for estimation with hybrid models. In the general case, inference in hybrid models is NP-Hard(Lerner & Parr, 2001). Approximate inference using techniques in k-best Gaussian filtering (Hofbaur & Williams, 2004; Lerner, 2002a) have been effective for hybrid state estimation. These methods represent the system state as a mixture of Gaussians that are enumerated in decreasing order of likelihood. Using this approach, our prior work developed a method that can be applied to models that have autonomous mode transitions, that is, transitions that depend on the continuous state variables, nonlinear dynamics, and many components (Hofbaur & Williams, 2004). Owing to their efficient representation and focused search, the k-best method has been successfully applied to large systems with as many as 450,000 discrete states. Excessive focusing during search may, however, lead to diagnostic errors if the correct diagnosis is not among the leading set of hypothese. In this paper, we demonstrate this empirically using a simulated acrobatic robot.

An alternative approach is to use stochastic sampling to allow the system to perform greater exploration, rather than performing a purely greedy search. An example of such as method is Particle Filtering, which approximates the posterior state distribution using a finite number of particles(Doucet, 1998)(Morales-Menéndez, de Freitas, & Poole, 2002). Examples of diagnosis systems include (Verma et al., 2001) and (Dearden & Clancy, 2002). These particles are evolved stochastically and resampled based on an appropriately defined importance weighting. The inefficiency of sampling in high-dimensional spaces has limited the effectiveness of Particle Filtering for state estimation in hybrid systems(Doucet, de Freitas, Murphy, & Russell, 2000). To avoid these problems, we therefore propose an efficient stochastic sampling approach for CPHA based on Rao-Blackwellised Particle Filtering. Rao-Blackwellised Particle Filtering exploits a tractable substructure in the underlying model by sampling only a subspace of the system state, while estimating the remainder using an efficient analytical method.

Rao-Blackwellised Particle Filtering has been used for hybrid state estimation in Switching Linear Dynamic Systems(Freitas, 2002)(Morales-Menéndez et al., 2002), in which the discrete state d is a Markov chain with a known transition probability $p(d_t|d_{t-1})$, and the continuous state evolves linearly, with system and observation matrices dependent on d_t. Under such conditions, the continuous estimate for each sequence of discrete state assignments can be computed with a Kalman Filter.

In many domains, however, such as rocket propulsion systems (Koutsoukos, Kurien, & Zhao, 2002) or life-support systems (Hofbaur & Williams, 2002a), simple Markovian transitions $p(d_t|d_{t-1})$ are not sufficiently expressive. In these domains, the transitions of the discrete variables often also need to depend on the continuous state. For example, the probability of a failure transition for a certain component may depend on its current operating temperature, which is part of the continuous state. Such transitions are called *autonomous*. In the terminology of hybrid Bayesian networks, these correspond to discrete nodes with continuous parents. Furthermore, many systems consist of several interconnected components, each of which is in its own behavioral mode. Representing the joint mode of all the components would be inefficient. In these cases, it is desirable to represent the mode with several mode variables. Finally, many systems have nonlinear dynamics. Each of these properties - autonomous transitions, interconnected components, and nonlinear dynamics, are expressed naturally in CPHA.

In this paper, we first extend Rao-Blackwellised Particle Filtering to handle autonomous mode transitions, nonlinearities and concurrency. Applying Rao-Blackwellisation schemes to models with autonomous transitions is difficult, since the discrete and continuous state spaces of these models tend to be coupled. The key innovation in our algorithm is that it reuses the continuous state estimates in the importance sampling step of the particle filter. We extend the class of autonomous transitions that can be handled over our previous work in (Funiak & Williams, 2003) and (Hofbaur & Williams, 2002a) to multivariable linear transition guards, in the case of piecewise constant transition distributions, and to polynomial transition distributions of arbitrary order, in the case of single variables. Our RBPF algorithm handles nonlinear dynamics by using an Extended Kalman Filter (Anderson & Moore, 1979) or an Unscented Kalman Filter (Julier & Uhlmann, 1997). We presented this innovation in (Funiak & Williams, 2003), this was also proposed independently by (Hutter & Dearden, 2003).

Next, these developments provide the foundation for introducing a a unified treatment of k-best enumeration and RBPF approaches to hybrid state estimation. Both these approaches represent the belief state by a mixture of Gaussians for a subset of mode 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 trade-offs 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.

Finally, in this paper we develop such an algorithm. The new algorithm uses Rao-Blackwellised particle filtering to generate, stochastically, additional candidates to add to 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 k-best successor enumeration. This algorithm makes use of the efficient properties of both k-best enumeration and RBPF, while being probabilistically sound.

We integrate the mixed algorithm with our HME system and demonstrate, using a simulated acrobatic robot, that the mixed algorithm is effective for both a concentrated and flat posterior distribution. The mixed algorithm shows a dramatic increase in robustness for a small performance penalty.

2. Models and State Estimation Methods

Probabilistic hybrid models and hybrid estimation methods date back to the 1970s (Akashi & Kumamoto, 1977) and are useful in many applications, including visual tracking (Pavlovic, Rehg, Cham, & Murphy, 1999) and fault diagnosis (Narasimhan, Biswas, Karasai, & Zhao, 2000)(Hofbaur & Williams, 2002a). In this section, we review a formalism for modeling probabilistic hybrid systems, known as Concurrent Probabilistic Hybrid Automata. We then define the hybrid state estimation problem and outline the existing approach to hybrid state estimation for CPHA, based on greedy enumeration. This lays the groundwork for the new approach, based on Rao-Blackwellised Particle Filtering, which is described in Section 3.

2.1 Concurrent Probabilistic Hybrid Automata

We have previously developed Concurrent Probabilistic Hybrid Automata (CPHA) (Hofbaur & Williams, 2004), a formalism for modeling engineered systems that consist of a large number of concurrently operating components with non-linear dynamics.

A CPHA model consists of a network of concurrently operating Probabilistic Hybrid Automata (PHA), connected through shared continuous input/output variables. Each PHA represents one component in the system and has both discrete and continuous hidden state variables. The automaton interacts with the other automata in the surrounding world

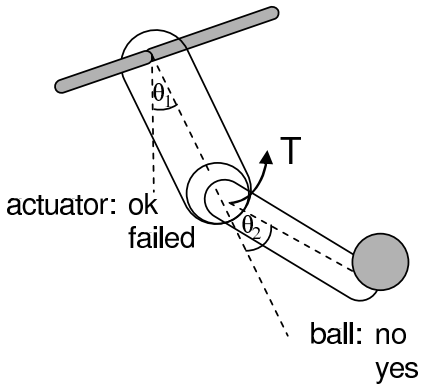


Figure 1: A two-link acrobatic robot.

through shared continuous variables, and its discrete state determines the evolution of its continuous variables.

Definition 1 *A Probabilistic Hybrid Automaton is a tuple* $\langle \mathbf{x}, \mathbf{w}, F, T, X_0, \mathcal{X}_d, \mathcal{U}_d \rangle$ *(Hofbaur & Williams, 2002a):*

- $\mathbf{x}$ denotes the *hybrid state* of the automaton. $\mathbf{x} \triangleq \mathbf{x}_d \cup \mathbf{x}_c$ where $\mathbf{x}_d$ denotes the discrete state variables $\mathbf{x}_d \in \mathcal{X}_d$ and $\mathbf{x}_c$ denotes the continuous state variables $\mathbf{x}_c \in \mathbb{R}^{n_x}$.[1]
- $\mathbf{w}$ denotes the set of *input/output variables*, which consists of *command* (discrete input) variables $\mathbf{u}_d \in \mathcal{U}_d$, continuous input/output variables $w_c \in \mathbb{R}^{n_w}$, and Gaussian noise variables $\mathbf{v}_c \in \mathbb{R}^{n_v}$.
- $F : \mathcal{X}_d \rightarrow F_{DE} \cup F_{AE}$ specifies the *continuous evolution* of the automaton for each discrete mode, in terms of the set of discrete-time difference equations F_{DE} and the set of algebraic equations F_{AE} over the variables $\mathbf{x}_c$, $\mathbf{w}_c$, and $\mathbf{v}_c$.
- $T : \mathcal{X}_d \rightarrow 2^{\mathcal{P} \cup \mathcal{C}}$ specifies the discrete transition distribution of the automaton as a finite set of transition probabilities $p_{\tau i} \in \mathcal{P}$ over the target modes $\mathcal{X}_d$ and their associated guard conditions $c_i \in \mathcal{C}$ over $\mathbf{x}_c \cup \mathbf{u}_d$. The guard conditions c_i form a partition of the space $\mathbb{R}^{n_x} \times \mathcal{U}_d$.
- X_0 is a distribution for the initial state of the automaton, with a Gaussian distribution $p(\mathbf{x}_{c,0} | \mathbf{x}_{d,0})$ for the continuous state $\mathbf{x}_{c,0}$, conditioned on each discrete mode $\mathbf{x}_{d,0}$.

To illustrate this definition, consider the two-link acrobot, shown in Figure 1. The robot swings on a bar and may catch a ball of known mass, whenever it is on the right side ($\theta_1 > 0.55$). The body of the robot can be modeled as a PHA, where the hidden continuous state consists of four variables, representing the angles and angular velocities at two joints. The hidden discrete state, as shown in Figure 2, consists of the variable `ball`, representing whether or not the robot carries a ball of mass m_{ball}. The continuous dynamics for each

1. We let lowercase bold symbols, such as $\mathbf{v}$, denote both the *set* of variables $\{v_1, \ldots, v_l\}$ and the *vector* $[v_1, \ldots, v_l]^T$.

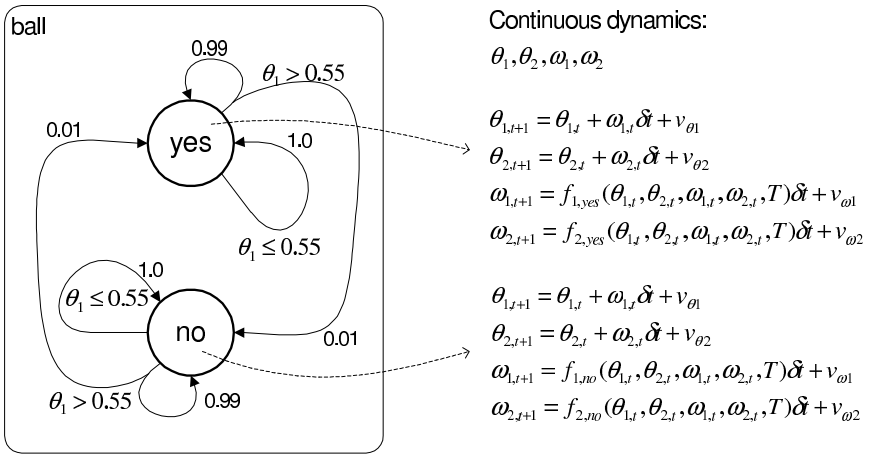


Figure 2: A PHA for the two-link body of the acrobot system. Left: transition model for the discrete state of the body. Right: evolution of the automaton's continuous state, one set of equations for each mode.

mode can be derived using Lagrangian mechanics(see Paul, 1982) and turned into a set of discrete-time difference equations, using the Euler approximation.

The transition function $T(\mathbf{d})$, for some mode $\mathbf{d}$, specifies the transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1}, \mathbf{u}_{d,t})$. Each tuple $\langle p_\tau, c\rangle \in T(\mathbf{d})$ defines the transition distribution as having a constant value of p_τ in the regions satisfied by the guard c. For the acrobot example, when `ball` = `no`, the probability of transitioning to mode `yes` is 0.01, whenever $\theta_1 > 0.55$, and 0 otherwise. The transition function can thus specify conditional distributions $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1}, \mathbf{u}_{d,t})$ that are piecewise constant in the continuous state $\mathbf{x}_{c,t-1}$. While in keeping with previous workwe retain this definition for PHA, in Section 4.2 we show that our efficient hybrid estimation method can handle piecewise *polynomial* transition functions.

Most engineered systems consist of several concurrently operating components. Composition of PHA provides a method for specifying a model for the overall system, by specifying PHA models for its components and then combining these models. Composed automata are connected through shared continuous input/output variables, which corresponds to connecting the system's physical components through natural phenomena, such as forces, pressures, and flows.

In order to compose PHA, we combine their hidden state variables and their discrete and continuous evolution functions:

Definition 2 *The composition $\mathcal{CA}$ of two Probabilistic Hybrid Automata $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as a tuple $\langle \mathbf{x}, \mathbf{w}, F, T, X_0, \mathcal{X}_d, \mathcal{U}_d \rangle$, where*

$$\begin{aligned}
\mathbf{x} &= \mathbf{x}_d \cup \mathbf{x}_c, \text{ with } \mathbf{x}_d \triangleq \mathbf{x}_{d1} \cup \mathbf{x}_{d2} \text{ and } \mathbf{x}_c \triangleq \mathbf{x}_{c1} \cup \mathbf{x}_{c2}, \\
\mathbf{w} &\triangleq \mathbf{w}_1 \cup \mathbf{w}_2, \\
F(\mathbf{x}_d) &\triangleq F_1(\mathbf{x}_{d1}) \cup F_2(\mathbf{x}_{d2}), \\
T(\mathbf{x}_d) &\triangleq T_1(\mathbf{x}_{d1}) \times T_2(\mathbf{x}_{d2}), \\
X_0(\mathbf{x}) &= X_{01}(\mathbf{x}_2) X_{02}(\mathbf{x}_2), \\
\mathcal{X}_d &\triangleq \mathcal{X}_{d1} \times \mathcal{X}_{d2}, \text{ and} \\
\mathcal{U}_d &\triangleq \mathcal{U}_{d1} \times \mathcal{U}_{d2}.
\end{aligned}$$

We define the transitions of each component as being independent, conditioned on $\mathbf{x}_{c,t-1}$. The overall continuous evolution of the CPHA is determined by taking the union of algebraic and difference equations for each component PHA. The equations $F(\mathbf{d})$ are then solved using Groebner bases (Buchberger & Winkler, 1998) and causal analysis (Nayak, 1995) into the standard form

$$\begin{aligned}
\mathbf{x}_{c,t} &= \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{c,t-1}, \mathbf{v}_{x,t-1}; \mathbf{x}_{d,t}) \\
\mathbf{y}_{c,t} &= \mathbf{g}(\mathbf{y}_{c,t-1}, \mathbf{u}_{c,t-1}, \mathbf{v}_{y,t-1}; \mathbf{x}_{d,t}).
\end{aligned} \tag{1}$$

The composition of a set of PHA is well-defined, provided that they satisfy *compatibility* conditions; see (Hofbaur, 2003) for an initial development of this topic. In particular, PHA are closed under composition; composing two compatible automata results in a valid PHA. Due to historical reasons, we use the term *Concurrent Probabilistic Hybrid Automaton* (CPHA) to refer to a composed model and reserve the term PHA for a single-automaton model.

Using composition, the PHA for an acrobot body can be augmented using a PHA model of the torque actuator, and a model of the angular position sensor, measuring θ_2. A schematic of the CPHA is shown in Figure 3, and the full discrete transition model is shown in Figure 4. The actuator exerts the commanded torque in the `ok` mode, but exerts zero torque in the `failed` mode. The position sensor is modeled as adding Gaussian white noise to the true value of θ_2.

2.2 Hybrid State Estimation

Given a hybrid model of the system, our goal is to estimate its state from a sequence of control inputs and observations:

Definition 3 Hybrid State Estimation *Given a CPHA model of the system $\mathcal{CA}$ and the sequence of control inputs $\mathbf{u}_0, \ldots, \mathbf{u}_t$ and observed outputs $\mathbf{y}_0, \ldots, \mathbf{y}_t$, at time t determine the conditional probability $\langle \mathbf{x}_{d,t}, \mathbf{x}_{c,t} \rangle = p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$.*

The hybrid estimate of $\langle \mathbf{x}_{d,t}, \mathbf{x}_{c,t} \rangle$ can take on several forms, depending on the task addressed. In fault diagnosis, one is typically concerned about the most likely mode (MAP mode estimate) of the system or the distribution over the set of possible modes. In tracking,

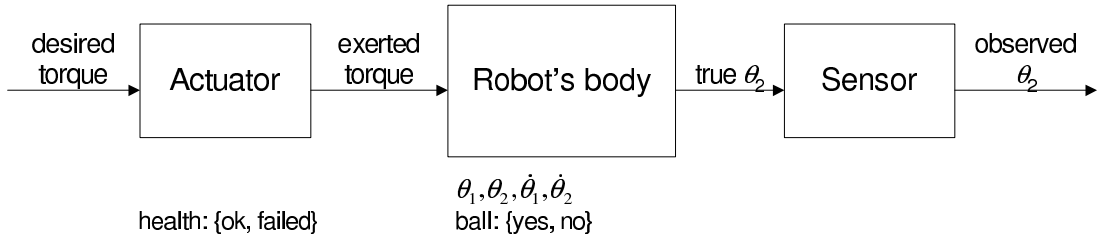


Figure 3: A composed model for the acrobatic robot in Figure 1. Each component is modeled with one Probabilistic Hybrid Automaton. The component automata are shown in rectangles, with their state variables shown beneath.

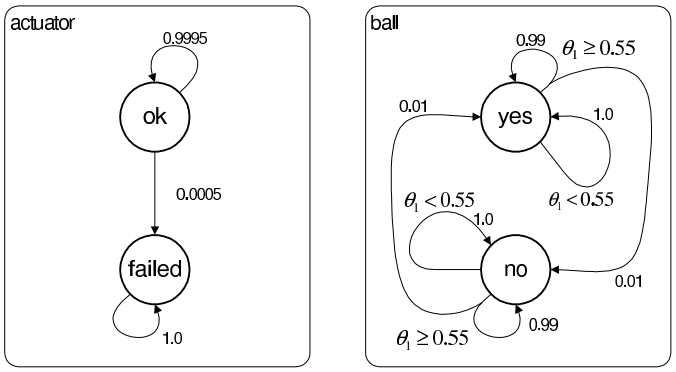


Figure 4: Discrete transition model for the acrobot CPHA. If the actuator has failed, it exerts no torque. When the robot catches the ball, `ball=yes`, and the mass of the lower link increases.

on the other hand, the primary goal is to filter out the continuous state of the system. In general, we can frame the hybrid state estimation problem as that of approximating the posterior distribution over $\langle \mathbf{x}_{d,t}, \mathbf{x}_{c,t} \rangle$ and use this distribution to compute the derived characteristics, such as the MAP estimate.

2.3 K-best Enumeration

Existing methods for hybrid estimation with CPHA models have used a k-best enumeration approach(Hofbaur & Williams, 2004). This section outlines the approach.

The desired distribution $p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ can be expressed as a sum of posterior distributions for all discrete mode trajectories that end in state $\mathbf{x}_{d,t}$:

$$p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = \sum_{\mathbf{x}_{d,0:t-1}} p(\mathbf{x}_{d,0:t}, \mathbf{x}_{c,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t}). \tag{2}$$

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:

$$p(\mathbf{x}_{d,0:t}, \mathbf{x}_{c,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = p(\mathbf{x}_{d,0:t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) p(\mathbf{x}_{c,t} | \mathbf{x}_{d,0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}). \tag{3}$$

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, 2002b). The weight of each mixture component is then computed using the belief state update (Funiak & Williams, 2003)

$$p(\mathbf{x}_{d,0:t} | \mathbf{y}_{1:t}) = b(\mathbf{x}_{d,0:t}) \propto P_O \cdot P_T \cdot b(\mathbf{x}_{d,0:t-1}). \tag{4}$$

In this equation, $P_T \triangleq p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ is the prior probability of transitioning to a state $\mathbf{x}_{d,t}$, given the past mode trajectory and past observations; we refer to this as the *transition prior*. $P_O \triangleq p(\mathbf{y}_t | \mathbf{x}_{d,0:t}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ is the measurement update. Both P_T and P_O can be calculated approximately(Hofbaur & Williams, 2004).

Naturally, tracking all possible mode sequences is infeasible; the number of such sequences increases exponentially with time. Indeed, inference in probabilistic hybrid models, including SLDS, hybrid Bayesian networks, and CPHA, has been shown to be NP-hard (Lerner & Parr, 2001). Nevertheless, in many domains, efficient inference is possible by employing two strategies: *pruning* (branching) and *collapsing* (merging) (see Figure 5). Pruning removes some branches from the belief state, based on the evidence observed so far, while collapsing combines sequences with the same mode at their fringe to a single hypothesis (Blom & Bar-Shalom, 1988; Lerner, Parr, Koller, & Biswas, 2000).

For mode estimation in purely discrete systems, k-best filtering methods have been demonstrated to great effect(Williams & Nayak, 1996). K-best filtering methods focus the state estimation on sequences with high posterior probability. Typically, a k-best filter starts with a set of mode sequences at one time step and expands these sequences to obtain the set of leading sequences at the next time step.

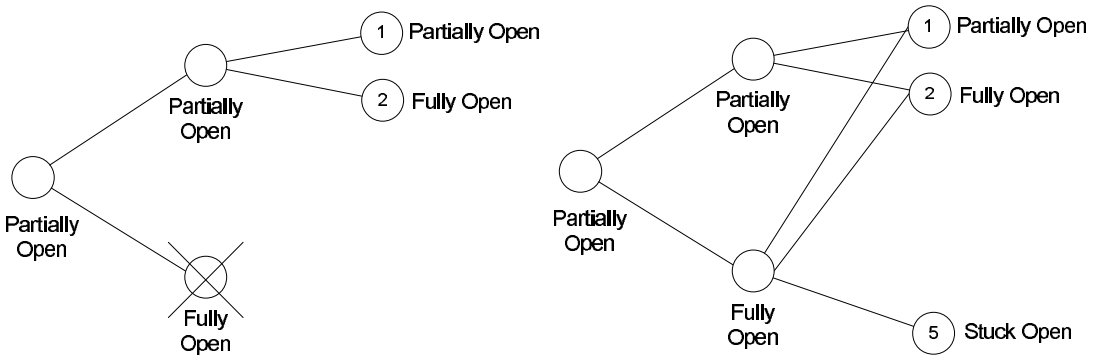


Figure 5: Pruning (left) and collapsing strategies (right).

K-best enumeration has also been shown to be effective for hybrid estimation. One approach is to expand all the successors of all sequences and compute their transition prior and observation likelihoods (Lerner et al., 2000). With additional independence assumptions on the model, such as those in CPHA, an efficient solution is to frame the expansion as a search and solve it using a combination of branch and bound and A* algorithms (Hofbaur & Williams, 2004). The pseudocode for this algorithm applied to CPHA is shown in Figure 7 and Figure 8. At each time-step t, k distinct mode trajectories are tracked. The algorithm finds the k most likely mode trajectories at time $t+1$, by performing a tree search. An example of this tree search is shown in Figure 6 for the acrobot in Figure 1. Each node of the tree is a partial assignment to the modes of the PHA components in the CPHA. The leaves of the tree are those nodes for which a full assignment $\mathbf{x}_{d,t+1}$ has been made, and the observation probability $P_O = p(\mathbf{y}_{t+1}|\mathbf{x}_{d,0:t+1}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ has been calculated.

This tree can be searched efficiently using A* to find the k leaf nodes with the greatest posterior probability $p(\mathbf{x}_{d,0:t+1}|\mathbf{y}_{1:t+1}, \mathbf{u}_{0:t+1}) \propto P_O \cdot P_T \cdot b(\mathbf{x}_{d,0:t})$. The transition prior P_T can be separated into the transition priors for individual components, P_{Ti}. Doing this and taking the negative logarithm gives the cost of a leaf node, which is to be minimized:

$$-ln(b(\mathbf{x}_{d,0:t+1}|\mathbf{y}_{1:t+1}, \mathbf{u}_{0:t})) = -ln(b(\mathbf{x}_{d,0:t})) - ln(P_O) - \sum_{i=1}^{n} ln(P_{Ti}). \tag{5}$$

For a node corresponding to an assignment to j of the n components, the cost g is defined as:

$$g = -ln(b(\mathbf{x}_{d,0:t})) - \sum_{l=1}^{j} ln(P_{Tl}). \tag{6}$$

The cost to go can then be lower-bounded using the admissible heuristic function h:

$$h = -\sum_{l=j+1}^{n} ln(\max p_{\tau l}). \tag{7}$$

Here, $\max p_{\tau j}$ is the maximum value of $p(x_{d,t+1}|x_{d,t}, \mathbf{x}_{c,t-1}, \mathbf{u}_{d,t})$ for component j over all possible guard conditions. Since in the CPHA model this has a finite number of possible values, computing this is straightforward.

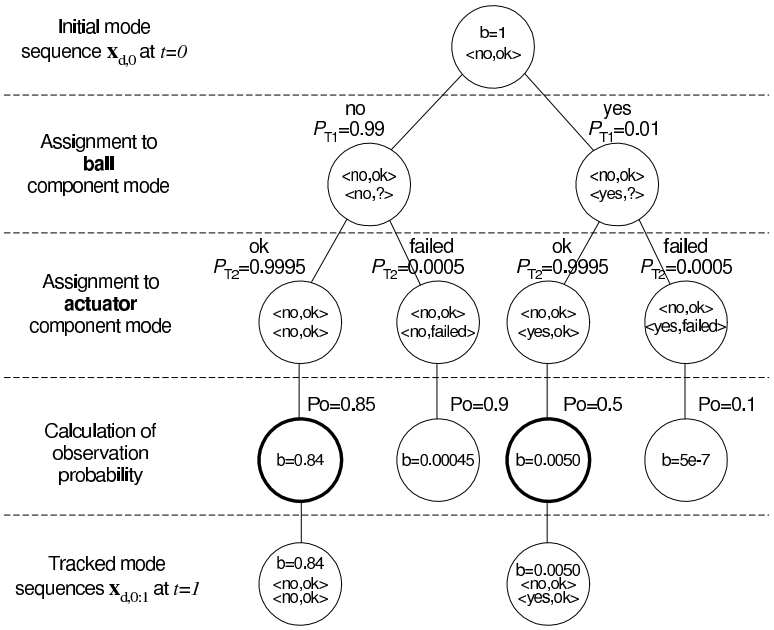


Figure 6: K-best successor enumeration for acrobot CPHA, framed as a tree search. The root node is the initial mode at time $t = 0$. Each node in the tree corresponds to a partial assignment to mode variables at time $t = 1$. With $k = 2$, A* search finds the two leaf nodes with the greatest $b(\mathbf{x}_{d,0:1})$, which correspond to the mode trajectories up to $t = 1$ with the greatest posterior probability.

1. Initialization
   - create a node s corresponding to each non-zero value of $p(\mathbf{x}_{d,0})$
   - initialize $f(s) = -ln(p(\mathbf{x}_{d,0}))$
   - initialize the estimate mean $\hat{\mathbf{x}}_{c,0}^{(s)} \leftarrow \mathbb{E}[\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(s)}]$
   - initialize the estimate covariance $\mathbf{P}_0^{(s)} \leftarrow Cov(\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(s)})$
   - add all nodes $\mathbf{x}_{d,0}^{(s)}$ to priority queue
2. For $t = 1, 2, \ldots$
   (a) A* Search Step
   - **While** $size(new_kbest) < k$ **do**
     - Remove node s from priority queue with lowest $f = g + h$: $s \leftarrow pop_from_queue()$
     - **If** s is a leaf node (has full assignment to component modes and P_O computed), then
       * Add s to new_kbest
     - **Else**
       * Expand node s to successors: $expanded_nodes \leftarrow expand_to_successors(s)$
       * Add expanded nodes to priority queue: $push_onto_queue(expanded_nodes)$

   (b) Normalization Step
   - Normalize new k-best nodes: $normalize(new_kbest)$

Figure 7: Hybrid Estimation for CPHA using k-best Enumeration.

The algorithm in Figure 7 takes nodes from the queue starting with the lowest $f = g+h$. The function in Figure 8 expands the node to its successors, which updates the cost g using the transition function for the next component. If all of the components have mode assignments, the cost is updated using the observation function P_O and put back onto the queue. We use the modified observation function used by (Maybeck & Stevens, 1991), which does not contain the pdf normalization term; the final step in the algorithm renormalizes the posterior probabilities.

This search is guaranteed to find the leaf node with the lowest cost since h is an admissible heuristic. The lowest cost node is removed from the tree, and the process repeats until the k best leaf nodes are found. Hence the k-best mode sequences at $t + 1$ successors can be found without enumerating all of the possible mode sequences. The k-best enumeration approach has been shown empirically to be an efficient technique for hybrid state estimation in systems that exhibit autonomous mode transitions, nonlinear dynamics and concurrency (Hofbaur & Williams, 2004).

This completes our review of CPHA and k-best enumeration. In Section 3 we develop a complementary, stochastic method based on Rao-Blackwellised Particle Filtering. By combining these methods, making use of the insight from the empirical comparison in Section 6, we develop a robust, memory-efficient method that balances exploration and exploitation (Section 5).

1. Expand Node s to Successors

- **If** component number $j \leq N$
  - Increment component number j: $j \leftarrow j+1$
  - For each possible transition of component j to mode $\mathbf{x}_{d\,j,t}$
    - Update partial mode assignment with $\mathbf{x}_{d\,j,t}$
    - Compute component transition prior: $P_{Tj} \leftarrow p(\mathbf{x}_{d\,k,t}|\mathbf{x}^{(i)}_{d\,k,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
    - Update node cost: $g(s) \leftarrow g(s) - ln(P_{Tj})$
    - Update heuristic: $h(s) \leftarrow -\sum_{l=j+1}^{n} ln(\max p_{\tau l})$
  - **Return** successor nodes
- **Else**
  - Perform a Kalman Filter update: $\tilde{\mathbf{x}}^{(s)}_{c,t}, \tilde{\mathbf{P}}^{(s)}_{t}, \mathbf{r}^{(s)}_{t}, \mathbf{S}^{(s)}_{t} \leftarrow UKF(\hat{\mathbf{x}}^{(s)}_{c,t-1}, \mathbf{P}^{(s)}_{t-1}, \mathbf{x}^{(s)}_{d,t})$
  - Calculate observation probability: $P_O \leftarrow p(\mathbf{y}_t|\mathbf{x}_{d,1:t}, \mathbf{y}_{1:t-1})$
  - Update node cost: $g(s) \leftarrow g(s) - ln(P_O)$
  - Update heuristic: $h(s) \leftarrow 0$
  - **Return** successor nodes

Figure 8: Node Expansion for k-best Enumeration Algorithm.

3. Rao-Blackwellised Particle Filtering for CPHA

The key contribution of this section is an approximate Rao-Blackwellised particle filtering (RBPF) algorithm for CPHA that handles autonomous mode transitions, that is, those that depend on the continuous state, as well as nonlinear system dynamics and concurrency.

3.1 Overview of Rao-Blackwellised Particle Filtering for CPHA

Our first step in developing a mixed exploitation/exploration strategy for CPHA is to develop a memory-efficient monte carlo filtering approach for CPHA, which complements our greedy k-best approach for CPHA. To achieve memory-efficiency, we use particles to represent Gaussian distributions conditioned on a particular mode sequence, and we develop a Gaussian particle filter for CPHA as an instance of a Rao-Blackwellised Particle Filter(Akashi & Kumamoto, 1977; Morales-Menéndez et al., 2002)

Our algorithm is illustrated in Figure 9. In the spirit of prior approaches to RBPF (Akashi & Kumamoto, 1977; Morales-Menéndez et al., 2002), our algorithm exploits the structure in the estimation problem and samples only the discrete mode sequences. Conditioned on each sampled sequence, the new algorithm approximates the associated continuous state distribution as a Gaussian in closed form, using an Extended (Anderson & Moore, 1979) or an Unscented Kalman Filter (Julier & Uhlmann, 1997). Each particle holds a sample trajectory $\mathbf{x}^{(i)}_{d,0:t}$ and the corresponding continuous estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t}, \mathbf{P}^{(i)}_{t} \rangle$. This is described in detail in Section 3.3.

The algorithm starts by taking a fixed number of random samples from the initial distribution over the mode variables $p(\mathbf{x}_{d,0})$ (Step 1). For each sampled mode $\mathbf{x}^{(i)}_{d,0}$, the corresponding initial continuous distribution $p(\mathbf{x}_{c,0}|\mathbf{x}^{(i)}_{d,0})$ is specified by the PHA model. At each time-step, the algorithm then uses the model to expand the mode sequences of each particle and to update the corresponding continuous estimates (see Figure 9, Step 2). This is

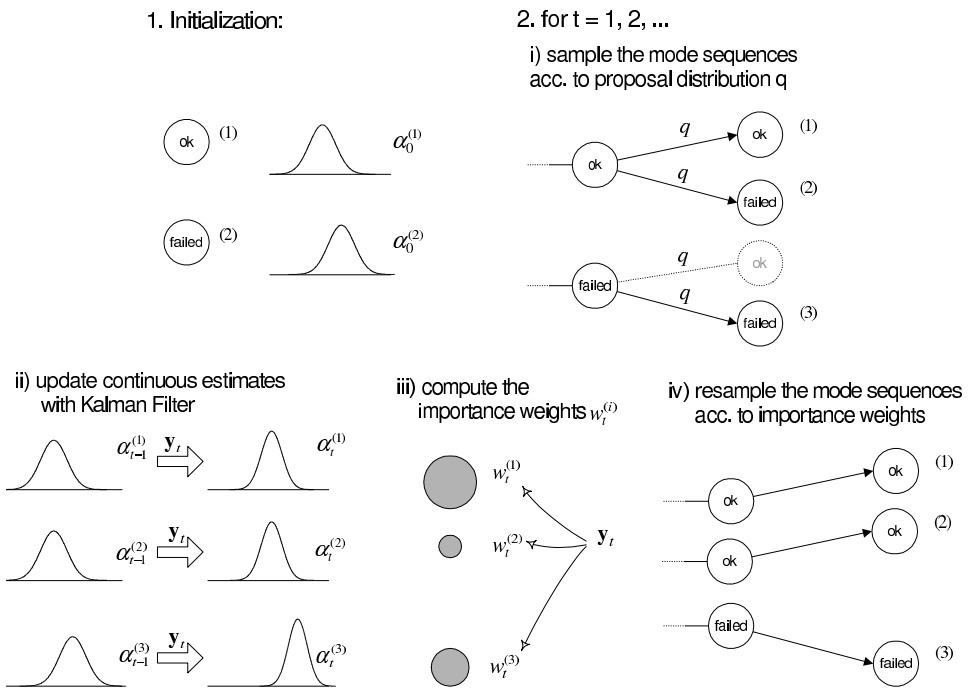


Figure 9: Rao-Blackwellised Particle Filter for PHA.

done by first evolving each particle by taking one random sample $\mathbf{x}_{d,t}^{(i)}$, for each particle, from a suitably chosen proposal distribution $q(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. Intuitively, the proposal is a distribution close to the true distribution that we are trying to determine, $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$; that is, the posterior probability of a given mode sequence given all of the observations up to time t. The posterior distribution is difficult to calculate in closed form, so instead we sample from the proposal distribution, which we choose to be easy to calculate. We then compensate for the discrepancy between the proposal distribution and the true distribution by assigning an importance weight $w_t^{(i)}$ for each new mode sequence $\mathbf{x}_{d,0:t}^{(i)}$. Finally, the resampling step duplicates particles according to their weighting, thereby adjusting the number of particles wherever the proposal distribution does not match the desired distribution(Doucet, 1998).

In order to instantiate this algorithm we must define the proposal distribution and the importance weight. For the proposal distribution we choose the distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, which is the transition prior mentioned in Section 2.3. We show in Section 3.4 that this distribution can be calculated efficiently, even when the model includes autonomous mode transitions.

The importance weights then correct for the discrepancy between the proposal and the posterior distribution by taking into account the latest observation $\mathbf{y}_t$. In Section 3.6 we show that this can be computed using the Kalman Filter innovation.

Finally, in Section 3.7 the algorithm outlined in Figure 9 is extended to deal with Concurrent Probabilistic Hybrid Automata.

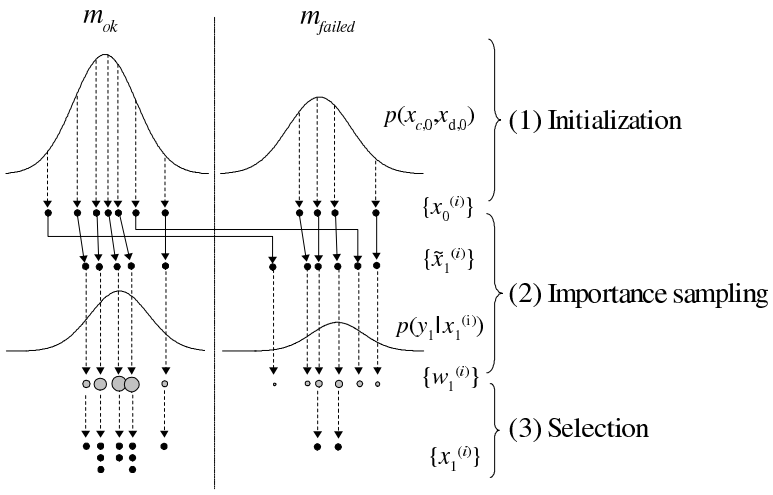


Figure 10: The three steps of a simple particle filter for a PHA model with one discrete and one continuous variable.

3.2 Rao-Blackwellised Particle Filtering

In this section we summarize briefly Particle Filtering for hybrid state estimation, and review Rao-Blackwellised Particle Filtering.

Particle filters approximate the posterior distribution for the hybrid state $\mathbf{x}_t$ with a set of sampled sequences $\{\mathbf{x}_{0:t}^{(i)}\}$. These samples are evolved sequentially and approximate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{0:t}, \mathbf{u}_{0:t})$ as the probability density function

$$p\bar{}_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(x). \tag{8}$$

In the simplest solutions, the samples are taken from the complete hybrid state $\mathcal{X}_d \times \mathbb{R}^{n_x}$, and are evolved in three steps, as illustrated in Figure 10. In the first, initialization step, the algorithm samples the initial distribution $p(\mathbf{x}_0)$; thus, effectively approximating the posterior at $t = 0$. Then, in each iteration, the sequences $\{\mathbf{x}_{d,0:t}^{(i)}, \mathbf{x}_{c,0:t}\}$ are evolved by taking *one* random sample $\tilde{\mathbf{x}}_t^{(i)}$ from an appropriately-chosen *proposal distribution*, and by assigning importance weights that account for the differences between the proposal and the posterior distributions. The final step selects a number of off-spring for each particle according to its weight, thus duplicating the "good" ones and removing the "bad" ones.

In practice, sampling in high-dimensional spaces can be inefficient, since many particles may be needed to cover the probability space and attain a sufficiently accurate estimate. Several methods have been developed to reduce the variance of the estimates, including decomposition (Ng, Peshkin, & Pfeffer, 2002) and abstraction (Verma, Thrun, & Simmons, 2003). One particularly effective method is Rao-Blackwellised Particle Filtering (Akashi & Kumamoto, 1977; Casella & Robert, 1996; Doucet, 1998). This method is based on a fundamental observation that with some estimation problems, a particular part of the desired

1. Initialization
   - For $i = 1, \ldots, N$ draw a random sample $\mathbf{r}_0^{(i)}$ from the prior distribution $p(\mathbf{r}_0)$ and let $\alpha_0^{(i)} \leftarrow p(\mathbf{s}_0|\mathbf{r}_0^{(i)})$
2. For $t = 1, 2, \ldots$
   (a) Importance sampling step
   - For $i = 1, \ldots, N$
     - draw a random sample $\tilde{\mathbf{r}}_t^{(i)}$ from the proposal $q(\mathbf{r}_t|\mathbf{r}_{0:t-1}^{(i)}, \mathbf{y}_{0:t}, \mathbf{u}_{0:t})$
     - let $\tilde{\mathbf{r}}_{0:t}^{(i)} \leftarrow (\mathbf{r}_{0:t-1}^{(i)}, \tilde{\mathbf{r}}_t^{(i)})$
   - For $i = 1, \ldots, N$, compute the importance weights: $w_t^{(i)} \leftarrow \frac{p(\mathbf{y}_t|\tilde{\mathbf{r}}_{0:t}^{(i)}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t}) p(\tilde{\mathbf{r}}_t^{(i)}|\tilde{\mathbf{r}}_{0:t-1}^{(i)}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t})}{q(\tilde{\mathbf{r}}_t^{(i)}|\tilde{\mathbf{r}}_{0:t-1}^{(i)}, \mathbf{y}_{0:t}, \mathbf{u}_{0:t})}$
   - For $i = 1, \ldots, N$ normalize the importance weights $w_t^{(i)}$
   (b) Exact step
   - Update $\alpha_t^{(i)}$ given $\alpha_{t-1}^{(i)}$, $r_t^{(i)}$, $r_{t-1}^{(i)}$, $\mathbf{y}_t$, $\mathbf{u}_{t-1}$, and $\mathbf{u}_t$ with a domain-specific procedure (such as a Kalman Filter)
   (c) Selection step
   - Select N particles (with replacement) from $\{\tilde{\mathbf{r}}_{0:t}^{(i)}\}$ according to the normalized weights $\{\tilde{w}_t^{(i)}\}$ to obtain samples $\{\mathbf{r}_{0:t}^{(i)}\}$

Figure 11: Generic RBPF algorithm. (Murphy & Russell, 2001)

distribution can be determined efficiently without using a sampling approach. By factoring out this portion, we obtain a more efficient approach that only samples the remaining variables.

Formally, if we partition the state variables into two sets, $\mathbf{r}$ and $\mathbf{s}$, we can use the chain rule to express the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ as

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) &= p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \\ &= p(\mathbf{s}_{0:t}|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) p(\mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \end{aligned} \tag{9}$$

Thus, we expand the posterior in terms of the sequence of random variables $\mathbf{r}_{0:t}$ and in terms of the sequence $\mathbf{s}_{0:t}$ conditioned on $\mathbf{r}_{0:t}$. The key to this formulation is that if we can compute analytically the conditional distribution $p(\mathbf{s}_{0:t}|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ or its marginal $p(\mathbf{s}_t|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$, then we only need to sample the sequences of variables $\mathbf{r}_{0:t}$, not $\langle \mathbf{s}_{0:t}, \mathbf{r}_{0:t} \rangle$. Intuitively, far fewer particles will be needed in this way to reach a given precision of the estimate, since for each sampled sequence $\mathbf{r}_{0:t}$, the corresponding state space $\mathbf{s}$ is covered by an analytical solution, rather than a finite number of samples.

In Rao-Blackwellised particle filtering (RBPF), each particle holds not only the samples $\mathbf{r}_{0:t}^{(i)}$, but also a *parametric* representation of the distribution $p(\mathbf{s}_t|\mathbf{r}_{0:t}^{(i)}, \mathbf{y}_{1:t})$ for each sample i, which we denote by $\alpha_t^{(i)}$. This representation holds sufficient statistics for $p(\mathbf{s}_t|\mathbf{r}_{0:t}^{(i)}, \mathbf{y}_{1:t})$, such as the mean vector and the covariance matrix of a Gaussian distribution.[2] The posterior is thus approximated as a mixture of the distributions $\alpha_t^{(i)}$ at the sampled points $\mathbf{r}_{0:t}^{(i)}$:

$$p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \approx \sum_1^N \alpha_t(i) \delta_{\mathbf{r}_{0:t}^{(i)}}(\mathbf{r}_{0:t}). \tag{10}$$

2. Certain distributions can be encoded compactly in terms of a *sufficient statistic*. Given this statistic, the distribution is defined completely.

A generic RBPF method is outlined in Figure 11, and, except for the initialization and the addition of the exact step, it is identical to the particle filter, illustrated in Figure 10. Under weak assumptions, the Rao-Blackwellised estimate converges to the estimated value as $N \rightarrow +\infty$, with a variance smaller than the non-Rao-Blackwellised particle filtering method with the same number of particles(Doucet et al., 2000). Hence the RBPF is superior given a fixed number of particles; however, the run-time performance of the filter will depend on the cost of the exact update for $\alpha_t^{(i)}$.

Having reviewed Rao-Blackwellisation, we now describe how we apply this concept to PHA.

3.3 Rao-Blackwellisation for Hybrid Estimation with PHA

When estimating the hybrid state of Probabilistic Hybrid Automata, the posterior distribution over the continuous state, $p(\mathbf{x}_{c,t}|\mathbf{x}_{d,0:t}, \mathbf{u}_{0:t})$, can be approximated efficiently in an analytical form using an Extended or Unscented Kalman Filter. We can, therefore, apply Rao-Blackwellisation to the hybrid estimation problem for PHA by taking $\mathbf{r} = \mathbf{x}_d$ and $\mathbf{s} = \mathbf{x}_c$.

We sample the mode sequences $\mathbf{x}_{d,0:t}^{(i)}$ with a particle filter and, for each sampled sequence $\mathbf{x}_{d,0:t}^{(i)}$, we estimate the continuous state with a Kalman Filter. The result of Kalman Filtering for each sampled sequence $\mathbf{x}_{d,0:t}^{(i)}$ is the estimated mean $\hat{\mathbf{x}}_{c,t}^{(i)}$ and the error covariance matrix $\mathbf{P}_t^{(i)}$. The samples $\mathbf{x}_{d,0:t}^{(i)}$ serve as an approximation of the posterior distribution over the mode sequences, $p(\mathbf{x}_{d,0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t})$, while each continuous estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle$ serves as a Gaussian approximation of the conditional distribution $p(\mathbf{x}_{c,t}|\mathbf{x}_{d,0:t} = \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{1:t}; \mathbf{u}_{0:t}) \triangleq \alpha_t^i$. Since the estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle$ merely approximates α_t^i, we are not performing a strict Rao-Blackwellisation; nevertheless, the distribution will be accurate up to the approximations in the Extended or the Unscented Kalman Filter.

The continuous estimate for each new mode sequence $\mathbf{x}_{d,0:t}^{(i)}$ is updated as shown in Figure 9ii). Since in a PHA, each mode assignment $\mathbf{d}$ over the variables $\mathbf{x}_d$ is associated with transition and observation functions

$$\begin{aligned} \mathbf{x}_{c,t} &= \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}; \mathbf{d}) + \mathbf{v}_x(\mathbf{d}) \\ \mathbf{y}_{c,t} &= \mathbf{g}(\mathbf{x}_{c,t}, \mathbf{u}_t; \mathbf{d}) + \mathbf{v}_y(\mathbf{d}), \end{aligned} \tag{11}$$

we update each estimate $\hat{\mathbf{x}}_{c,t-1}^{(i)}, P_{t-1}^{(i)}$ with a Kalman Filter, using the transition function $\mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}; \mathbf{d})$, observation function $\mathbf{g}(\mathbf{x}_{c,t}, \mathbf{u}_t; \mathbf{d})$, and noise variables $\mathbf{v}_x(\mathbf{x}_{d,t}^{(i)})$ and $\mathbf{v}_y(\mathbf{x}_{d,t}^{(i)})$, to obtain a new estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, P_t^{(i)} \rangle$.

Hence Rao-Blackwellisation can be applied to hybrid estimation with PHA. We now describe the rest of the algorithm in detail.

3.4 Proposal distribution

In particle filtering, the proposal distribution is chosen to be one that is close to the desired distribution, but that can also be calculated efficiently in closed form. In this section

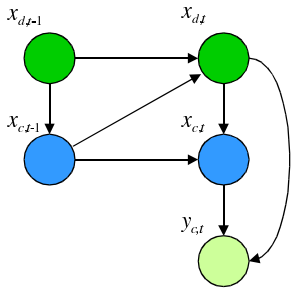


Figure 12: Conditional dependencies in PHA among the state variables $\mathbf{x}_c, \mathbf{x}_d$ and the output $\mathbf{y}$, expressed as a dynamic Bayesian network (Dean & Kanazawa, 1989). The edge from $\mathbf{x}_{c,t-1}$ to $\mathbf{x}_{d,t}$ represents the dependence of $\mathbf{x}_{d,t}$ on $\mathbf{x}_{c,t-1}$, that is, autonomous transitions.

we specify the proposal distribution $q(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ and show how it is calculated efficiently, while taking into account autonomous mode transitions.

An autonomous mode transition is a form of guarded transition for which the transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ depends explicitly on the continuous state $\mathbf{x}_{c,t-1}$. In PHA, the transition distribution is specified as a finite set of guard conditions c and their associated transition probabilities p_τ. Each guard condition specifies a region over the continuous state and automaton's input/output variables, for which $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) = p_\tau$. Hence the transition distribution is piecewise constant over $\mathbf{x}_{c,t-1}$ (see Figure 13).

We choose the proposal distribution to be $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$. This distribution expresses the probability of the transition from the mode $\mathbf{x}^{(i)}_{d,0:t-1}$ to each mode $\mathbf{x}_{d,t} \in \mathcal{X}_d$ and is similar in its form to the transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in a Markov process. However, it is conditioned on a complete discrete state sequence and all previous observations and control actions, rather than simply on the previous state. This is because $\{\mathbf{x}_{d,t}\}$ alone is *not* an HMM process: due to the autonomous transitions, knowing only $\mathbf{x}_{d,t-1}$ does not tell us what the distribution of $\mathbf{x}_{d,t}$ is. The distribution of $\mathbf{x}_{d,t}$ is known only when conditioned on the mode *and* the continuous state for the previous time step (see Figure 12). Since the continuous state must be estimated, rather than being observed directly, autonomous transitions make calculation of the proposal challenging.

We are able to calculate the proposal distribution efficiently for each tracked mode sequence $\mathbf{x}^{(i)}_{d,0:t-1}$ by using the corresponding continuous estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1} \rangle$ as follows: From the total probability theorem, the proposal distribution is equal to the joint distribution of $\mathbf{x}_{d,t}$ and $\mathbf{x}_{c,t-1}$, marginalized over $\mathbf{x}_{c,t-1}$. The joint distribution can then be expressed in terms of the discrete transition probability conditioned on the previous state, and the continuous state distribution conditioned on the i-th sequence, $p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \triangleq \alpha^i_{t-1}$:

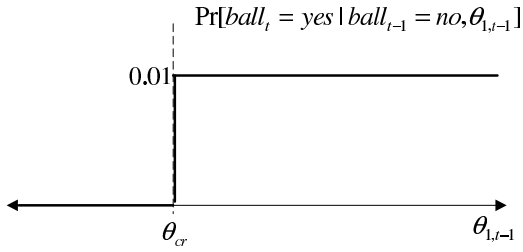


Figure 13: Probability of a mode transition `ball=no` to `ball=yes` as a function of $\theta_{1,t-1}$.

$$
\begin{aligned}
&p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \\
&\quad = \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})\ d\mathbf{x}_{c,t-1} \\
&\quad = \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{0:t}) p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})\ d\mathbf{x}_{c,t-1} \\
&\quad = \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ d\mathbf{x}_{c,t-1} \qquad (12)
\end{aligned}
$$

Here, the third equality comes from the independence assumptions made in the model; the distribution of $\mathbf{x}_{d,t}$ is independent of the observations $\mathbf{y}_{1:t-1}$ and mode assignments prior to time $t-1$, given the state at time $t-1$.

The proposal distribution can therefore be expressed as an integral over two known quantities; first, the transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$, which is expressed in the PHA model; and second, the continuous state distribution α^i_{t-1}. The latter is approximated by the continuous estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1} \rangle$.

Typically, when performing Rao-Blackwellisation, the integral in Equation 12 is difficult to evaluate efficiently(Murphy & Russell, 2001). In this section we show that for PHA, however, efficient evaluation of this integral is possible.

For the piecewise constant transition distribution in PHA, the left term in the integral in Equation 12, $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ takes on only a finite number of values $p_{\tau j}(\mathbf{x}_{d,t})$. Hence we can split the integral domain into the sets X_j that satisfy the guard condition c_j and factor out the transition probability $p_{\tau j}$:

$$
\begin{aligned}
&\int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ d\mathbf{x}_{c,t-1} \\
&\quad = \sum_j \int_{X_j} p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ d\mathbf{x}_{c,t-1} \\
&\quad = \sum_j p_{\tau j}(\mathbf{x}_{d,t}) \int_{X_j} p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ d\mathbf{x}_{c,t-1} \\
&\quad = \sum_j p_{\tau j}(\mathbf{x}_{d,t}) \Pr_{\alpha^{(i)}_{t-1}}[X_j] \qquad (13)
\end{aligned}
$$

Here, $\Pr_{\alpha_{t-1}^{(i)}}[X_j]$ is the probability that guard condition c_j is satisfied.

The second equality holds because, for the region X_j, the conditional distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ is fixed and equal to $p_{\tau j}$. Therefore, in each summed term, we multiply the transition distribution $p_{\tau j}$ by the probability of satisfying the guard condition c_j in the distribution $\alpha_{t-1}^{(i)}$. Hence, the key contribution for PHA is that, given the probability of satisfying each guard condition c_j, the proposal distribution for each sample i is calculated by summing over all of the possible guard conditions.

3.5 Evaluating the probability of satisfying transition guards

Given the derivation in the previous section, the remaining challenge in computing the proposal distribution is to evaluate the probability of satisfying the guard condition c_j, given the distribution $\alpha_{t-1}^{(i)} = p(\mathbf{x}_{c,t-1}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$. For the following derivation, we assume without loss of generality, that the guard conditions are only over the continuous state.[3]

As described in Section 3.3, the posterior distribution $\alpha_{t-1}^{(i)}$ is approximated using a Gaussian distribution with mean $\hat{\mathbf{x}}_{c,t-1}^{(i)}$ and covariance $\mathbf{P}_{t-1}^{(i)}$. The probability of $\mathbf{x}_{c,t}$ being in the guard region X_j is then simply an integral over a known Gaussian distribution:

$$\Pr_{\alpha_t^{(i)}}[X_j] \approx \frac{1}{(2\pi)^{n_c/2}|\mathbf{P}_{t-1}^{(i)}|^{1/2}} \int_{X_j} e^{-\frac{1}{2}(\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})^T {\mathbf{P}_{t-1}^{(i)}}^{-1} (\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})} \, d\mathbf{x}_c \tag{14}$$

When the guard conditions are of the form $x < c$ or $x \leq c$, for some constant c, such as $\theta_1 < 0.55$, the integral in Equation 14 simplifies to evaluating the cumulative density function $D(c)$ of the normal variable $\mathcal{N}(\mu, \sigma^2)$, where $\mu = (\hat{\mathbf{x}}_{c,t-1}^{(i)})_x$ is the mean of variable x in $\hat{\mathbf{x}}_{c,t-1}^{(i)}$ and $\sigma^2 = (\mathbf{P}_t^{(i)})_x$ is its variance (Figure 14):

$$D(c) \triangleq \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{c} e^{-(x-\mu)^2/(2\sigma^2)} dx. \tag{15}$$

The cumulative density function $D(c)$ is evaluated using standard numerical methods, such as a trapezoidal approximation or using a table lookup. In order to evaluate the probability of the complementary guards $x > c$ or $x \geq c$, we take the complement of the cumulative density function, $1 - D(c)$.

The above forms of guard conditions can be viewed as a special case of a more general form, in which x falls into an interval $[l, u]$.[4], where l, u are in the extended set of real numbers $\mathbb{R}^+ \triangleq \mathbb{R} \cup \{-\infty, +\infty\}$ that includes positive and negative infinity. In these cases, the probability of satisfying a guard condition can be expressed as the difference of the c.d.f at the endpoints of the interval, $D(u) - D(l)$.

3. Guard conditions involving discrete or continuous input variables, such as $c_c(\mathbf{x}_c) \wedge c_d(\mathbf{u}_d)$, are handled by setting $\Pr_{\alpha_{t-1}^{(i)}}[X_j] \equiv 0$ whenever $c_d(\mathbf{u}_d)$ is not satisfied. More complex guards are transformed to a number of simpler guards using elementary rules of logic.
4. Whether the interval is closed or open matters only if x can have a zero variance. It is straightforward to generalize the discussion here to open and half-open intervals.

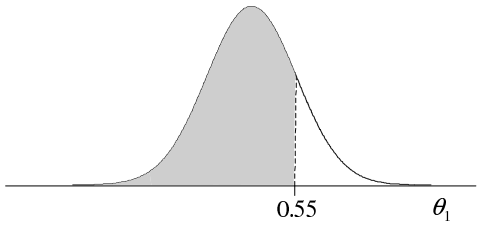


Figure 14: Evaluating single-variate guard conditions.

We have previously used this technique with our k-best enumeration approach for CPHA(Hofbaur & Williams, 2002a). Using this approach for RBPF, we are able to calculate the proposal distribution in Section 3.4 efficiently for a particular class of autonomous mode transitions; those whose guard conditions are singlevariate intervals. In Section 4.1 we generalize this method to apply to CPHA with multivariate linear guard conditions.

Within the CPHA modeling formalism, however, the transition distribution, that is, $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$, is still constrained to be piecewise constant in $\mathbf{x}_{c,t-1}$. We present results that show that this contraint can be relaxed. In Section 4.2 we generalize the approach in this section to calculate the transition prior $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ for arbitrary polynomial transition distributions. This transition prior is used in the Rao-Blackwellised Particle Filter, as the proposal distribution, and in k-best enumeration. Hence this contribution expands the class of autonomous mode transitions that can be handled by *both* approaches to hybrid estimation.

3.6 Importance weights

In this section we describe how, given our choice of proposal distribution, the importance weight can be calculated. The importance weight compensates for the discrepancy between the proposal distribution, which we chose to be the transition prior $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, and the desired distribution, $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. The importance weight determines how many duplicates of a given particle are generated, thereby adjusting the number of particles for which the proposal distribution did not match the desired distribution. In our case, the importance weight incorporates the latest observation in order to update the prior distribution, to give the posterior distribution.

Given our choice of proposal distribution, the weights $w^{(i)}_t$ simplify to

$$w^{(i)}_t \triangleq \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}^{(i)}_{d,0:t}, \mathbf{y}_{0:t-1})p(\tilde{\mathbf{x}}^{(i)}_{d,t}|\tilde{\mathbf{x}}^{(i)}_{d,0:t-1}, \mathbf{y}_{0:t-1})}{q(\tilde{\mathbf{x}}^{(i)}_{d,t}; \tilde{\mathbf{x}}^{(i)}_{d,0:t-1}, \mathbf{y}_{0:t})} = p(\mathbf{y}_t|\tilde{\mathbf{x}}^{(i)}_{d,0:t}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t}) \tag{16}$$

This expression represents the likelihood of the observation $\mathbf{y}_t$, given a complete mode sequence $\tilde{\mathbf{x}}^{(i)}_{d,0:t}$, inputs $\mathbf{u}_{0:t}$, and previous observations $\mathbf{y}_{0:t-1}$. PHA, like most hybrid models, do not directly provide this likelihood and only provide the probability of an observation $\mathbf{y}$, conditioned on the discrete and continuous state. This likelihood is approximated using the Kalman Filter innovation at time t as follows:

In the Kalman Filter predict/measurement cycle, the Gaussian distribution $\alpha^{(i)}_{t-1} = \mathcal{N}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1})$ is propagated through the continuous transition and observation functions

for mode $\mathbf{x}_{d,t}^{(i)}$ (Equation 11). For SLDS models, this gives a Gaussian distribution for the observed value $\mathbf{y}_t$, with mean $\mathbf{y}_p$ and covariance $S_t^{(i)}$. The Kalman Filter *innovation*, $\mathbf{r} = \mathbf{y}_t - \mathbf{y}_p$, is defined as the difference between the expected observation and the actual observation. For SLDS models, the observation likelihood in Equation 16 is calculated *exactly* as

$$w_t^{(i)} = \frac{1}{(2\pi)^{N/2}|\mathbf{S}_t^{(i)}|^{1/2}} e^{-0.5\mathbf{r}^T(\mathbf{S}_t^{(i)})^{-1}\mathbf{r}} \tag{17}$$

(Blom & Bar-Shalom, 1988).

A similar approach leads to an efficient approximation of the weight in the case when the model contains nonlinear dynamics and autonomous transitions. Due to the nonlinearities and autonomous transitions, the conditional distribution $\alpha_{t-1}^{(i)} = p(\mathbf{x}_{c,t-1}|\tilde{\mathbf{x}}_{d,0:t}^{(i)}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t})$ is no longer strictly Gaussian. Nevertheless, if we approximate it with the estimated Gaussian $\mathcal{N}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)})$, as we have done in the previous subsections, we can compute the weight in Equation 16 from the Extended(Anderson & Moore, 1979) or the Unscented(Julier & Uhlmann, 1997) Kalman Filter measurement update. For example, with an Extended Kalman Filter, the observation likelihood is computed by first propagating the Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)})$ through the system model in mode $\mathbf{x}_{d,t}^{(i)}$:

$$\begin{aligned}
\hat{\mathbf{x}}_{c,t}^{(i-)} &= \mathbf{f}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{u}_{t-1}) && (18)\\
\mathbf{A} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}_c}|\hat{\mathbf{x}}_{c,t}^{(i)} && (19)\\
\mathbf{P}_t^{(i-)} &= \mathbf{A}\mathbf{P}_t^{(i-)}\mathbf{A}^T + \mathbf{Q}, && (20)
\end{aligned}$$

where $\mathbf{Q} = cov(\mathbf{v}_x(\mathbf{x}_{d,t}^{(i)}))$ is the system noise in mode $\mathbf{x}_{d,t}^{(i)}$. This leads to the observation prediction $\mathbf{y}_p$ with covariance $\mathbf{S}_t^{(i)}$:

$$\begin{aligned}
\mathbf{y}_p &= \mathbf{g}(\hat{\mathbf{x}}_{c,t}^{(i-)}, \mathbf{u}_t) && (21)\\
\mathbf{C} &= \frac{\partial \mathbf{g}}{\partial \mathbf{x}_c}|\hat{\mathbf{x}}_{c,t}^{(i-)} && (22)\\
\mathbf{S}_t^{(i)} &= \mathbf{C}\mathbf{P}_t^{(i-)}\mathbf{C}^T + \mathbf{R}, && (23)
\end{aligned}$$

where $\mathbf{R} = cov(\mathbf{v}_y(\mathbf{x}_{d,t}^{(i)}))$ is the observation noise in mode $\mathbf{x}_{d,t}^{(i)}$. The observation likelihood in Equation 16 can then be approximated with normal p.d.f.

$$w_t^{(i)} = \frac{1}{(2\pi)^{N/2}|\mathbf{S}_t^{(i)}|^{1/2}} e^{-0.5\mathbf{r}^T(\mathbf{S}_t^{(i)})^{-1}\mathbf{r}}. \tag{24}$$

We have therefore presented an novel, approximate Rao-Blackwellised particle filtering algorithm for Probabilistic Hybrid Automata. This is able to handle autonomous mode transitions and nonlinear dynamics. in Section 3.7 we extend this method to handle concurrent Probabilistic Hybrid Automata (CPHA).

3.7 Rao-Blackwellised Particle Filtering for CPHA

In practice, a model will be composed of several concurrently operating automata that represent individual components of the underlying system. In this manner, the design of the models can be split on a component-by-component basis, thus enhancing the reusabiliy of the models and reducing modeling costs. In this section we extend our Rao-Blackwellised Particle Filter for PHA, developed in the previous section, to handle concurrent PHA models; see Section 2.1 for an overview of CPHA.

In CPHA, components transitions are conditionally independently, given the current discrete and continuous state. Therefore, it is possible to compute the transition probabilities $P_{\mathcal{T}}^{(i)}$ for each tracked mode sequence component-wise (Nayak & Williams, 1997)(Hofbaur & Williams, 2002a). This property is exploited by our algorithm in the importance sampling step, whereby the samples are evolved according to the transition distribution $P_{\mathcal{T}}$ on a component-by-component basis.

Recall that the algorithm in Section 3 sampled the mode sequences according to the proposal distribution $q(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \triangleq P_{\mathcal{T},t}^{(i)}$. This represents the prior probability of being in the mode $\mathbf{x}_{d,t}$ at time t, conditioned on the previous sequence of modes $\mathbf{x}_{d,0:t-1}^{(i)}$ and observations $\mathbf{y}_{1:t-1}$, leading to that mode. Given this choice of the proposal, the importance weights simplify to

$$w_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \triangleq P_{\mathcal{O},t}^{(i)}. \tag{25}$$

When sampling mode sequences in CPHA, we use the same proposal distribution. The only difference is that now, instead of computing the transition probability for every value in the domain $\mathcal{X}_d$ of the discrete variables $\mathbf{x}_d$, we evaluate it only for the individual component's discrete domain $\mathcal{X}_{dj}$, and obtain the joint transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ as a product of component transition distributions $\prod_j p(\mathbf{x}_{dj,t}|\mathbf{x}_{dj,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, for all components j in the model (see Section 3.4).

The final algorithm is shown in Figure 15.[5] To summarize, using a Rao-Blackwellised Particle Filtering approach, this algorithm is able to estimate efficiently the hybrid state of Concurrent Probabilistic Hybrid Automata, which have autonomous mode transitions, nonlinear dynamics, and many concurrently operating components.

4. Generalizing Autonomous Mode Transitions

In this section we extend the class of autonomous mode transitions that can be handled by both k-best enumeration and Rao-Blackwellised Particle Filtering for CPHA. We describe the generalization to multivariable linear transition guards, in the case of piecewise constant transition distributions, and to polynomial transition distributions of arbitrary order, in the case of single variables.

5. Note that, compared to the generic RBPF algorithm in Figure 11, the importance weight is now calculated after the exact step, because the innovation mean and covariance, computed in the Kalman Filter update step, are used to compute the importance weight.

1. Initialization
   - For $i = 1, \ldots, N$
     - draw a random sample $\mathbf{x}_{d,0}^{(i)}$ from the prior distribution $p(\mathbf{x}_{d,0})$
     - initialize the estimate mean $\hat{\mathbf{x}}_{c,0}^{(i)} \leftarrow \mathbb{E}[\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(i)}]$
     - initialize the estimate covariance $\mathbf{P}_0^{(i)} \leftarrow Cov(\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(i)})$
2. For $t = 1, 2, \ldots$
   (a) Importance sampling step
   - For $i = 1, \ldots, N$
     - For each component k
       * compute the transition distribution $p(\mathbf{x}_{dk,t}|\mathbf{x}_{dk,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
       * sample $\tilde{\mathbf{x}}_{d,t\ k}^{(i)} \sim p(\mathbf{x}_{dk,t}|\mathbf{x}_{dk,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
     - let $\tilde{\mathbf{x}}_{d,0:t}^{(i)} \leftarrow (\mathbf{x}_{d,0:t-1}^{(i)}, \langle \tilde{\mathbf{x}}_{d,t\ 1}^{(i)}, \ldots, \tilde{\mathbf{x}}_{d,t\ n_c}^{(i)} \rangle)$

   (b) Exact step
   - For $i = 1, \ldots, N$
     - perform a KF update: $\tilde{\mathbf{x}}_{c,t}^{(i)}, \tilde{\mathbf{P}}_t^{(i)}, \mathbf{r}_t^{(i)}, \mathbf{S}_t^{(i)} \leftarrow UKF(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)}, \tilde{\mathbf{x}}_{d,t}^{(i)})$
     - compute the importance weight: $w_t^{(i)} \leftarrow \mathcal{N}(\mathbf{r}_t^{(i)}, \mathbf{S}_t^{(i)})$

   (c) Selection step
   - normalize the importance weights $w_t^{(i)}$
   - Select N particles (with replacement) from $\{\langle \tilde{\mathbf{x}}_{d,0:t}^{(i)}, \tilde{\mathbf{x}}_{c,t}^{(i)}, \tilde{\mathbf{P}}_t^{(i)} \rangle\}$ according to the normalized weights $\{\tilde{w}_t^{(i)}\}$ to obtain particles $\{\langle \mathbf{x}_{d,0:t}^{(i)}, \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle\}$

Figure 15: Rao-Blackwellised Particle Filter for CPHA.

4.1 Multi-variate guard conditions

The key challenge in handling autonomous mode transitions is to compute the transition prior $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ efficiently.

In the case of CPHA, this probability can be expressed as a sum over a finite number of terms, where each term is a product of the probability $\Pr_{\alpha_{t-1}^{(i)}}[X_j]$ of a guard condition c_j being satisfied, and the corresponding transition probability $p_{\tau j}$ (Equation 13). The remaining challenge is to calculate $\Pr_{\alpha_{t-1}^{(i)}}[X_j]$. Our previous work showed how this can be carried out efficiently for guard conditions that are single-variate intervals(Funiak & Williams, 2003)(Hofbaur & Williams, 2004). In this section we show that this can be generalized to rectangular and linear multivariate guard conditions.

4.1.1 Rectangular Multivariate Guard Conditions

Multivariate guard conditions are often needed to represent more complex constraints on transitions than can be handled by single-variate interval conditions. For example, consider a system with two tanks, connected by a pipe at height h. The mode transitions for the flow between the two tanks are constrained by how the fluid heights in the two tanks, h_1 and h_2, compare to h. In this system, the transition into the `no-flow` mode would be conditioned on the guard condition $(h_1 < h) \wedge (h_2 < h)$.

In general, the rectangular multi-variate guard conditions will take the form $\bigwedge_{i \in I}(x_i \in [l_i, u_i])$, where x_i are distinct continuous state variables and $I \triangleq \{i_1, \ldots, i_n\}$ are their indices.

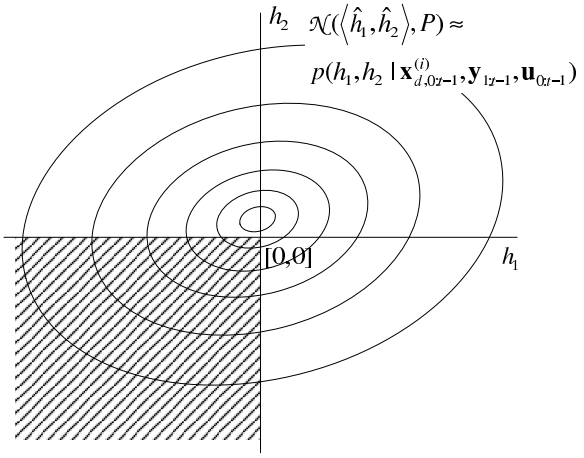


Figure 16: Rectangular integral over a Gaussian approximation of the posterior density of h_1 and h_2, $p(h_1, h_2 | \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$.

Evaluating the probability of such a multi-variate guard condition amounts to evaluating the multi-dimensional (hyper)rectangular integral over a Gaussian distribution (see Figure 16):

$$\Pr_{\alpha^{(i)}_t}[X_j] \approx \frac{1}{(2\pi)^{n_c/2}|\mathbf{P}_I|^{1/2}} \int_{l_{i_1}}^{u_{i_1}} \int_{l_{i_2}}^{u_{i_2}} \cdots \int_{l_{i_n}}^{u_{i_n}} e^{-\frac{1}{2}(\mathbf{x}_c - \hat{\mathbf{x}}_{c,I})^T \mathbf{P}_I{}^{-1} (\mathbf{x}_c - \hat{\mathbf{x}}_{c,I})} \, d\mathbf{x}_c, \tag{26}$$

where $\hat{\mathbf{x}}_{c,I}$ is the mean of guard values, selected from the continuous state estimate $\hat{\mathbf{x}}^{(i)}_{c,t-1}$, and $\mathbf{P}_I$ is the covariance matrix of guard values, selected from the estimate covariance $\mathbf{P}^{(i)}_{t-1}$. Rectangular integrals over Gaussian distributions are evaluated efficiently using numerical methods, such as those presented in (Joe, 1995; Genz & Kwong, 2000). As an alternative, one could use Monte Carlo methods to evaluate the integral 26. Numerical methods tend to perform better, hence this is the approach used in our system.

4.1.2 Linear multi-variate guards

Sometimes, transition guards are best represented by a linear combinations of continuous variables. For example, in a two-tank system, the direction of the flow between the two tanks depends on the heights in the two tanks. Hence, the mode variable for the flow direction would be guarded by the linear guards $h_1 - h_2 > 0$ and $h_1 - h_2 < 0$ (see Figure 17). While it would be possible to include $h_1 - h_2$ as a derived state variable in the model, doing so would increase the computational complexity of the Kalman Filter update by one dimension, and would make the covariance matrix singular. A singular covariance matrix prevents the Kalman Filter update essential to the Rao-Blackwellised Particle from being carried out. Instead, we apply a linear transform to the Gaussian distribution, thus reducing the computation to an instance of the rectangular multivariate integral performed in Section 4.1.1.

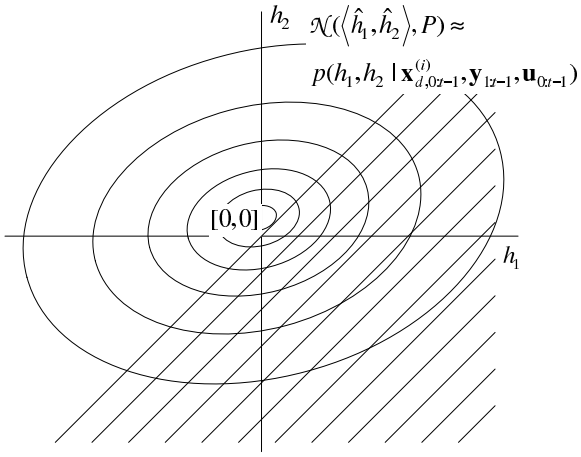


Figure 17: Linear guard condition $h_2 < h_1$ over the Gaussian approximation of the posterior density of h_1 and h_2.

Suppose that the guard condition c is expressed as a conjunction of clauses $\bigwedge_{i=1}^{n} l_i < \mathbf{a}_i \mathbf{x}_c < u_i$, where $\mathbf{a}_i$ is the vector of *guard coefficients* that specify condition c_i. Such guard conditions correspond to a convex space that is formed as an intersection of hyper-planes $l_i < \mathbf{a}_i \mathbf{x}_c$ and $\mathbf{a}_i \mathbf{x}_c < u_i$. $\mathbf{A} \triangleq \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix}^T$ be the matrix of the guard coefficients and define $\mathbf{z} \triangleq \mathbf{A}\mathbf{x}_{c,t-1}$ be the derived vector with n elements. Then the guard condition $c \triangleq \bigwedge_{i=1}^{n} l_i < \mathbf{a}_i \mathbf{x}_c < u_i$ is equivalent to the guard condition $\bigwedge_{i=1}^{n} l_i < \mathbf{z}_i < u_i$. The probability of the guard condition c being satisfied can thus be evaluated as an integral

$$\int_{l_{i_1}}^{u_{i_1}} \int_{l_{i_2}}^{u_{i_2}} \cdots \int_{l_{i_n}}^{u_{i_n}} p(\mathbf{z}_t | \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) \, d\mathbf{x}_c, \tag{27}$$

over the rectangular region $[l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_n, u_n]$.

For this general case, the posterior distribution of $\mathbf{z}$ is as intractable as the posterior distribution $\alpha^{(i)}_{t-1}$ of $\mathbf{x}_{c,t-1}$, thereby defeating the purpose of the proposal distribution. However, since we approximate $\alpha^{(i)}_{t-1}$ with a Gaussian $\mathcal{N}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1})$, the distribution of $\mathbf{z}$ is Gaussian, with a mean $\mathbf{A}\hat{\mathbf{x}}^{(i)}_{c,t-1}$ and a covariance $\mathbf{A}\mathbf{P}^{(i)}_{t-1}\mathbf{A}^T$. Therefore, the key result is that the linear guard conditions can once again be evaluated as a rectangular integral over a Gaussian distribution.

4.2 Polynomial Transition Distributions

The PHA formalism specifies a finite set of guarded transitions between discrete modes, each with a constant transition probability, given that the guard condition is satisfied. Hence the transition distribution $p(\mathbf{x}_{d,t} | \mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ is piecewise constant in $\mathbf{x}_{c,t-1}$ (Figure 13).

In this section we present a method for calculating the transition prior, in other words $p(\mathbf{x}_{d,t} | \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, when the transition distribution $p(\mathbf{x}_{d,t} | \mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ is *not* piecewise constant. In particular, we show that the integral in Equation 12 can be calculated

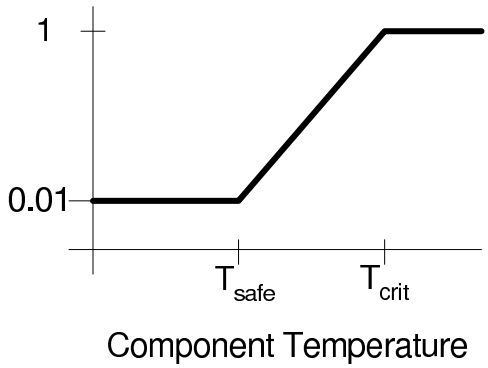


Figure 18: Probability of transition into `failed` mode for a simple component as a function of temperature. Below the safe temperature, T_{safe}, there is a small failure probability. Above this, the failure probability increases linearly until T_{crit}, at which point failure is guaranteed.

efficiently when the transition distribution is described by a piecewise polynomial function of arbitrary order, for transition distributions defined over a single variable. An example of a piecewise polynomial transition distribution is shown in Figure 18.

This contribution greatly expands the class of models about which our hybrid estimation approaches can reason, since piecewise polynomial functions of arbitrary order can approximate any piecewise smooth function to arbitrary accuracy.

For a transition distribution defined over a single variable x in $\mathbf{x}_{c,t-1}$, meaning that, $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) = p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1})$ the transition prior can be written as:

$$\begin{aligned} &p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \\ &\quad = \int_x p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\, dx. \end{aligned} \tag{28}$$

As in Section 3.5 we approximate $p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$ using a Gaussian distribution with mean μ and covariance σ. As before, we assume the cumulative distribution function $D(c)$, given by Equation 15, can be calculated efficiently.

If $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1})$ is piecewise constant in x, the transition prior is evaluated using the method described in Section 3.4. This requires the evaluation of $D(c)$ at the boundary values of every guard condition. Consider now the case where the transition distribution is piecewise *linear* in x:

$$\begin{aligned} p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) &= a_{0,j} + a_{1,j}x \\ x &\in X_j \end{aligned} \tag{29}$$

When calculating the transition prior in Equation 28, the integral can be split into a finite number of integrals of the following form:

$$\int_x p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ dx$$
$$= \frac{1}{\sigma\sqrt{2\pi}} \sum_j \int_{X_j} (a_{0,j} + a_{1,j}) x e^{-(x-\mu)^2/(2\sigma^2)}\ dx \tag{30}$$

The integral for each guard condition can be rewritten as:

$$\int_{X_j} a_{1,j}(x-\mu) e^{-(x-\mu)^2/(2\sigma^2)}\ dx + \int_{X_j} (a_{0,j} + a_{1,j}\mu) e^{-(x-\mu)^2/(2\sigma^2)}\ dx \tag{31}$$

We now use the result that the integral,

$$\int_l^u (x-\mu) e^{-(x-\mu)^2/(2\sigma^2)}\ dx = \left[-\sigma^2 e^{-(x-\mu)^2/2\sigma^2}\right]_l^u \tag{32}$$

can be calculated in closed form using a substitution of the form $v = (x-\mu)^2$. Assuming the guard region X_j takes the form $x \in [l, u]$ this means the first term in Equation 31 can be calculated in closed form. The second term in Equation 31 is an interval integral over the Gaussian p.d.f. for x, and this is evaluated using the the cumulative distribution function $D(c)$, as in Equation 15. Hence for linear transition functions, the transition prior can be calculated efficiently.

We now generalize this result to polynomial transition functions of the form:

$$p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) = \sum_{i=0}^{n_p} a_{i,j} x^i$$
$$x \in X_j. \tag{33}$$

The transition prior in Equation 28 can now be decomposed further, according to each term of the polynomial:

$$\int_x p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ dx$$
$$= \sum_j \sum_i \int_{X_j} a_{i,j} x^i p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ dx. \tag{34}$$

The polynomial in Equation 33 can be rewritten as:

$$\sum_{i=0}^{n_p} a_{i,j} x^i = \sum_{i=0}^{n_p} a'_{i,j} (x-\mu)^i, \tag{35}$$

and correspondingly the transition prior can be written as:

$$\int_x p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, x, \mathbf{u}_{t-1}) p(x|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ dx$$
$$= \sum_j \sum_i \frac{1}{\sigma\sqrt{2\pi}} \int_{X_j} a'_{i,j} (x-\mu)^i e^{-(x-\mu)^2/(2\sigma^2)}\ dx \tag{36}$$

By repeated integration by parts, the more general form of the integral in Equation 32,

$$\int_l^u (x-\mu)^i e^{-(x-\mu)^2/2\sigma^2}\, dx$$
$$= \left[-\sigma^2(x-\mu)^{i-1}e^{-(x-\mu)^2/2\sigma^2}\right]_l^u + \int_l^u (i-1)(x-\mu)^{i-2}\sigma^2 e^{-(x-\mu)^2/2\sigma^2}\, dx, \quad (37)$$

can be reduced to a summation of closed form terms, plus either the integral in Equation 32, in the case of odd i, or the integral in Equation 15, in the case of even i. In the first case the integral is evaluated entirely in closed form, while in the second case the only non-closed form term is the integral over the p.d.f., which is evaluated using $D(c)$. Furthermore, all of the terms that cannot be evaluated in closed form are scaled versions of the same integral over the p.d.f.; hence, $D(c)$ needs only to be evaluated at the boundaries of the guard regions, as was the case for the piecewise constant transition function.

Thus, for transition distributions over single variables, the transition prior is calculated efficiently for transition distributions that are piecewise polynomial. This contribution greatly expands the class of models about which both Rao-Blackwellised particle filtering and existing k-best methods for CPHA can reason, since piecewise polynomial functions of arbitrary order can approximate any piecewise smooth function to arbitrary accuracy.

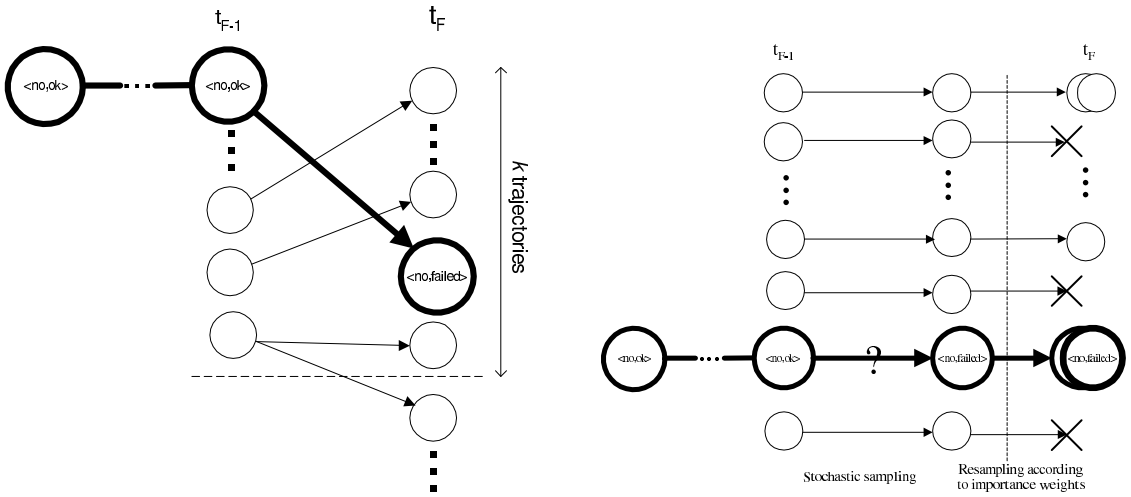
5. Hybrid Estimation using a Mixed Stochastic/Greedy Method

5.1 A Unified Treatment of k-best Enumeration and RBPF

Our objective, as stated in the introduction, is a robust, memory efficient method for Gaussian filtering of CPHA. Robustness is achieved by balancing exploration with exploitation, while memory efficiency is achieved by using a mixture-of-Gaussians representation. To this end, Section 2.3 described our previous method for Hybrid Estimation with CPHA based on greedy successor enumeration. Section 3 described a new exploration method based on Rao-Blackwellised Particle Filtering. These methods lend themselves to unification, in that they represent the belief state as a mixture of Gaussians, with each Gaussian representing a mode trajectory $\mathbf{x}_{d,0:t}$, and in both methods the continuous distribution, conditioned on each trajectory, $p(\mathbf{x}_{c,t}|\mathbf{x}_{d,0:t}, \mathbf{y}_{1:t})$ may be approximated using a Kalman Filter. In addition, both methods use the same approach to evaluating the transition prior $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ in the presence of autonomous mode transitions.

Finally, both methods approximate the posterior belief state by tracking only a subset of the reachable mode trajectories. The key difference between the two methods is the approach used to select which mode trajectories to track; k-best enumeration greedily selects the k trajectories with the highest posterior probability $p(\mathbf{x}_{d,0:t}|\mathbf{y}_{1:t})$, while RBPF uses stochastic sampling. To gain insight into the relative behavior of these two methods, and their appropriate combination, in Section 5.2 we evaluate their performance on a fault detection scenario. We show that for a posterior distribution that is flat across many distinct hypotheses, stochastic methods can be more successful than greedy enumeration.

However by representing the posterior probability $p(\mathbf{x}_{d,1:t}|\mathbf{y}_{1:t})$ using the number of repeated particles, RBPF introduces additional approximation and typically reduces the number of distinct trajectories tracked at any time. This means that for a relatively con-



(a) Failure detection with k-best enumeration (b) Failure detection with RBPF

Figure 19: In (a), 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. In (b), the question mark indicates that the generation of the successor corresponding to the true trajectory is generated stochastically.

centrated posterior, k-best enumeration will typically outperform RBPF. These results are demonstrated in Section 6 using the simulated acrobatic robot.

In Section 5.3 we develop a new algorithm that unifies the two approaches, exploiting the differences between them in order to increase the robustness of the algorithm to changes in the variance of the posterior distribution. In Section 6.1 experimental results are presented that validate this insight.

5.2 Comparing Greedy and Stochastic Search

Consider the acrobatic robot modeled in Figures 1, 2 and 3.

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. 19(a), 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.[6] By contrast, with the RBPF, mode transitions are sampled stochastically (Fig. 19(b)). For the true trajectory to be tracked, the mode transition into the failure

6. It is insufficient to consider the fault transition several time steps later. A mode trajectory that has the fault trajectory occurring much later will have an estimate of the continuous mode that does not match the observations.

state must be sampled. Since this transition has a low prior many particles will be needed, in order to detect the fault.

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, 2002b). 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.

5.3 A Novel Combined Greedy/Stochastic Algorithm

In this section we present a novel algorithm that combines the greedy and stochastic approaches, of k-best enumeration and RBPF respectively, to explore the discrete mode trajectories of the system (Fig. 20). 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. In Section 6 we show empirically that the new algorithm is more robust than k-best and RBPF taken individually, with only a minor performance penalty.

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. 4, 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

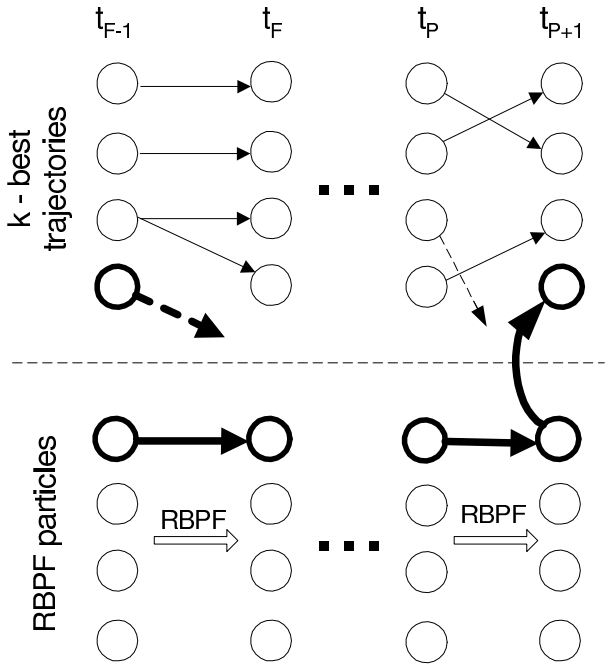


Figure 20: Failure detection with our mixed method. At time t_F the true trajectory (bold line) is no longer among the trajectories with the k highest posteriors. However, the true trajectory is sampled stochastically and becomes a member of the RBPF particle set. At time t_{P+1}, the posterior of the true trajectory becomes large enough for it to be included in the leading k trajectories.

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.

The pseudocode for the resulting algorithm is shown in Fig. 21. Part 2(b) of the algorithm implements an A* search for the k successors with the highest posterior. As in Section 2.3, partial paths in the search tree correspond to partial assignments of modes to components. A goal candidate is one that has a full assignment of modes, and for which the posterior probability $b(\mathbf{x}_{d,1:t})$ has been calculated. The two key additions to the method in Section 2.3 are as follows:

1. *Addition of RBPF particles to search queue:* In 2(a), the RBPF update step is carried out as described in Section 3.7, and the resulting particles are added to the search queue as candidates. 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. Part 2(b) ensures that unique trajectories only are added to the set of k best.

1. Initialization
   - Initialize Rao-Blackwellised Particle Filter with $N - k$ particles
   - Initialize k-best trajectories
2. For $t = 1, 2, \ldots$
   (a) RBPF update
      - Do RBPF update and add candidates to priority queue
   (b) A* search for successors with highest posterior
      - Add current trajectories to priority queue
      - **While** $size(new_kbest) < k$ **do**
        - Remove candidate from priority queue
        - If candidate is a goal then
          * Pop candidates until a unique one is found
          * Add unique candidate to new_kbest
        - **Else**
          * Expand candidate to its successors
          * Add successors to priority queue
      - Normalize new k-best trajectories

Figure 21: Belief state update using greedy and stochastic search.

In the code shown, uniqueness is ensured 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.

6. Experimental Results

In this section we present three main contributions. First, using individual estimation runs, we demonstrate empirically the insight in Section 5.2, that both k-best enumeration and Rao-Blackwellised Particle Filtering can perform poorly depending on whether the posterior distribution is concentrated in relatively few mode trajectories, or is flat across many. Secondly, we perform a detailed empirical analysis of the relative performance of k-best enumeration and RBPF that confirms this result and illustrates the need for our robust algorithm that balances the complementary approaches of greedy exploitation and stochastic exploration. Finally, we demonstrate that the new mixed algorithm is significantly more robust than either approach alone. We do not compare the Rao-Blackwellised approach with a standard particle filtering approach, since this analysis has been carried out extensively by Doucet, Gordon, and Krishnamurthy (2001b), Morales-Menéndez et al. (2002), Hutter and Dearden (2003); this analysis revealed that Rao-Blackwellisation increases the efficiency of the particle filtering approach dramatically.

We consider the acrobatic robot introduced in Section 2.1. In this model, the system dynamics are represented by four continuous variables: θ_1, the angle that the robot holds with the vertical plane, θ_2, the angle between the robot's torso and its legs, and the corresponding angular velocities ω_1 and ω_2. The discrete state of the hybrid model for this system consists of two variables, representing whether or not the robot holds a ball on its lower link (variable `ball`) and whether or not its actuator has broken (variable `actuator`). The complete discrete transition model is shown in Figure 4. The goal of hybrid estimation

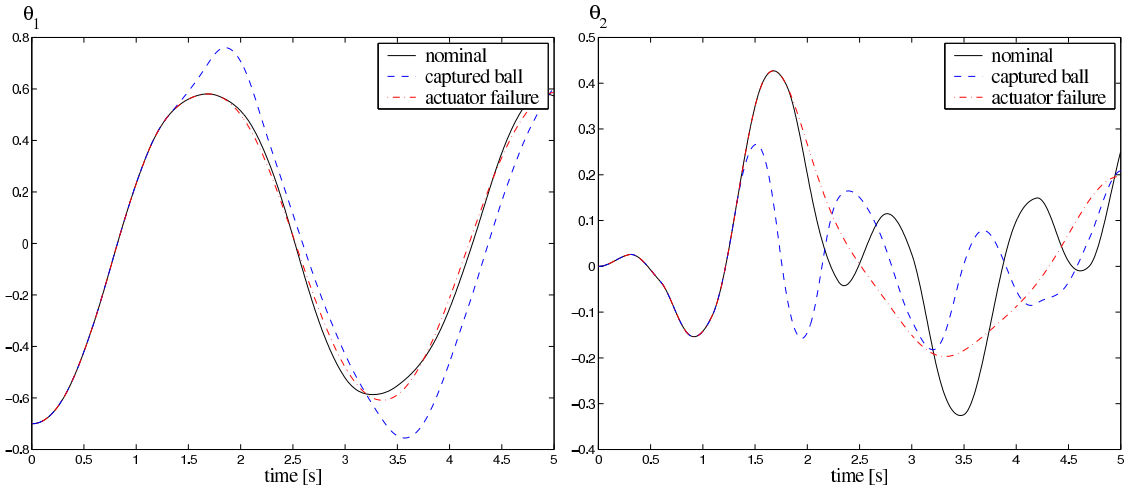


Figure 22: The evolution of θ_1 (left) and θ_2 (right) for the acrobot model scenarios.

in this case is to filter out the acrobot's hybrid state from a sequence of noisy observations of θ_2.

While the acrobot model is small, it demonstrates interesting challenges for hybrid state estimation. First, the dynamic model for a two-link system such as the acrobatic robot is highly nonlinear. Second, with four continuous state variables, the hybrid model is already too large to be handled by non-Rao-Blackwellised particle filters in real time. Third, the symptoms exhibited by mode changes are very subtle. Over ten time steps, the difference in the continuous trajectory between the *nominal* and *failure* scenarios described below is less than 0.04rad, which is the standard deviation of our chosen observation noise. Furthermore, experimental results using a small model yield insight about the underlying algorithms much more easily than a larger one. Finally, whether a particular hybrid estimation problem is challenging or not is not necessarily related to the size of the underlying model; in particular, we show in this section that other factors, such as the concentration of the posterior distribution, can be more important.

We considered the following three scenarios for hybrid estimation with the acrobot model:

1. In the *nominal* scenario, the robot remains in the nominal mode $\langle$`ball=no`, `actuator=ok`$\rangle$ for the duration of the experiment.

2. In the *ball* scenario, the robot captures a ball at time $t = 1.3$s and keeps it for the rest of the experiment. Capturing a ball increases the weight m_2 at the end of the lower link and changes the resulting trajectory, as shown in Figure 22.

3. In the *failure* scenario, the robot's actuator breaks at $t = 1.5$. This event causes the actuator to stop exerting any torque, and alters the robot's trajectory, as shown in Figure 22.

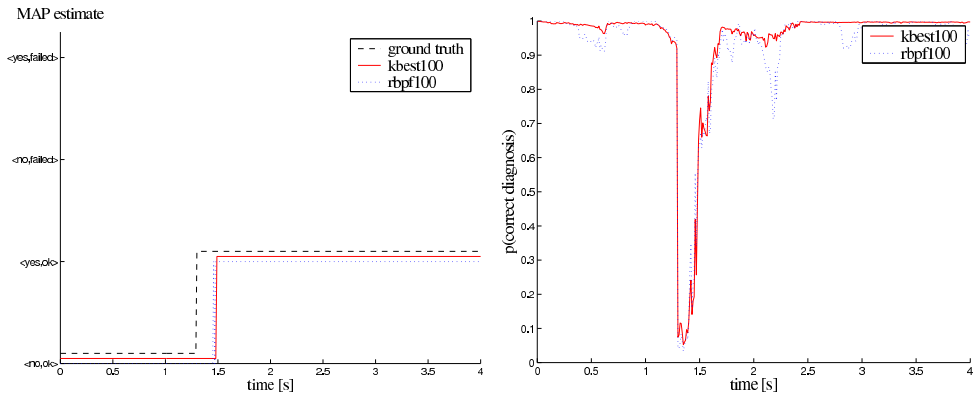


Figure 23: A single run for the ball capture scenario with concentrated posterior. Left: Maximum a posteriori (MAP) mode estimate computed by the Rao-Blackwellised Particle Filter (rbpf) and the k-best filter (kbest). Right: probability of the correct diagnosis $\langle$`ball=yes`, `actuator=ok`$\rangle$ for $t \geq 1.3s$.

6.1 Individual Estimation Runs

In this section we present results for hybrid estimation with the ball scenario and the failure scenario for single executions of the k-best algorithm and the new Rao-Blackwellised Particle Filter algorithm for CPHA.

6.1.1 Concentrated Posterior

First, we examine the original acrobatic robot model, shown in Figures 2 and 4. 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.

Figure 23 shows the maximum a posteriori (MAP) estimate of the discrete state by k-best and RBPF estimation filters for the ball scenario, for a single simulation run with 100 tracked sequences. Both k-best and RBPF algorithms estimate the MAP diagnosis correctly, except for an uncertain area close to the transition. In this uncertain area, the probability assigned by both filters to the correct diagnosis drops to less than 0.1, eventually returning to approximately 1. The reason for the delay between when the ball transition occurs, and when it is identified as the most likely diagnosis, is that the posterior probability of the true mode trajectory builds up only over time. The low probability of a transition to $\langle$`ball=yes`, `actuator=no`$\rangle$ biases the prior towards the $\langle$`ball=no`, `actuator=ok`$\rangle$ diagnosis. Only over time does the observation likelihood due to the correct diagnosis build up to compensate for this bias.

The new Rao-Blackwellised Particle Filter has tracked the continuous state of the acrobot system very well, even for the continuous states θ_1, ω_1, and ω_2 which were not directly observed. Figure 24 shows the tracking of θ_1 for the ball scenario. The tracked estimate for both k-best and RBPF is very close to the ground truth.

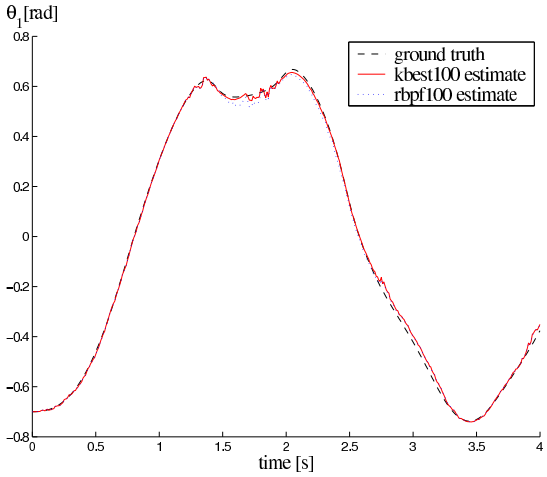


Figure 24: Filtered θ_1 for an execution of the ball scenario.

Figure 25 shows the results for a single simulation run with the failure scenario. In this case the k-best algorithm is able to diagnose the fault after a delay. [7] The probability assigned to the ground truth drops close to zero immediately after the fault, but gradually increases as the observations reveal that this is in fact the most likely diagnosis. However the Rao-Blackwellised Particle Filter does not diagnose the correct mode, even after several seconds. The probability assigned by the filter to the ground truth drops to zero just after the transition, and remains there. As described in Section 5.2, this is because the low prior of the fault transition makes it unlikely that the true mode trajectory is sampled unless many more particles are used. The true mode trajectory is therefore discarded. By contrast, k-best enumeration is able to retain the true mode sequence, since the posterior distribution is concentrated in relatively few mode sequences.

6.1.2 Flat Posterior

We now consider single simulation results for a modified model, in which the posterior distribution is spread out across many distinct mode trajectories. The modified acrobot model has the same components as before, however the model parameters are different. The new transition priors are shown in Figure 26. Note that the probability of the acrobot catching a ball has changed from 0.05 to 0.5. Also, the mass of the ball is decreased to 1kg. The increased transition probability means that whenever $\theta_1 \geq 0.55$ the number of mode sequences with a high prior grows exponentially. Because the effect of a transition on observations is initially small, a large number of distinct trajectories will have high posterior probability.

The modified acrobot model is an example where the fair sampling of the Rao-Blackwellised Particle Filter can outperform the greedy search in the k-best filter. In the failure scenario, for example, where the actuator stops exerting torque, over a small number of time steps the difference between the failure dynamics and the nominal dynamics is very subtle. The

7. In the failure scenario the delay is greater since the transition prior to the `actuator=failed` mode is far lower than for the transition to the `has-ball=no` mode.

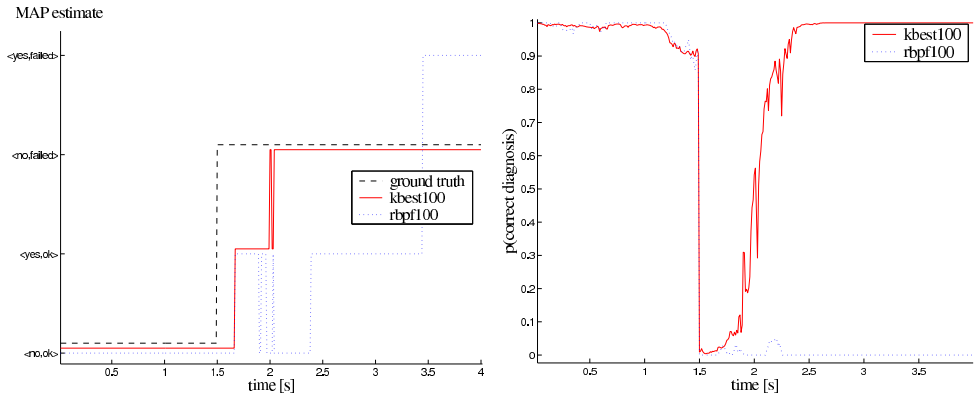


Figure 25: A single run for the actuator failure scenario with concentrated posterior. Left: Maximum a posteriori (MAP) estimate computed by the RBPF and the k-best filter. Right: probability of the correct diagnosis $\langle$`ball=no`, `actuator=failed`$\rangle$ for $t \geq 1.5s$.

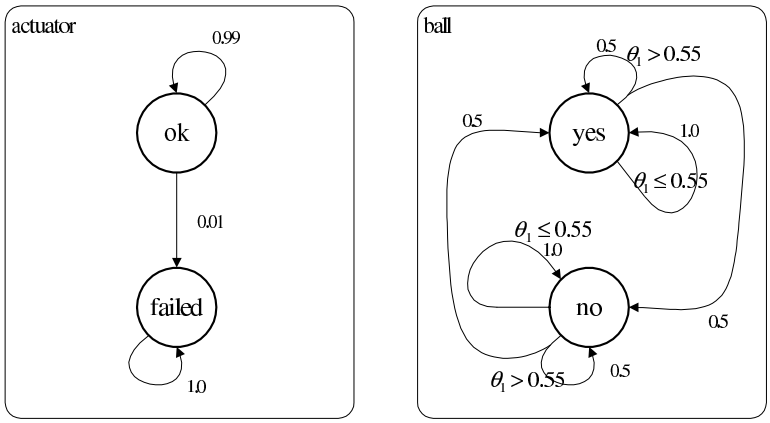


Figure 26: Probabilistic hybrid automata for the body and actuator of the modified acrobot example

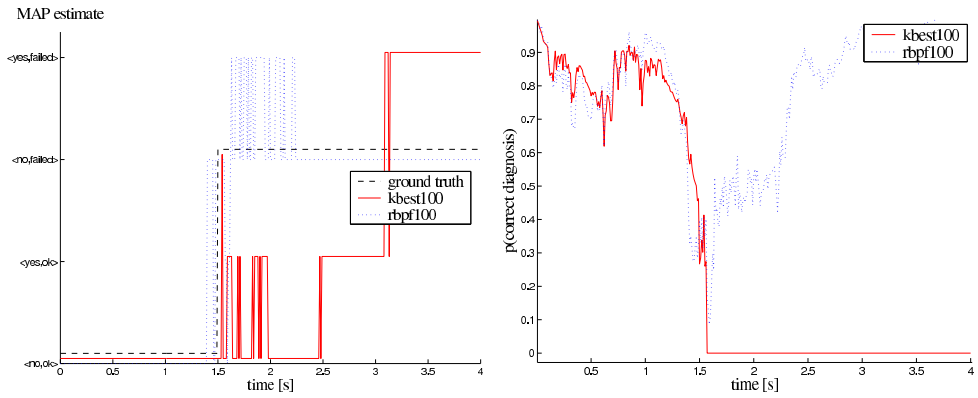


Figure 27: A single run for the actuator failure scenario with flat posterior. Left: Maximum a posteriori (MAP) estimate computed by the RBPF and the k-best filter. Right: probability of the correct diagnosis $\langle$`ball=no`, `actuator=failed`$\rangle$ for $t \geq 1.5s$.

k-best filter tracks the hypotheses in strictly best-first order and in this scenario there are a very large number of distinct hypotheses with high posterior likelihoods. If k is too small, *no* sequences with transitions to the `actuator=failed` mode will be tracked. RBPF, on the other hand, does not suffer from this problem since transitions are *sampled* stochastically. This exploration method can stochastically choose the correct mode sequence even if it is not strictly among the k best hypotheses, and hence will be more successful at tracking a system such as this.

Figure 27 shows the results for a single hybrid estimation run on the failure scenario, this time with the flat posterior distribution. The k-best enumeration approach fails to diagnose the fault correctly, The stochastic sampling approach of Rao-Blackwellised Particle Filtering, on the other hand, samples the failure transition fairly. When 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. With a flat posterior therefore, RBPF clearly outperforms k-best enumeration.

Figure 28 shows the continuous tracking of θ_1 for both k-best and RBPF algorithms. The RBPF has much more accurate continuous tracking than the k-best algorithm for this example as a direct result of its lower fraction of diagnostic errors. The state variable θ_1 is not observed directly, hence without an accurate estimate of the discrete mode sequence, the continuous dynamics of the system are unknown. This leads to large estimation errors in the continuous state.

These experimental results, therefore, provide empirical validation of the insight described in Section 5.2. While these are examples of single simulation runs only, in Section 6.2 we carry out an extensive performance comparison that confirms this result and provides empirical motivation for a mixed method that balances greedy and stochastic approaches to hybrid estimation.

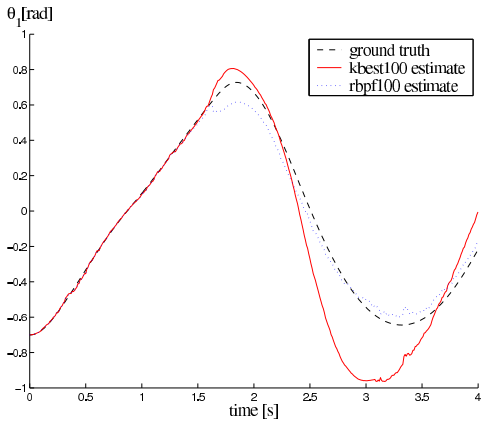


Figure 28: Continuous tracking of θ_1 with the failure scenario for a flat posterior

6.2 Performance Comparison

In this section we carry out a detailed empirical comparison of the performance of k-best enumeration and the new Rao-Blackwellised Particle Filter for CPHA. We show that the new mixed method is significantly more robust than either method alone.

6.2.1 Performance metrics

One of the biggest obstacles to evaluating the performance of hybrid state estimation algorithms is that inference with hybrid models is, in general, NP-hard (Lerner & Parr, 2001), and it is very difficult to obtain the true posterior distribution $p(\mathbf{x}_{c,t}, \mathbf{x}_{d,t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. Sometimes, this distribution can be approximated by a particle filter with a large number of samples; however, the accuracy of such approximations may not be bounded tightly enough.

Instead, we use the following two metrics for a given algorithm with a fixed number of tracked sequences:

1. The percentage of the diagnostic faults, defined as $\frac{\#\ of\ wrong\ diagnoses}{\#\ time\ steps}$. Wrong diagnoses are defined as MAP estimates of the discrete state at the fringe that are not the same as the ground truth.

2. The mean square estimation error 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 mode estimate, and $\mathbf{x}_{c,t}$ is the continuous state ground truth. This measure is averaged over all time steps and experiments.

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.

6.2.2 Concentrated Posterior

Figures 29 through 31 show the percentage of diagnostic errors and the mean square tracking error for the three scenarios considered for the acrobot model with concentrated posterior.

Figure 30 shows that for the ball scenario, the k-best algorithm makes almost no diagnostic errors. The Rao-Blackwellised Particle Filter on the other hand, makes more than 40 per cent diagnostic errors on average for small k; this decays to a minimum value as the number of tracked sequences increases.[8] In addition, there is far greater variance in the case of the RBPF; in an application such as fault diagnosis, this is particularly undesirable, since reliable performance is essential.

The Rao-Blackwellised Particle Filter gives a higher number of diagnostic errors because the approach is inherently stochastic; hence there is no guarantee that the correct mode transition will be sampled. When the failure transition is not sampled close to where it occurred in reality, the diagnostic error persists. Hence the fraction of diagnostic errors is greater, and has a far higher variance, than for the k-best approach.

The k-best filter, on the other hand, tracks k *distinct* sequences and calculates the posterior likelihood of these sequences exactly. When the posterior distribution is concentrated in relatively few mode sequences, the k-best filter will track the sequence corresponding to the ground truth as well as alternative sequences that are initially more likely, due to low transition prior. Hence the k-best filter does not make as many diagnostic errors as the Rao-Blackwellised Particle Filter with the ball scenario. Figure 30 shows that the mixed method performs almost as well as k-best enumeration, with a similarly small variance in the results.

Figure 31 shows a somewhat different trend for the failure scenario. For small k, both RBPF and k-best have a high proportion of diagnostic errors; for $k > 20$, however, the k-best algorithm outperforms the Rao-Blackwellised Particle Filter as was the case with the ball scenario. Also, whereas the performance of the RBPF improves gradually as k increases, the k-best algorithm shows a large shift in performance between $k = 20$ and $k = 50$. In this case the mixed method behaves in a similar manner to k-best enumeration, except that the shift in performance occurs at approximately double the number of tracked sequences. This is because only half of the sequences are being evolved greedily.

Finally, the results show that for a given algorithm, the failure scenario requires a far higher number of tracked sequences to achieve the same diagnostic accuracy. This is because the failure transition has an extremely low prior probability; hence a large number of sequences must be tracked for the true sequence to be included for long enough for the observation probability to dominate the low transition prior.

These results are the same when we compare the performance with respect to the run time of the algorithms. Figure 32 shows this comparison. We found that the run times of the three algorithms were different by a factor less than 1.5. This has a negligible effect on the log scales against which we compare the performance of the two algorithms.

8. This minimum value is non-zero, because for a small fraction of the time steps immediately after the true transition, the MAP diagnosis is not the same as the ground truth

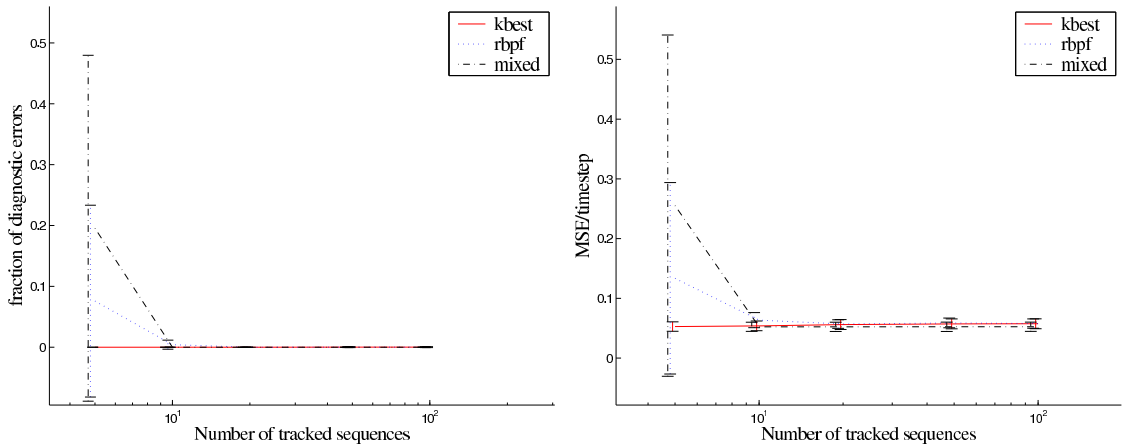


Figure 29: Performance for the nominal scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

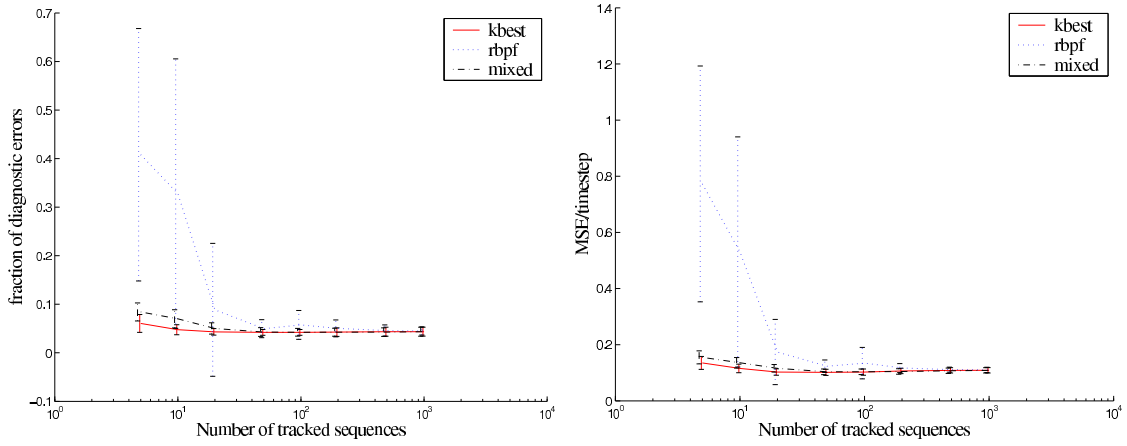


Figure 30: Performance for the ball scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

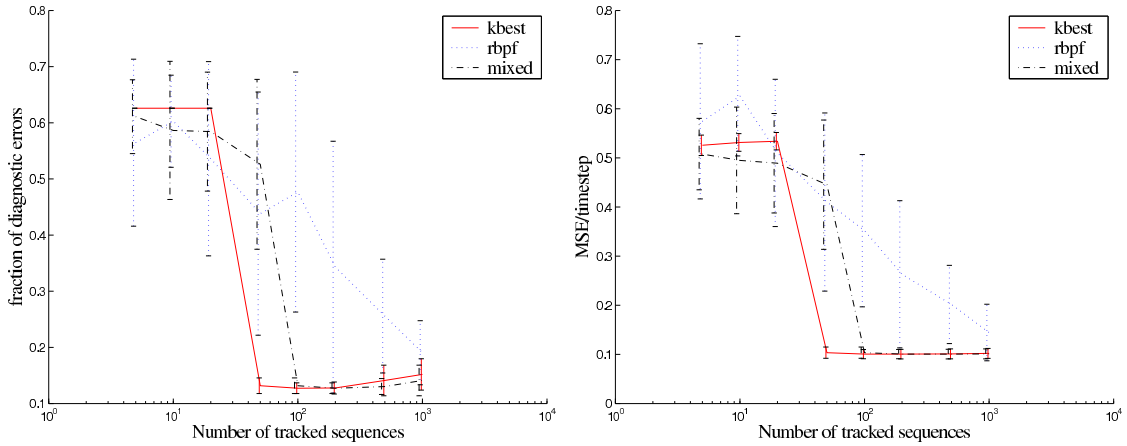


Figure 31: Performance for the actuator failure scenario with concentrated posterior. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state.

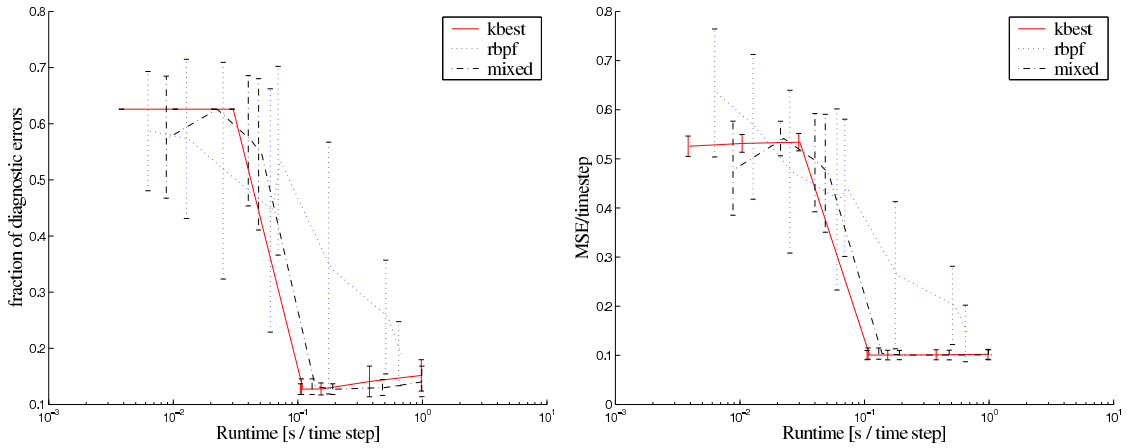


Figure 32: Performance for the failure scenario for concentrated posterior with respect to run time. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

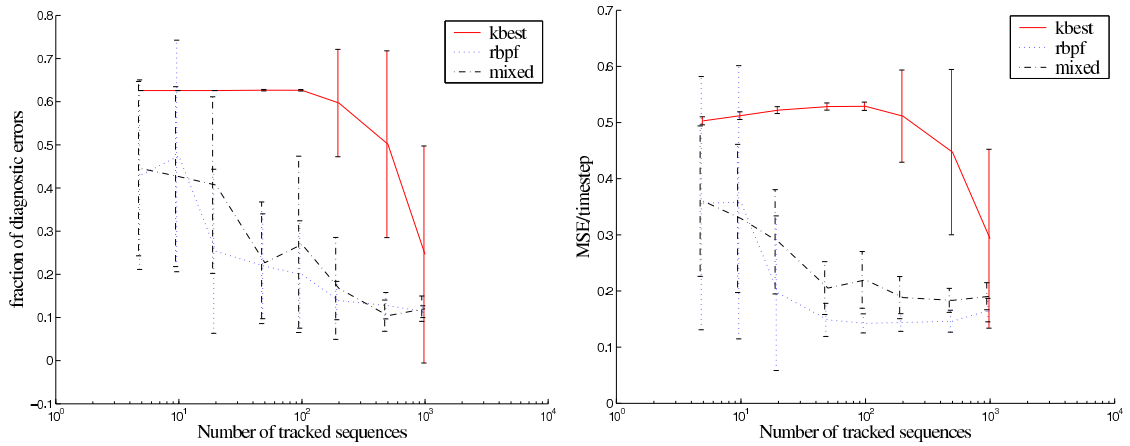


Figure 33: Performance for the failure scenario with flat posterior. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

6.3 Acrobot Model with Flat Posterior

We now compare the performance of the k-best algorithm and the RBPF when estimating the hybrid state of the modified acrobot model, which has a flat posterior distribution. Figure 33 shows the performance of the k-best and particle filtering algorithms for the failure scenario. 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, while the RBPF samples the actuator failure transition fairly. The mixed method exploits this stochastic approach, performing only slightly worse than the RBPF alone.

6.4 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. By contrast, the performance of the Rao-Blackwellised Particle Filter converges less quickly than the k-best filter to a low fraction of diagnostic errors (See Figure 31). Since the RBPF approximates the true posterior and duplicates high likelihood hypotheses, increasing the number of particles simply makes the approximation closer to the true posterior. Hence there is a gradual convergence and not the phase shift seen for k-best.

The critical value of k that greedy enumeration needs to track in order to perform well, depends on the concentration of the posterior distribution. If the posterior distribution is concentrated in a relatively small number of distinct sequences, the k-best method will perform well even for small k. In these cases, k-best enumeration typically outperforms RBPF, since it does not duplicate hypotheses, leading to unnecessary approximation. For a flat posterior, on the other hand, 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.

We have shown, empirically, that 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, by balancing greedy and stochastic search, the new method is much more robust to the choice of model parameters than the RBPF and k-best individually.

7. Related work

Several algorithms have addressed the problem of the exponential growth of Gaussian mixtures. One class of solutions are multiple-model estimation schemes, which maintain a pre-determined number of mode sequences. These include the generalized pseudo-Bayesian algorithm (GPB) (Ackerson & Fu, 1970), the Detection/Estimation Algorithm (DEA) (Tugnait, 1982), the Interacting Multiple Model (IMM) algorithm (Blom & Bar-Shalom, 1988), and residual correlation Kalman filter bank (Hanlon & Maybeck, 2000). All of these techniques have a fixed-time, deterministic strategy for pruning discrete mode sequences.

More recently, Lerner et al. (2000) proposed a k-best filtering solution for SLDS models. In addition to pruning, their algorithm implements several techniques not present in our algorithm, including collapsing of the mode sequences, smoothing, and weak decomposition. This approach was later extended in (Lerner, 2002a), to the setting of hybrid dynamic Bayesian networks with SoftMax transitions, using numerical integration techniques instead of the Kalman Filter. Similar to ours, their algorithm provides an any-time solution to the hybrid state estimation problem.

In the particle filtering community, several papers (Avitzour, 1995; Kitagawa, 1996) have proposed using the bootstrap particle filter to perform state estimation in hybrid models. An early application of the Rao-Blackwellisation method to reducing the variance of sampling in SLDS models was introduced by Akashi and Kumamoto (1977). Their algorithm, named the Random Sampling Algorithm (RSA), sampled the sequences of mode assignments using the distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. Metropolis and Ulam (1949) and Gordon, Salmond, and Smith (1993) introduced the Selection Step, which is crucial for the convergence of sequential Monte Carlo methods and framed the problem in the general particle filtering framework. In addition, they proved several properties regarding the convergence and variance reduction of Rao-Blackwellisation schemes. Doucet, Freitas, and Gordon (2001a) further extended this work and described an algorithm for fixed-lag smoothing with MCMC steps. Finally, Morales-Menéndez et al. (2002) introduced a procedure, called one-step look-ahead, which computes the total probability for the sequences stemming from a given sample and moves the selection step before the importance sampling step, at the cost of evaluating the Kalman Filter residual for all successor modes. All of these techniques were designed for linear switching models without autonomous transitions.

Concurrently with our work published in (Funiak & Williams, 2003), the combination of a Rao-Blackwellised particle filter with an Unscented Kalman Filter for fault detection was proposed independently by Hutter and Dearden (2003). In addition, they incorporated the look-ahead step proposed by Morales-Menéndez et al. (2002).

Two complementary approaches for improving the performance of particle filters were proposed by Thrun, Langford, and Verma (2001) and Verma et al. (2003). The first one,

the Risk-sensitive Particle Filter, incorporates a model of cost into the sampling process. The cost is implemented automatically using an MDP value function tracking. The second approach improves the performance of particle filtering by automatically choosing an appropriate level of abstraction in a multiple-resolution hybrid model. Maintaining samples at a lower resolution prevents hypotheses from being eliminated due to a lack of samples.

8. Conclusion

In this paper, we investigated the problem of estimating the state of a system represented with probabilistic hybrid models. We presented an efficient Rao-Blackwellised particle filtering algorithm, developed in Section 3.7, that handles the autonomous mode transitions, concurrency, and nonlinearities present in Concurrent Probabilistic Hybrid Automata (CPHA). Through the technique of Rao-Blackwellised particle filtering, our algorithm significantly reduces the dimensionality of the sampled space and improves the performance of particle filtering. The key insight to addressing the autonomous transitions was to reuse the continuous estimates associated with the tracked mode sequences. We extended the class of autonomous mode transitions that can be handled by both k-best enumeration and Rao-Blackwellised Particle Filtering for CPHA to multivariable linear transition guards, in the case of piecewise constant transition distributions, and to polynomial transition distributions of arbitrary order, in the case of single variables.

This new algorithm allows a unified treatment of approximate hybrid estimation through both Rao-Blackwellised particle filtering and k-best filtering(Hofbaur & Williams, 2002a, 2002b). A simulated acrobatic robot was used to develop insight about the relative performance of the two approaches. The results showed that when the posterior is concentrated in a few nominal or single-fault sequences, the k-best filter is a clear winner. However, when the distribution over mode trajectories is relatively flat, the trajectory corresponding to the correct diagnosis may be left out of the leading set of mode sequences. In such situations, the random sampling approach of the Rao-Blackwellised particle filter is more successful.

We therefore developed a new algorithm that combines greedy and stochastic search by tracking two sets of mode 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. Simulations showed that the algorithm is more robust than k-best and RBPF taken individually, with only a minor performance penalty.

References

Ackerson, G., & Fu, K. (1970). On state estimation in switching environments. *IEEE Transactions on Automatic Control*, *15*, 10–17.

Akashi, H., & Kumamoto, H. (1977). Random sampling approach to state estimation in switching environments. *Automatica*, *13*, 429–434.

Anderson, B., & Moore, J. (1979). *Optimal Filtering*. Information and System Sciences Series. Prentice Hall.

Avitzour, D. (1995). A stochastic simulation Bayesian approach to multitarget tracking. *IEE Proceedings on Radar Sonar and Navigation*, *142*(2), 41–44.

Blom, H., & Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, *33*.

Buchberger, B., & Winkler, F. (Eds.). (1998). *Gröbner Bases and Applications*. Cambridge Univ. Press.

Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, *83*(1), 81–94.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150.

Dearden, R., & Clancy, D. (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., de Freitas, J., Murphy, K., & Russell, S. (2000). Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

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

Doucet, A., Gordon, N. J., & Krishnamurthy, V. (2001b). Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, *49*(3), 613–624.

Doucet, A. (1998). On sequential simulation-based methods for bayesian filtering. Tech. rep. CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering.

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

Funiak, S., & Williams, B. (2003). Multi-modal particle filtering for hybrid systems with autonomous mode transitions. In *Proceedings of SafeProcess 2003 (also published in DX-2003*.

Genz, A., & Kwong, K.-S. (2000). Numerical evaluation of singular multivariate normal distributions. *J. Stat. Comp. Simul*, *68*(1-21).

Gomes, C. P., Selman, B., & Kautz, H. (1998). Boosting combinatorial search through randomization. In *Proceedings of the 15th National Conference on Artificial Intelligence*.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-GaussianBayesian state estimation. *IEE Proceedings - F*, *140*(2), 107–113.

Hanford, A. (2002). Advanced life support baseline values and assumptions document. Tech. rep. CTSD-ADV-484, NASA, Johnson Space Center, Houston.

Hanlon, P., & Maybeck, P. (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., & Williams, B. C. (2002a). Mode estimation of probabilistic hybrid systems. In *Intl. Conf. on Hybrid Systems: Computation and Control*.

Hofbaur, M. W., & Williams, B. C. (2002b). Hybrid diagnosis with unknown behavioral modes. In *Proceedings of the 13th International Workshop on Principles of Diagnosis (DX02)*, pp. 97–105.

Hofbaur, M. W., & Williams, B. C. (2004). Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*.

Hofbaur, M. (2003). *Hybrid Estimation and its Role in Automation*. Habilitationsschrift, Faculty of Electrical Engineering, Graz University of Technology, Austria.

Hutter, F., & Dearden, R. (2003). The gaussian particle filter for diagnosis of non-linear systems. In *Proceedings of the 14th International Conference on Principles of Diagnosis(DX'03)*, pp. 65–70, Washington, DC, USA.

iRobot (2005). Robots for the real world. http://www.irobot.com.

JIMO (2005). Jupiter icy moons orbiter. http://www.jpl.nasa.gov/jimo.

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

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th Symposium on Aerospace/Defense Sensing, Simulation and Controls*.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Comptuational and Graphical Statistics*, *5*, 1–25.

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

Lerner, U. (2002a). *Hybrid Bayesian Networks for Reasoning About Complex Systems*. Ph.D. thesis, Stanford University.

Lerner, U. (2002b). *Hybrid Bayesian Networks for Reasoning About Complex Systems*. Ph.D. thesis, Stanford University.

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

Lerner, U., & Parr, R. (2001). Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pp. 310–318, Seattle, Washington.

Martin, O. B., Williams, B. C., & Ingham, M. D. (2005). Diagnosis as approximate belief state enumeration for probabilistic concurrent constraint automata. In *20th National Conference on Artificial Intelligence*.

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

Metropolis, N., & Ulam, S. (1949). The monte carlo method. *American Statistical Association*, *44*, 335–341.

Mikaelian, T., Williams, B. C., & Sachenbacher, M. (2005). Monitoring and diagnosis of systems with software-extended behavior. In *Proceedings of 20th National Conference on Artificial Intelligence*, pp. 327–333.

Montemerlo, M., Pineau, J., Roy, N., Thrun, S., & Verma, V. (2002). Experiences with a mobile robotic guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada. AAAI.

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

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

Narasimhan, S., Biswas, G., Karasai, G., & Zhao, F. (2000). Building observers to address fault isolation and control problems in hybrid dynamic systems. In *Proceedings of the IEEE Internation Conference on Systems, Man, and Cybernetics (SMC 2000)*, pp. 2393–2398.

Nayak, P. (1995). *Automated Modelling of Physical Systems*. Lecture Notes in Artificial Intelligence. Springer.

Nayak, P., & Williams, B. C. (1997). Fast context switching in real-time propositional reasoning. In *Proceedings of AAAI-97*, pp. 50–56.

Ng, B., Peshkin, L., & Pfeffer, A. (2002). Factored particles for scalable monitoring. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*.

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

Pavlovic, V., Rehg, J., Cham, T.-J., & Murphy, K. (1999). A dynamic Bayesian networ approach to figure tracking using learned dynamic models. In *Proceedings of ICCV*.

Sachenbacher, M., & Williams, B. C. (2004). Diagnosis as semiring-based constraint optimization. In *Proc. 16th European Conference on Artificial Intelligence*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Thrun, S., Langford, J., & Verma, V. (2001). Risk sensitive particle filters. In *Neural Information Processing Systems (NIPS)*.

Tugnait, J. (1982). Detection and estimation for abruptly changing systems. *Automatica*, *18*, 607–615.

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

Verma, V., Thrun, S., & Simmons, R. (2003). Variable resolution particle filter. In *In Proceedings of International Joint Conference on Artificial Intelligence*. AAAI.

Williams, B. C., Chung, S., & Gupta, V. (2001). Mode estimation of model-based programs: Monitoring systems with complex behavior. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Williams, B. C., & Nayak, P. (1996). A model-based approach to reactive self-configuring systems. In *Proceedings of AAAI-96*, pp. 971–978.

MSL (2005). Mars science laboratory. http://marsprogram.jpl.nasa.gov/missions/future/msl.html.