Gaussian Particle Filtering for Concurrent Hybrid Models with Autonomous Transitions

Stanislav Funiak and **Lars J. Blackmore** and **Brian C. Williams**
MIT Computer Science and Aritificial Intelligence Laboratory
32 Vassar Str.
Cambridge, MA 02139 USA

Abstract

Robotic and embedded systems have become increasingly pervasive in every-day applications, ranging from space probes and life support systems to autonomous rovers. In order to act robustly in the physical world, robotic systems must handle the uncertainty and partial observability inherent in most real-world situations. Probabilistic hybrid models are convenient modeling tool for many applications, including fault diagnosis and visual tracking. In probabilistic hybrid models, the hidden state is represented with discrete and continuous state variables that evolve probabilistically. The hidden state is observed indirectly, through noisy observations. A challenge is that real-world systems are non-linear, consist of a large collection of concurrently operating components, and exhibit autonomous mode transitions, that is, discrete state transitions that depend on the continuous dynamics.

In this paper, we introduce an efficient algorithm for hybrid state estimation that combines Rao-Blackwellised particle filtering with a Gaussian representation. Conceptually, our algorithm samples trajectories traced by the discrete variables over time and, for each trajectory, estimates the continuous state with a Kalman Filter. A key insight to handling the autonomous transitions is to reuse the continuous estimates in the importance sampling step. We extended the class of autonomous transitions that can be efficiently handled by Gaussian techniques and provide a detailed empirical evaluation of the algorithm on a dynamical system with four continuous state variables. Our results indicate that our algorithm is substantially more efficient than non-Rao-Blackwellised approaches. Though not as good as a k-best filter in nominal scenarios, our algorithm outperforms a k-best filter when the correct diagnosis has too low a probability to be included in the leading set of trajectories. Through these accomplishments, the paper lays ground work for a unifying stochastic search algorithm that shares the benefits of both methods.

1. Introduction

Robotic and embedded systems have become increasingly pervasive in a variety of applications. Space missions, such as Mars Science Laboratory (MSL) and the Jupiter Icy Moons Orbiter (JIMO), 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, to provide a renewable supply of oxygen, water, and food. Here on Earth, robotic assistants, such as CMU's Pearl and iRobot's Roomba, 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 this paper, we investigate the problem of estimating the state of systems with probabilistic hybrid models. 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 thus thought of as extensions of discrete models, such as hidden Markov models (Russell & Norvig, 2003) or dynamic Bayesian networks (Dean & Kanazawa, 1989), to continuous dynamical 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 often provide an appropriate level of modeling abstraction when purely discrete, qualitative models are too coarse, while purely continuous, quantitative models are too fine-grained. 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. Probabilistic hybrid models can thus be viewed as a natural successor to discrete model-based diagnosis systems, such as Livingstone (Williams & Nayak, 1996) and Titan (Williams, Ingham, Chung, & Elliott, 2003).

State estimation techniques for probabilistic hybrid models have traditionally focused on a restricted subset of conditional linear Gaussian models, such as Switching Linear Dynamical Systems(Shumway & Stoffer, 1991; Kim, 1994), 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. The number of tracked estimates can be kept down to an acceptable level by using one of a variety of methods, including the well-known interactive multiple model (IMM) algorithm (Blom & Bar-Shalom, 1988), Rao-Blackwellised particle filtering (Akashi & Kumamoto, 1977; Doucet, 1998) and, more recently, efficient k-best filtering (Lerner, Parr, Koller, & Biswas, 2000; Hofbaur & Williams, 2002a). Systems with non-linear dynamics, such as a rover drive subsystem (Hutter & Dearden, 2003) can be handled by employing variations of the Kalman Filter, such as an Extended Kalman Filter (Anderson & Moore, 1979) or an Unscented Kalman Filter (Julier & Uhlmann, 1997).

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. Such transitions

are called *autonomous*[1], and are substantially more challenging to address. 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.

Two generalizations have recently been proposed that address the above deficiencies of switching linear dynamical systems: hybrid dynamic Bayesian networks (hybrid DBNs) and Concurrent Probabilistic Hybrid Automata (CPHA) (Hofbaur & Williams, 2002a; Hofbaur, 2003). Hybrid dynamic Bayesian networks provide a factored representation of the transition and observation distributions. In hybrid DBNs, 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. 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 nodel of them system.

Recent advances in k-best Gaussian filtering (Hofbaur & Williams, 2002a; Lerner, 2002) have shown great promise for hybrid state estimation. These methods represent the state as a mixture of Gaussians that are enumerated in the decreasing order of likelihood. Owing to their efficient representation and focused search, these methods have been succesfully applied to large systems with as many as 450,000 discrete states. Excessive focusing during search may, however, come at a price if the correct diagnosis is not among the leading set of hypotheses.

In this paper, we present an alternative solution for CPHA, based on the technique of Rao-Blackwellised particle filtering. The key principle of Rao-Blackwellised Particle Filtering is to decrease the amount of sampled space by estimating a tractable subspace with a closed-form solution. In the spirit of prior approaches to k-best filtering and Rao-Blackwellised particle filtering, our algorithm tracks the sequences of mode assignments and, for each sequence, estimates the state with an Extended Kalman Filter (Anderson & Moore, 1979) or an Unscented Kalman Filter (Julier & Uhlmann, 1997). By sampling the sequences of mode assignments, our algorithm is substantially mode efficient than traditional particle filters and improves upon the performance of a k-best filter in the cases when the posterior consists of a large number of nominal sequences.

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. Specifically, the transition probabilities are computed by integrating the Gaussian over the set, corresponding to each transition guard, similar to (Hofbaur & Williams, 2002a). We provide a derivation of the transition probabilities when transition guards are linear. In order to perform state estimation in multi-component systems, represented as a CPHA, we either evolve them according to their priors or compute the optimal proposal distributions for single-component transitions and then combine them as a proposal distribution for the overall model.

In this paper we demonstrate the algorithm on a highly nonlinear two-link dynamical system, shown in Figure 1, and compare it to the corresponding k-best filtering algorithm

1. In the terminology of hybrid Bayesian networks, these correspond to discrete nodes with continuous parents.

(Hofbaur & Williams, 2002a) on a number of characteristic scenarios. We show that although our algorithm is not as good as the k-best filter when most of the posterior is concentrated in a small number of hypotheses, it outperforms the k-best filter when the posterior is spread-out among a large number of similar nominal sequences that prevent the correct hypothesis from being considered by the k-best filter. Our results thus lay the ground work for a unifying approach, in which k-best filtering is interleaved with Gaussian particle filtering to improve upon the performance of both.

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 (Hofbaur & Williams, 2002a). In this section, we first define a simple hybrid model, called Switching Linear Dynamical Systems (SLDS) (Shumway & Stoffer, 1991; Kim, 1994), and briefly summarize an extension to this model, called Concurrent Probabilistic Hybrid Automata (Hofbaur & Williams, 2002a; Hofbaur, 2003) to large, multi-component systems. We then define the hybrid state estimation problem and lay out the basic techniques for addressing this problem.

2.1 Switching Linear Dynamical Systems

A switching linear dynamical system (SLDS) (Shumway & Stoffer, 1991; Kim, 1994), also known as a jump Markov linear Gaussian model, is a special form of hybrid model, in which the hidden state is represented by a finite Markov chain d, with transition distribution $p(d_t|d_{t-1})$, and a continuous vector $\mathbf{x}_c$. The continuous vector evolves linearly, according to mode-dependent system and input matrices $\mathbf{A}(d)$, $\mathbf{B}(d)$:

$$\mathbf{x}_{c,t} = \mathbf{A}(d)\mathbf{x}_{c,t-1} + \mathbf{B}(d)\mathbf{u}_{c,t-1} + \mathbf{v}_x(d). \tag{1}$$

The hidden state is observed indirectly, with additive white Gaussian noise:

$$\mathbf{y}_t = \mathbf{C}(d)\mathbf{x}_{c,t} + \mathbf{v}_y(d), \tag{2}$$

where $C(d)$ is a mode-dependent observation matrix and $\mathbf{v}_y(d)$ is a normally-distributed noise variable, with zero mean and a known covariance $\Lambda_{\mathbf{v}_y(d)}$.

2.2 Concurrent Probabilistic Hybrid Automata

Modeling with SLDS is often insufficient, because they only allow a single variable to represent the discrete mode of the system. This is inefficient for modeling large systems with many modes. In addition, it becomes cumbersome to specify the complete continuous model in the standard transition and observation function format. To address these drawbacks, we have previously developed Concurrent Probabilistic Hybrid Automata (CPHA) (Hofbaur & Williams, 2002a; Hofbaur, 2003), 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

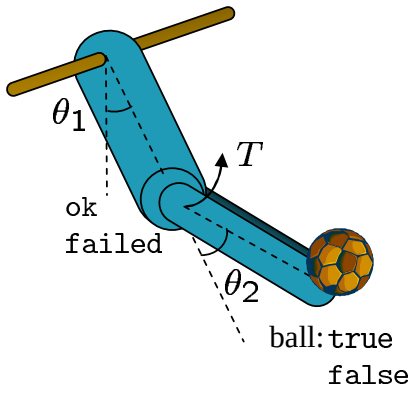

Figure 1: A two-link acrobatic robot.

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 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; Hofbaur, 2003):*

- $\mathbf{x}$ denotes the *hybrid state* of the automaton, which consists of discrete variables $\mathbf{x}_d \in \mathcal{X}_d$ and continuous state variables $\mathbf{x}_c \in \mathbb{R}^{n_x}$.[2]
- $\mathbf{w}$ 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 $\mathbf{v}_c \in \mathbb{R}^{n_v}$.
- $F : \mathcal{X}_d \rightarrow 2^{\mathcal{F}_{DE} \cup \mathcal{F}_{AE}}$ specifies the *continuous evolution* of the automaton as set of first-order discrete-time difference equations $F_{DE} \subset \mathcal{F}_{DE}$ and algebraic equations $F_{AE} \subset \mathcal{F}_{AE}$ 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 modes $\mathcal{X}_d$ and their associated guard conditions $c_i \in \mathcal{C}$ over $\mathbf{x}_c \cup \mathbf{u}_d$, which form a partition of $\mathbb{R}^{n_x} \times \mathcal{U}_d$.
- X_0 is a conditional Gaussian distribution for the initial state of the automaton.

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 ($\theta_1 > 0$). The goal is to estimate its continuous state and whether or not it holds a ball at any given time step, by observing the angle between the two links.

A PHA model for this system has one discrete, binary variable and four continuous real variables (see Figure 2). The continuous dynamics in each mode are derived using

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

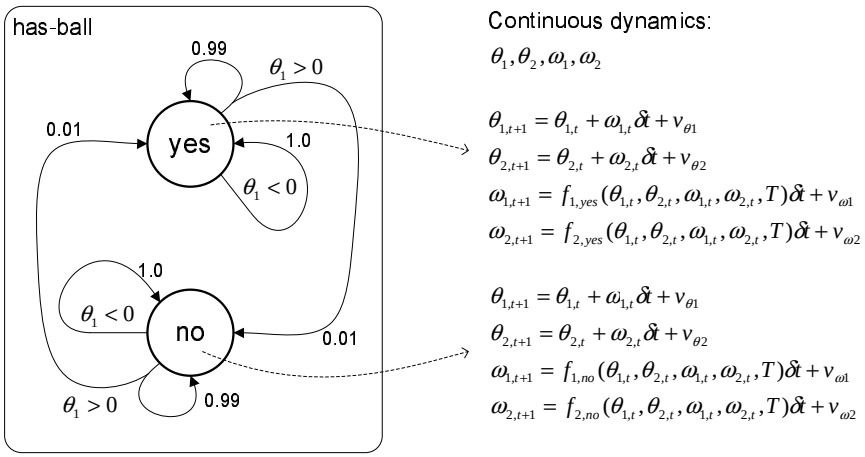

Figure 2: A PHA for a two-link acrobot system. Left: transition model for the discrete state of the system. Right: evolution of the automaton's continuous state, one set of equations for each mode.

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 $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1}, \mathbf{u}_{d,t})$ to be p_τ in the regions satisfied by the guard c. For example, when `has-ball` = `no`, the probability of transitioning to mode `yes` is 0.01 whenever $\theta_1 > 0$ and 0 otherwise. The transition function can thus specify conditional distributions $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1})$, which changes sharply with the continuous state $\mathbf{x}_{c,t-1}$. This can be viewed as a limiting case of the SoftMax distribution (Koller, Lerner, & Angelov, 1999).

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:

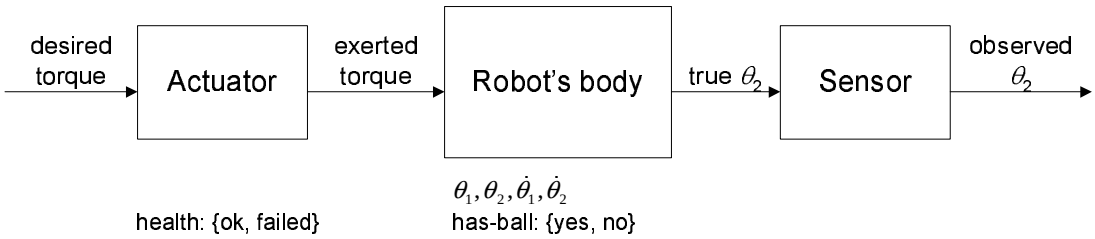

Figure 3: A composed model for the acrobatic robot in Figure 1. Each component is modeled with one Probabilistic Hybrid Automaton. The component automata are shown in rectangles, with their state variables shown beneath.

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*

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

As before, the overall continuous evolution of the composed model varies in each mode, and is determined by taking the union of algebraic and difference equations for each component PHA. The discrete transitions for CPHA are defined independently for each component, conditioned on the continuous state. For example, the transition probability

$$p(\texttt{actuator}_t = \texttt{failed}, \texttt{ball}_t = \texttt{no} | \texttt{actuator}_{t-1} = \texttt{ok}, \texttt{ball}_{t-1} = \texttt{no}, \mathbf{x}_{c,t-1}) \tag{3}$$

is defined as a product of independent transitions

$$\begin{aligned}
&p(\texttt{actuator}_t = \texttt{failed} \mid \texttt{actuator}_{t-1} = \texttt{ok}, \mathbf{x}_{c,t-1}) \\
&\quad p(\texttt{ball}_t = \texttt{no} \mid \texttt{ball}_{t-1} = \texttt{no}, \mathbf{x}_{c,t-1})
\end{aligned} \tag{4}$$

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. PHA are closed under composition: composing two compatible automata results in a valid PHA. Nevertheless, 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-automon model.

One important aspect of PHA and CPHA is that, unlike in hybrid Bayesian networks(Lauritzen, 1992; Cowell, Dawid, Lauritzen, & Spiegelhalter, 1999), the causal relationships among continuous variables are not explicitly represented in the model. In

particular, the causal relationship among continuous variables can change, as function of the discrete state of the system. The model only specifies the continuous input variables $\mathbf{u}_c \subseteq \mathbf{w}_c$ that drive the system and the observed variables $\mathbf{y}_c \subseteq \mathbf{w}_c$, along with the distributions of the Gaussian noise variables $\mathbf{v}_c$. The equations $F(\mathbf{d})$ are then solved using Groebner bases (Buchberger & Winkler, 1998) and causal analysis (Nayak, 1995) into the standard form

$$\begin{aligned} \mathbf{x}_{c,t} &= \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{c,t-1}, \mathbf{v}_{x,t-1}; \mathbf{x}_{d,t}) \\ \mathbf{y}_{c,t} &= \mathbf{g}(\mathbf{y}_{c,t-1}, \mathbf{u}_{c,t-1}, \mathbf{v}_{y,t-1}; \mathbf{x}_{d,t}). \end{aligned} \tag{5}$$

Typically, we further assume that the noise is additive, that is,

$$\begin{aligned} \mathbf{x}_{c,t} &= \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{c,t-1}; \mathbf{x}_{d,t}) + \mathbf{v}_{x,t-1} \\ \mathbf{y}_{c,t} &= \mathbf{g}(\mathbf{y}_{c,t-1}, \mathbf{u}_{c,t-1}; \mathbf{x}_{d,t}) + \mathbf{v}_{y,t-1}. \end{aligned} \tag{6}$$

However, using the Unscented Kalman Filter (Wan & van der Merwe, 2000), it would be possible to address the more general setting of Equation 5.

2.3 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 $\mathbf{y}_0, \ldots, \mathbf{y}_t$, estimate the hybrid state $\langle \mathbf{x}_{d,t}, \mathbf{x}_{c,t} \rangle$ at time 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, 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.

2.4 Particle Filtering

Given a SLDS or a PHA model of a system, particle filters approximate the posterior of 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

$$p_{\bar{N}}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x). \tag{7}$$

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 4. In the first, initialization step, the algorithm samples the initial distribution $p(\mathbf{x}_0)$; thus, effectively approximating the

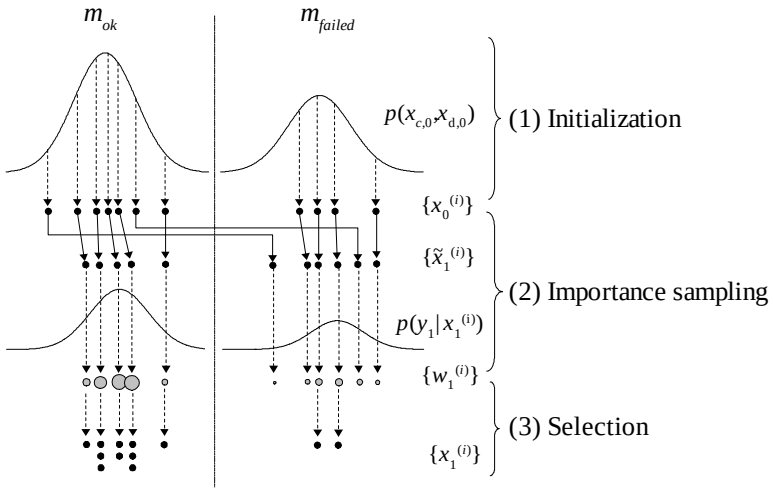

Figure 4: The three steps of a simple particle filter for a PHA model with one discrete and one continuous variable.

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 the an appropriately-chosen *proposal distribution*, and assigned 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-Blackwellisation (Akashi & Kumamoto, 1977; Casella & Robert, 1996; Doucet, 1998). This method is based on a fundamental observation that if the model has a tractable substructure, we may be able to factor it out with an efficient solution and only sample the remaining variables. In this manner, fewer samples are needed to obtain a given accuracy of the estimate.

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

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) &= p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \\ &= p(\mathbf{s}_{0:t}|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) p(\mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \end{aligned} \quad (8)$$

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.

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) 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 $\{\mathbf{r}_{0:t}^{(i)}\}$
   (c) 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)

Figure 5: Generic RBPF algorithm. (Murphy & Russell, 2001)

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 the distribution. The posterior is thus approximated as a mixture of the distributions $\alpha_t^{(i)}$ at the sampled points $\mathbf{r}_{0:t}^{(i)}$:

$$p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \approx \sum_1^N \alpha_t(i) \delta_{\mathbf{r}_{0:t}^{(i)}}(\mathbf{r}_{0:t}). \tag{9}$$

A generic RBPF method is outlined in Figure 5, and, except for the initialization and the addition of the exact step, it is identical to the particle filter, illustrated in Figure 4. Under weak assumptions, the Rao-Blackwellised estimate converges to the estimated value as $N \to +\infty$, with a variance smaller than the generic particle filtering method (Doucet, de Freitas, Murphy, & Russell, 2000). Therefore, at least based on a fixed number of samples, it is beneficial to sample as small a subset of the state space as possible. In practice, the run-time performance of the filter will depend on the relative cost of the exact update for $\alpha_t^{(i)}$.

2.5 Gaussian filtering in SLDS models

SLDS models have attractive properties that make them particularly amenable to Rao-Blackwellisation. If we take an arbitrary (but fixed) assignment $d_{0:t} \triangleq d_0, d_1, \ldots, d_t$ to the mode variables $\mathbf{x}_{d,0:t}$, then the initial distribution $p(\mathbf{x}_{c,0})$, the system matrices $\mathbf{A}, \mathbf{B}$, the observation matrices $\mathbf{C}, \mathbf{D}$, and the noise models will each be fixed for all $t' = 1, \ldots, t$. This means that once we fix the mode variables $\mathbf{x}_{d,0:t}$, we can construct an analytical estimate of the continuous state of the system up to time t (Akashi & Kumamoto, 1977) with a Kalman Filter (Anderson & Moore, 1979), see Figure 6.

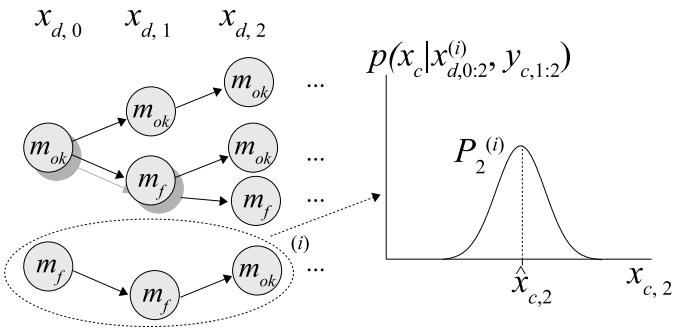

Figure 6: Structure in switching linear dynamical systems. Once we fix the mode up to time t, we can estimate the continuous state at time t analytically.

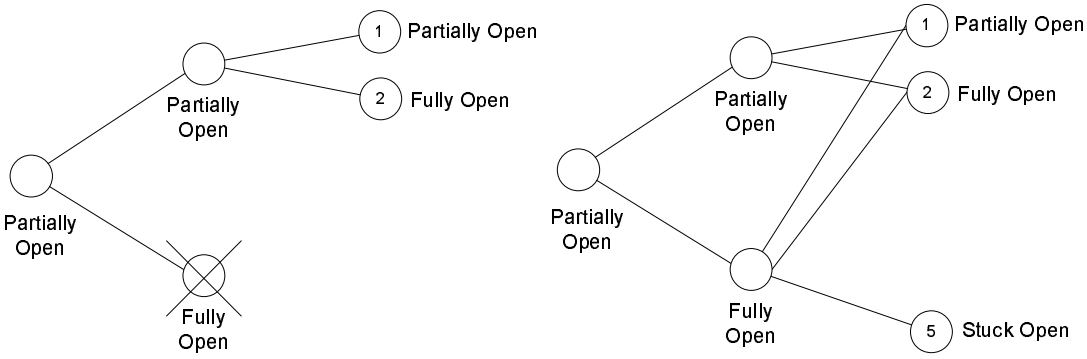

Figure 7: Pruning (left) and collapsing strategies (right).

Rao-Blackwellised particle filtering in SLDS models is a special case of a more general approach, which we call *Gaussian filtering* (Ackerson & Fu, 1970; Tugnait, 1982; Blom & Bar-Shalom, 1988; Hanlon & Maybeck, 2000; Lerner et al., 2000; Hofbaur & Williams, 2002a; Lerner, 2002; Funiak & Williams, 2003; Hutter & Dearden, 2003). The common property of Gaussian filtering methods is that they represent the posterior distribution compactly as a mixture of Gaussians. As in the RBPF for SLDS models, each Gaussian has an associated sequence or sequences of discrete mode assignments and represents the distribution of the continuous state, conditioned on this sequence.

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, such as fault diagnosis, where the prior distributions are heavily biased towards nominal transitions, efficient inference is possible, by employing two strategies: *pruning* (branching) and *collapsing* (merging) (see Figure 7). 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 et al., 2000).

In recent years, k-best filtering methods (Lerner et al., 2000; Hofbaur & Williams, 2002a; Lerner, 2002) have shown great promise. 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. One approach is to expand all the successors of all sequences and compute their transition probabilities 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 expansions as a search and solve it using a combination of branch and bound and A* algorithms (Hofbaur & Williams, 2002a).

3. Gaussian Particle Filtering for PHA

The key contribution of this chapter is an approximate Rao-Blackwellised particle filtering (RBPF) algorithm for PHA that handles the nonlinearities and autonomous transitions, that is, mode transitions dependent on the continuous state, present in PHA models. In the spirit of prior approaches to RBPF (Akashi & Kumamoto, 1977; Morales-Menéndez, de Freitas, & Poole, 2002) and k-best filtering (Lerner et al., 2000; Hofbaur & Williams, 2002a), our algorithm samples the mode sequences and, conditioned on each sampled sequence, estimates the associated continuous state with an Extended (Anderson & Moore, 1979) or an Unscented Kalman Filter (Julier & Uhlmann, 1997). Our key insight to addressing autonomous transitions for PHA is to reuse the continuous estimates in the importance sampling step of the filter. We extend the class of autonomous transitions that can be addressed efficiently. In addition, we demonstrate how the algorithm bridges the prior work in Rao-Blackwellised particle filtering and hybrid model-based diagnosis, laying the foundation for a principled unification of RBPF and k-best PHA filtering methods.

3.1 Tractable substructure in PHA

Recall that for SLDS and PHA models, this structure typically takes the form of the continuous state, conditioned on sequences of mode assignments (Akashi & Kumamoto, 1977; Hofbaur & Williams, 2002a). For a PHA, the continuous behavior of the system may change at each time step, in a similar way as it does in SLDS models. However, PHA pose two additional challenges to continuous state estimation: non-linearities and autonomous mode transitions. These two challenges translate to two approximations:

1. When nonlinear functions are used in the transition or an observation function, the posterior is typically nonlinear and non-Gaussian. This means that the continuous state tracking will typically incur error whenever the system is propagated through such equations.

2. When the PHA model has autonomous transitions, the posterior will be biased immediately after the transition towards the regions of those guards c_j, which have a higher associated transition probability $p_{\tau_j}(\mathbf{x}_{d,t}^{(i)})$.

The first approximation will be accurate to the first degree if an Extended Kalman Filter is used, or to the second degree if the Unscented Kalman Filter is used. In order to understand the second approximation, consider the example in Figure 8. This figure shows

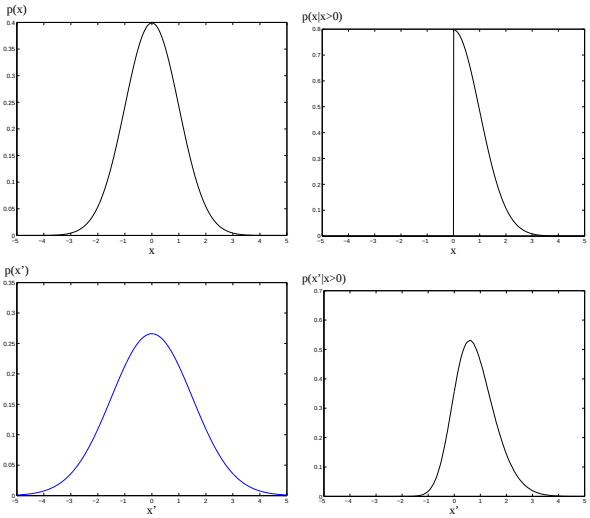

Figure 8: The top two graphs show a Gaussian distribution $p(x)$ (left) and a graph of this distribution when it is propagated through a constraint $x > 0$ (right). The bottom left graph shows the distribution of x when x is propagated through the model $x' = x + \mathcal{N}(0, 1)$ but ignores the constraint, while the bottom right graph shows the true distribution when the constraint is accounted for (obtained by importance sampling with a large number of samples).

the distribution $N(0, 1)$ of variable x, when it is first propagated through a constraint $x > 0$ (upper right-hand corner) and then is evolved according to the continuous model $x' = x + \mathcal{N}(0, 1)$ (lower right-hand corner). We see that by conditioning the variable on the event $x > 0$, its distribution is slightly skewed to the right and has a smaller variance. Nevertheless, the disitribution is still approximated well by a Gaussian, due to the normal noise added after the distribution is propagated through the constraint.

In our current algorithm, we make no special arrangements in order to account for the bias introduced by autonomous transitions. It may be possible, however, to compute the true mean and variance of the distribution after it has been propagated through the continuous constraint by numerical methods.

3.2 Overview of the algorithm

Since we can approximate the posterior distribution $p(\mathbf{x}_{c,t}|\mathbf{x}_{d,0:t}, \mathbf{u}_{0:t})$ efficiently in an analytical form, we can apply Rao-Blackwellisation to the hybrid estimation problem by taking $\mathbf{r} = \mathbf{x}_d$ and $\mathbf{s} = \mathbf{x}_c$. In other words, we sample the mode sequences $\mathbf{x}^{(i)}_{d,0:t}$ with a particle filter and, for each sampled sequence $\mathbf{x}^{(i)}_{d,0:t}$, we estimate the continuous state with a Kalman Filter. The result of Kalman Filtering for each sampled sequence $\mathbf{x}^{(i)}_{d,0:t}$ is the estimated mean $\hat{\mathbf{x}}^{(i)}_{c,t}$ and the error covariance matrix $\mathbf{P}^{(i)}_t$. The samples $\mathbf{x}^{(i)}_{d,0:t}$ 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 continu-

ous estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle$ serves as a Gaussian approximation of the conditional distribution $p(\mathbf{x}_{c,t} | \mathbf{x}_{d,0:t} = \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{0:t}; \mathbf{u}_{0:t}) \triangleq \alpha_t^i$. Since the estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle$ merely approximates α_t^i, we are not be performing a strict Rao-Blackwellisation; nevertheless, the results will be accurate up to the approximations in the Extended or the Unscented Kalman Filter.

Our algorithm is illustrated in Figure 9. Each particle now holds a sample sequence $\mathbf{x}_{d,0:t}^{(i)}$ and the corresponding continuous estimate $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)} \rangle$. 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 $\mathbf{x}_{d,0}^{(i)}$, the corresponding continuous distribution $p(\mathbf{x}_{c,0} | \mathbf{x}_{d,0}^{(i)})$ is specified by the PHA model.

The algorithm then proceeds to expand the mode sequences and updates the corresponding continuous estimates (see Figure 9, Step 2). In each time step, we first evolve each particle by taking one random sample $\mathbf{x}_{d,t}^{(i)}$, for each particle, from the proposal distribution $q(\mathbf{x}_{d,t}; \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. This distribution takes into account the transition model for the PHA from mode $\mathbf{x}_{d,t-1}^{(i)}$ to $\mathbf{x}_{d,t}$ and can be efficiently computed from the transition model and the continuous estimates, as described in the next section. For each new mode sequence $\mathbf{x}_{d,0:t}^{(i)}$, we compute the importance weight $w_t^{(i)}$. These importance weights take into account the latest observation $\mathbf{y}_t$ and are akin to the observation function $p(\mathbf{y}_t | \mathbf{x}_t)$ in a hidden Markov model. After we compute the importance weights, we resample the trajectories according to their importance weights, using one of the selection schemes, such as residual resampling (Higuchi, 1996; Liu & Chen, 1998) or stratified/systematic sampling (Kitagawa, 1996). This step will, in effect, direct the future expansion of the mode sequences into relevant regions of the state space.

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

$$\mathbf{x}_{c,t} = \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}; \mathbf{d}) + \mathbf{v}_x(\mathbf{d}) \tag{10}$$

$$\mathbf{y}_{c,t} = \mathbf{g}(\mathbf{x}_{c,t}, \mathbf{u}_t; \mathbf{d}) + \mathbf{v}_y(\mathbf{d}), \tag{11}$$

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 $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, P_t^{(i)} \rangle$.

3.3 Proposal distribution

In order to complete the algorithm outlined above, we need to specify the proposal distribution $q(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$, which determines how the discrete mode sequences are evolved. For simplicity we choose the distribution $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$. This distribution expresses the probability of the transition from the mode $\mathbf{x}_{d,0:t-1}^{(i)}$ to each mode $\mathbf{d} \in \mathcal{X}_d$ 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 $\mathbf{x}_{d,t-1}$

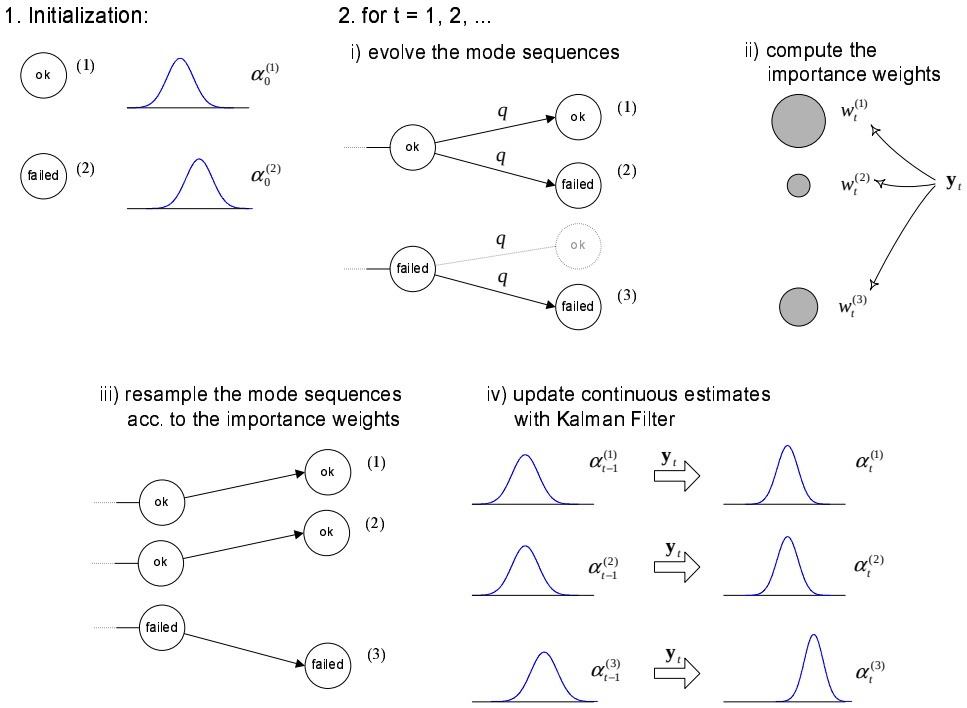

Figure 9: Gaussian particle filter for PHA.

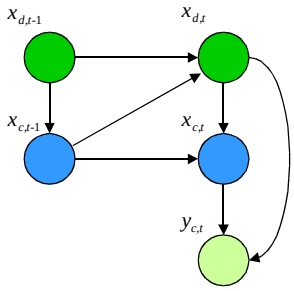

Figure 10: Conditional dependencies in PHA among the state variables $\mathbf{x}_c, \mathbf{x}_d$ and the output $\mathbf{y}$, expressed as a dynamic Bayesian network (Dean & Kanazawa, 1989). The edge from $\mathbf{x}_{c,t-1}$ to $\mathbf{x}_{d,t}$ represents the dependence of $\mathbf{x}_{d,t}$ on $\mathbf{x}_{c,t-1}$, that is, autonomous transitions.

alone 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 10).

The key idea is to compute the proposal distribution for each tracked mode sequence $\mathbf{x}^{(i)}_{d,0:t-1}$ using the corresponding continuous estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t}, \mathbf{P}^{(i)}_t \rangle$. Since the estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t}, \mathbf{P}^{(i)}_t \rangle$ captures the posterior distribution of the continuous state conditioned on the i-th sequence, $p(\mathbf{x}_{c,t-1} | \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, we can integrate it out to obtain a transition distribution con-

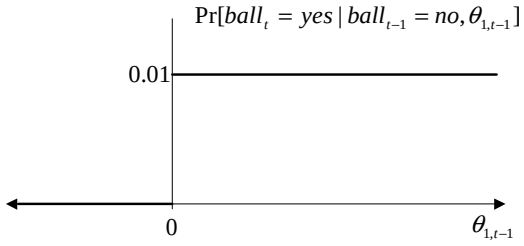

Figure 11: Probability of a mode transition `ball=no` to `ball=yes` as a function of $\theta_{1,t-1}$.

ditioned on $\mathbf{x}^{(i)}_{d,0:t-1}$, $\mathbf{y}_{0:t-1}$ and $\mathbf{u}_{0:t}$ alone:

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

The first equality follows from the total probability theorem. The second 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 $t-1$.

Typically, when performing Rao-Blackwellisation, the integral in Equation 12 is difficult to evaluate efficiently, as noted in (Murphy & Russell, 2001), since the integral in Equation 12 often does not have a closed form. For PHA, however, efficient evaluation of this integral is possible. Recall that the distribution of the discrete evolution of a PHA for one step is specified as a finite set of guards c and their associated transition probabilities p_τ. Each guard specifies a region over the continuous state and automaton's input/output variables, for which the transition distribution p_τ holds. For example, the acrobot model in Figure 2 has two guard conditions for the mode `has-ball=no`: $\theta_1 < 0$ and $\theta_1 > 0$, with associated transition probabilities to mode `yes` 0 and 0.01, respectively. Since the transition probability is constant in intervals $(-\infty; 0)$ and $(0; +\infty)$, the transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ takes on a finite number of values as a function of $\mathbf{x}_{c,t-1}$ (see Figure 11).

To see how this observation aids in the evaluation of the proposal distribution, consider the left term in the integral in Equation 12, $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$. Since this term takes on only a finite number of values $p_{\tau j}(\mathbf{x}_{d,t})$, we can split the integral domain into the sets

X_j that satisfy the constraints c_j and factor out the transition probability $p_{\tau j}$:

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

The second equality holds because, for the region X_j, the conditional distribution $p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$ 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 c_j in the distribution $\alpha^{(i)}_{t-1}$. Hence, the key contribution for PHA is that the discrete transition probability can be computed directly, by integrating the p.d.f distribution function $\alpha^{(i)}_{t-1}(\mathbf{x}_{c,t-1})$ for each sample i.

3.4 Evaluating the probability of a transition guard

Given the derivation in the previous section, the remaining challenge in computing the proposal distribution is to evaluate the probability of satisfying the guard c_j, given the distribution $\alpha^{(i)}_{t-1} = p(\mathbf{x}_{c,t-1}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$. Without loss of generality, assume that the guards are only over the continuous state. Guards of the form $c_c(\mathbf{x}_c) \wedge c_d(\mathbf{u}_d)$ can be handled by setting $\Pr_{\alpha^{(i)}_{t-1}}[X_j] \equiv 0$ whenever $c_d(\mathbf{u}_d)$ is not satisfied. More complex guards can be transformed to a number of guards using elementary rules of logic.

Since computing the an integral over the posterior distribution $\alpha^{(i)}_{t-1}$ exactly is often intractable, we approximate $\alpha^{(i)}_t$ with a Gaussian distributions with the estimated mean $\hat{\mathbf{x}}^{(i)}_{c,t-1}$ and covariance $\mathbf{P}^{(i)}_{t-1}$. While this approximation will introduce estimation error in the proposal, it allows us to compute the proposal distribution efficiently, since the problem simplifies to computing an integral over a Gaussian distribution with mean $\hat{\mathbf{x}}^{(i)}_{c,t-1}$ and covariance matrix $\mathbf{P}^{(i)}_{t-1}$:

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

This approach was suggested in (Hofbaur & Williams, 2002a) for single-variate guards of the form $x < c$ and $x > c$, where $x \in \mathbf{x}$ is a continuous state variable and c is a real constant. In this section, we first summarize their procedure and then show how it generalizes to multi-variate linear conditions.

Interval single-variate guards When the guards are of the form $x < c$ or $x \leq c$, for some constant c, such as $\theta_1 < 0.7$, the integral in Equation 14 simplifies to evaluating the

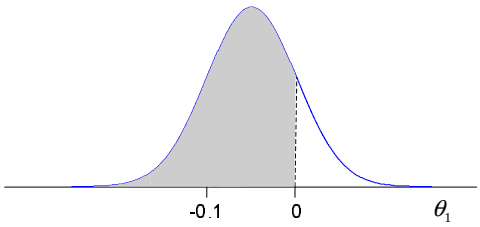

Figure 12: Evaluating single-variate guard conditions.

cumulative density function of the normal variable $\mathcal{N}(\mu, \sigma^2)$, where $\mu = (\hat{\mathbf{x}}^{(i)}_{c,t-1})_x$ is the mean of variable x in $\hat{\mathbf{x}}^{(i)}_{c,t-1}$ and $\sigma^2 = (\mathbf{P}^{(i)}_t)_x$ is its variance (Figure 12):

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

The cumulative density function $D(c)$ can be 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 $[l; u]$.[3], 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, $D(u) - D(l)$. Such guards are thus slightly more expressive, while maintaining the same computational complexity.

Rectangular multi-variate guards Multivariate guards are often needed to represent more complex constraints on transitions. For example, in 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 heights h_1 and h_2 in the two tanks compare to h. In this system, the transition into the `no-flow` mode with would be conditioned on the guard $(h_1 < h) \wedge (h_2 < h)$.

In general, the rectangular multi-variate guards 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. Evaluating the probability of such a multi-variate guard amounts to evaluating the multi-dimensional (hyper)rectangular integral over a Gaussian distribution (see Figure 13):

$$\Pr_{\alpha^{(i)}_t}[X_j] \approx \frac{1}{(2\pi)^{n_c/2}|\mathbf{P}_I|^{1/2}} \int_{l_{i_1}}^{u_{i_1}} \int_{l_{i_2}}^{u_{i_2}} \cdots \int_{l_{i_n}}^{u_{i_n}} e^{-\frac{1}{2}(\mathbf{x}_c - \hat{\mathbf{x}}_{c,I})^T {\mathbf{P}_I}^{-1} (\mathbf{x}_c - \hat{\mathbf{x}}_{c,I})} \, d\mathbf{x}_c, \tag{16}$$

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 can be evaluated efficiently using

3. 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.

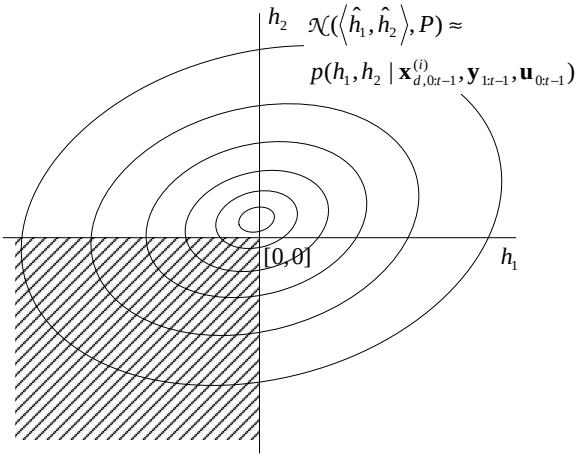

Figure 13: Rectangular integral over a Gaussian approximation of the posterior density of h_1 and h_2, $p(h_1, h_2|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$.

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 16; however, numerical methods tend to perform better.

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, mode variable for the flow would be guarded by the linear guards $h_1 - h_2 > 0$ and $h_1 - h_2 < 0$ (see Figure 14). 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 and would make the covariance matrix singular. Instead, the key idea is to apply a linear transform to the Gaussian distribution and reduce the computation to one of the previous two cases.

Suppose that the guard condition c is expressed as a conjunction of clauses $\bigwedge_{i=1}^{n} l_i < \mathbf{a}_i\mathbf{x}_c < u_i$. 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 $\mathbf{a}_i\mathbf{x}_c < u_i$ and can be viewed as a limiting case of SoftMax conditional distribution (Koller et al., 1999). Let $\mathbf{A} \triangleq \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n\end{bmatrix}^T$ be the matrix of the guard coefficients and $\mathbf{z} \triangleq \mathbf{A}\mathbf{x}_{c,t-1}$ be the derived vector with n elements. Then the guard $c \triangleq \bigwedge_{i=1}^{n} l_i < \mathbf{a}_i\mathbf{x}_c < u_i$ is equivalent to the guard $\bigwedge_{i=1}^{n} l_i < \mathbf{z}_i < u_i$. The probability of the guard c can thus be evaluated as an integral

$$\int_{l_{i_1}}^{u_{i_1}} \int_{l_{i_2}}^{u_{i_2}} \cdots \int_{l_{i_n}}^{u_{i_n}} p(\mathbf{z}_t|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})\ d\mathbf{x}_c, \tag{17}$$

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

In general, the posterior distribution of $\mathbf{z}$ is as intractable as the posterior distribution $\alpha^{(i)}_{t-1}$ of $\mathbf{x}_{c,t-1}$. Nevertheless, if we approximate $\alpha^{(i)}_{t-1}$ with a Gaussian $\mathcal{N}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1})$, the

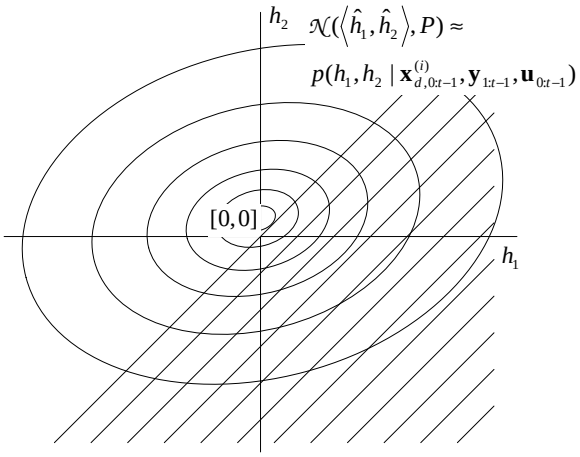

Figure 14: Linear guard $h_2 < h_1$ over the Gaussian approximation of the posterior density of h_1 and h_2.

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, linear guards can once again be evaluated as a rectangular integral over a Gaussian distribution.

3.5 Importance weights

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

$$w^{(i)}_t \triangleq \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}^{(i)}_{d,0:t}, \mathbf{y}_{0:t-1})p(\tilde{\mathbf{x}}^{(i)}_{d,t}|\tilde{\mathbf{x}}^{(i)}_{d,0:t-1}, \mathbf{y}_{0:t-1})}{q(\tilde{\mathbf{x}}^{(i)}_{d,t}; \tilde{\mathbf{x}}^{(i)}_{d,0:t-1}, \mathbf{y}_{0:t})} = p(\mathbf{y}_t|\tilde{\mathbf{x}}^{(i)}_{d,0:t}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t}) \tag{18}$$

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 (see Figure 10). Nevertheless, it is possible to compute the likelihood from the continuous estimate $\langle \hat{\mathbf{x}}^{(i)}_{c,t-1}, P^{(i)}_{t-1} \rangle$. In SLDS models, one may propagate the Gaussian distribution $\alpha^{(i)}_{t-1} = \mathcal{N}(\hat{\mathbf{x}}^{(i)}_{c,t-1}, \mathbf{P}^{(i)}_{t-1})$, represented by this estimate, through the continuous transition and observation functions for mode $\mathbf{x}^{(i)}_{d,t}$ (Equation 11). The resulting distribution corresponds to the measurement innovation $\langle \mathbf{r}, \mathbf{S} \rangle$, computed in the Kalman Filter, and can be used to compute the observation likelihood in Equation 18 *exactly* as

$$w^{(i)}_t = \frac{1}{(2\pi)^{N/2}|\mathbf{S}^{(i)}_t|^{1/2}} e^{-0.5\mathbf{r}^T(\mathbf{S}^{(i)}_t)^{-1}\mathbf{r}} \tag{19}$$

(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, it is still possible to compute the weight in Equation 18 from the Extended or the Unscented 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 $\mathbf{x}_{d,t}^{(i)}$:

$$\hat{\mathbf{x}}_{c,t}^{(i-)} = \mathbf{f}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{u}_{t-1}) \tag{20}$$

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_c}|\hat{\mathbf{x}}_{c,t}^{(i)} \tag{21}$$

$$\mathbf{P}_t^{(i-)} = \mathbf{A}\mathbf{P}_t^{(i-)}\mathbf{A}^T + \mathbf{Q}, \tag{22}$$

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 $\mathbf{S}_t^{(i)}$:

$$\mathbf{y}_p = \mathbf{g}(\hat{\mathbf{x}}_{c,t}^{(i-)}, \mathbf{u}_t) \tag{23}$$

$$\mathbf{C} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}_c}|\hat{\mathbf{x}}_{c,t}^{(i-)} \tag{24}$$

$$\mathbf{S}_t^{(i)} = \mathbf{C}\mathbf{P}_t^{(i-)}\mathbf{C}^T + \mathbf{R}, \tag{25}$$

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 18 can then be approximated with normal p.d.f.

$$w_t^{(i)} = \frac{1}{(2\pi)^{N/2}|\mathbf{S}_t^{(i)}|^{1/2}} e^{-0.5\mathbf{r}^T(\mathbf{S}_t^{(i)})^{-1}\mathbf{r}}. \tag{26}$$

3.6 Putting it all together

The final algorithm is shown in Figure 15. Note that the order of the Exact and Selection step of the generic RBPF algorithm in Figure 5 has been switched, because the innovation mean and covariance, computed in the Kalman Filter update step, are used to compute the importance weight.

Several straightforward optimizations can be employed to further improve the performance of the algorithm. First, since the algorithm is recursive and only depends on the latest state estimate and the latest mode assignment in a mode sequence, it is sufficient to maintain only the latest mode assignment $\mathbf{x}_{d,t}^{(i)}$, rather than the complete mode sequences $\mathbf{x}_{d,0:t}^{(i)}$. Furthermore, the algorithm can compute the transition probability $P_{\mathcal{T}}$ in the Importance Sampling step only once for each unique sample. This can be accomplished by maintaining the number of offspring N_i generated in the Selection step, and taking N_i random samples from the computed transition probability $P_{\mathcal{T}}$.

4. Gaussian 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

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 $\hat{\mathbf{x}}_{c,0}^{(i)} \leftarrow \mathbb{E}[\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(i)}]$ and 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 $i = 1, \ldots, N$
     - compute the transition distribution $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
     - sample $\tilde{\mathbf{x}}_{d,t}^{(i)} \sim p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
     - let $\tilde{\mathbf{x}}_{d,0:t}^{(i)} \leftarrow (\mathbf{x}_{d,0:t-1}^{(i)}, \tilde{\mathbf{x}}_{d,t}^{(i)})$

   (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: Gaussian particle filter for PHA.

the models can be split on a component-by-component basis, thus enhancing the reusabiliy of the models and reducing modeling costs. In this chapter we extend our Gaussian particle filter, developed in the previous section, to handle composed CPHA models, see Section 2.2 for an overview of CPHA. In CPHA, components transition independently, conditioned on 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 (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. We then show how sampling from the posterior (Akashi & Kumamoto, 1977; Doucet et al., 2000) can be adapted to CPHA. The key idea is to evaluate the observation function for mode transitions independently. This results in an improved proposal q that incorporates some information in the latest observations, without performing a Kalman Filter update for each successor mode.

4.1 Sampling from the prior

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 probability of being in the mode $\mathbf{x}_{d,t}$, 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

$$w_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \triangleq P_{\mathcal{O},t}^{(i)}. \tag{27}$$

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 $p(\mathbf{x}_{d\,k,t}|\mathbf{x}_{d\,k,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
       - sample $\tilde{\mathbf{x}}_{d,t\ k}^{(i)} \sim p(\mathbf{x}_{d\,k,t}|\mathbf{x}_{d\,k,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
     - 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 16: Gaussian particle filter for CPHA.

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 $\mathcal{X}_{d\,k}$, 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 $\prod_k p(\mathbf{x}_{d\,k,t}|\mathbf{x}_{d\,k,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$, for all components k in the model (see Section 3.3).

Figure 16 shows the pseudocode for the resulting algorithm. The algorithm is based on the algorithm presented in Section 3, except that in the importance sampling step, we compute the transition distribution and evolve the sampled mode sequences on a component-by-component basis.

4.2 Sampling with improved proposal

One possible drawback with using the transition distribution as a proposal in the fault diagnosis domain is that fault transitions typically have a low prior probability, and many particles may be needed to sample and detect the fault. If there is a significant amount of information contained in the observations, it may be useful to incorporate this information into the proposal, so that modes with high probability in the posterior distribution are also likely in the proposal. It may be possible to use domain-specific heuristics to guide the sampling process (Dearden & Clancy, 2002); however, such heuristics are difficult to construct and can be very fragile. One systematic solution is to use the optimal proposal

distribution[4], $q = p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{0:t-1}, \mathbf{y}_{0:t}, \mathbf{u}_{0:t})$ (Akashi & Kumamoto, 1977; Doucet et al., 2000). Let us expand it in terms of the transition distribution and the observation likelihood:

$$p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = \tag{28}$$

$$= \frac{p(\mathbf{x}_{d,t}, \mathbf{y}_t|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})}{p(\mathbf{y}_t|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})} \tag{29}$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_{d,t}, \mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) p(\mathbf{x}_{d,t}|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})}{\sum_{d \in \mathcal{X}_d} p(\mathbf{y}_t, \mathbf{x}_{d,t} = d|\mathbf{x}^{(i)}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})} \tag{30}$$

Hence, the distribution represents the "increment" in the posterior distribution of the mode sequence from time step $t-1$ to time step t (the numerator of Equation 30), among all the sequences extending $\mathbf{x}^{(i)}_{d,0:t-1}$ (the denominator of Equation 30).

In hybrid models, the optimal proposal distribution is a double-edged sword: Although the performance of the filter may improve on a per-sample basis, significantly more computation needs to be performed, in order to compute the proposal distribution. In practice, the trade-off will depend on the amount of information in the observations and on the prior probabilities of the faults. The reason is that, in order to compute the optimal proposal distribution, the algorithm would need to evaluate the observation likelihood at each reachable successor mode. While this approach may work in a single-component system that has only a few modes, its performance, as a function of the execution time, will degrade as the number of mode assignment increases. With a total of 450,000 modes in the BIO-Plex model (Hofbaur & Williams, 2004), the algorithm would have to perform 450,000 Kalman Filter updates *for each sample*. With one Kalman Filter update taking as much as 1ms for a four-variable continuous model, this approach would take at least ten hours to perform one iteration, when taking 70 samples for the model described in (Hofbaur & Williams, 2004).

Instead, our second algorithm incorporates the latest observations into the sampling process, but does not enumerate all the successor modes. The key idea is to compute the observation likelihood, $P_{\mathcal{O}}$, for each individual component transition, and combine these with the transition distribution, $P_{\mathcal{T}}$. In the importance sampling step, the algorithm now computes the observation likelihood for each component k and each mode m in that component. The transitions are treated independently: While we transition the component k into mode m and compute the observation likelihood for the newly evolved sequence, all of the other components remain in the same mode. Together with the transition distribution $P^{(i)}_{\mathcal{T},t\ k}$, this observation likelihood determines the probability of transitioning to mode m of the proposal distribution $q^{(i)}_{t\ k}$, for the component k. This proposal distribution is used to evolve the mode variables in the k-th component, in order to obtain the new sample $\tilde{\mathbf{x}}^{(i)}_{d,t\ k}$.

In the Exact step, the algorithm updates the continuous state, using the newly sampled mode, and computes the importance weights for the newly obtained samples. Recall from

4. This distribution is optimal in the sense that it minimizes the variance of importance weights (Doucet et al., 2000).

Section 2.4 that importance weights need to satisfy the following equality:

$$w_t^{(i)} = \frac{p(\mathbf{y}_t|\mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) p(\mathbf{x}_{d,t}^{(i)}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})}{q(\mathbf{x}_{d,t}^{(i)}; \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})}. \tag{31}$$

In this case, since the samples $\mathbf{x}_{d,t\ k}^{(i)}$ are evolved independently according to the proposal distribution $q_{t\ k}^{(i)}$ for each component k, the proposal distribution for the *vector* $\tilde{\mathbf{x}}_{d,t}^{(i)}$ becomes

$$q(\mathbf{x}_{d,t}^{(i)}; \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = \prod_k q_{t\ k}^{(i)}. \tag{32}$$

Similarly, the prior transition distribution $P_t^{(i)}$ for the sample $\tilde{\mathbf{x}}_{d,t}^{(i)}$ becomes

$$P_t^{(i)} = \prod_k P_{\mathcal{T},t\ k}^{(i)}. \tag{33}$$

Therefore, the weights $w_t^{(i)}$ in Equation 31 reduce to

$$w_t^{(i)} = \frac{\mathcal{N}(\mathbf{r}, \mathbf{S}_t^{(i)}) \prod_k P_{\mathcal{T},t\ k}^{(i)}}{\prod_k q_{t\ k}^{(i)}} \tag{34}$$

.

The algorithm works best when the effects of mode transitions are observed independently, or nearly independently, among the component mode variables. This occurs, for example, when the transitions occur in independent or weakly dependent components. In these cases, the observation likelihood in one component is independent of, or weakly-dependent on, the observation likelihoods in the other components. The proposal distribution q is then near the optimal proposal $q(\mathbf{x}_{d,t}^{(i)}; \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t}) = p(\mathbf{x}_{d,t}^{(i)}; \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$.

5. Experimental Results

Having developed our Gaussian particle filter algorithm for CPHA in the previous two sections, we now turn to evaluating its performance and comparing it to the k-best filter (Hofbaur & Williams, 2002a). While significant attention has been given to comparing the Gaussian and non-Gaussian particle filters (Doucet, Gordon, & Kroshnamurthy, 2001b; Morales-Menéndez et al., 2002; Hutter & Dearden, 2003), little attention has been given to comparing the relative performance of k-best and Gaussian particle filters. An initial step was taken by Lerner (2002). In order to bridge this gap, we evaluate the performance of the two algorithms on two principal scenarios that exhibit each method's strengths and weaknesses. We also evaluate the performance of sampling from the posterior, with the algorithm described in Section 4.2.

We consider the acrobatic robot example, introduced in Section 2.2. 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

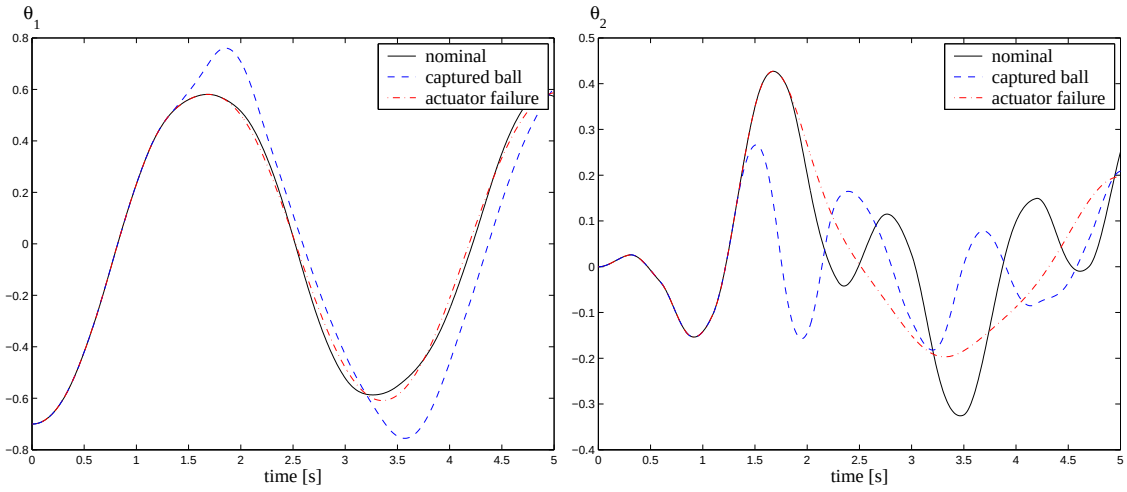

Figure 17: The evolution of θ_1 (left) and θ_2 (right) for the nominal acrobot model scenarios.

this example consists of two variables, representing whether or not the robot holds a ball on its lower link (variable `has-ball`) and whether or not its actuator has broken (variable `actuator`). The complete discrete transition model is shown in Figures 2 and 3. The goal of hybrid estimation in this example 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. The dynamic model for two-link systems like the acrobatic robot is highly nonlinear. Furthermore, with four continuous state variables, the hybrid model is already too large to be handled by non-Gaussian particle filters in real time. Finally, the symptoms exhibited by mode changes are very subtle. Over ten time steps, the difference in the continuous trajectory between the `actuator=ok` and the `actuator=failed` modes is less than 0.04rad, which is the standard deviation of our chosen observation noise.

5.1 Acrobot Model with Concentrated Posterior

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 ⟨`has-ball=no`, `actuator=ok`⟩ 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 17.

3. In the *failure* scenario, the robot's actuator breaks at $t = 1.8$. This event causes the actuator to stop exerting any torque, and alters the robot's trajectory, as shown in Figure 17.

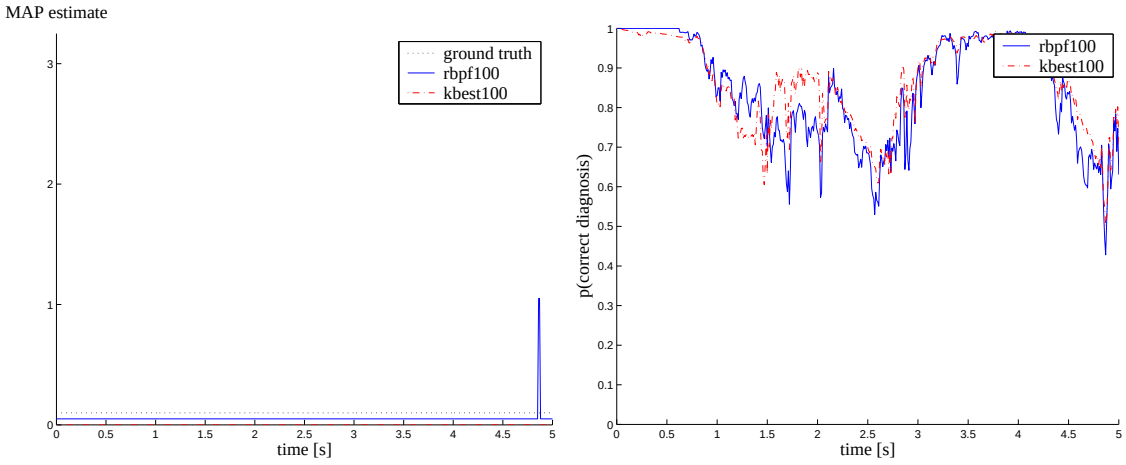

Figure 18: A single run for the nominal scenario using both the Gaussian particle filter (rbpf) and the Gaussian k-best filter (kbest). Left: MAP estimate computed by the Gaussian particle filter with 100 samples. Right: probability of the correct (nominal) diagnosis.

We first present results for hybrid estimation with these scenarios for single executions of the k-best algorithm and the Gaussian particle filter algorithm, and then compare the overall performance of the two algorithms.

5.1.1 Single executions

Figure 18 shows the maximum a posteriori (MAP) estimate of the discrete state by the two hybrid estimation filters for the nominal scenario. It can be seen that the Gaussian particle filter algorithm estimates the most likely mode as ⟨`has-ball=no`, `actuator=ok`⟩, which is in fact the ground truth, except for two timesteps between $t = 4.86$ and $t = 4.87$ seconds.

Figure 18 shows the estimated probability of the diagnosis corresponding to the ground truth, relative to the other possible diagnoses at each time step. For example, for $t < 0.5s$, all of the tracked particles correspond to the correct diagnosis, and so the probability of the correct diagnosis is one in this region. Between $t = 1$ and $t = 3$, however, the transition from mode `has-ball=no` to mode `has-ball=yes` is enabled. Since the likelihood of alternative diagnoses is increased in this time period, the algorithm's MAP estimate has a lower confidence, around 0.7.

Figure 20 shows the results for the ball scenario where the robot catches the ball at $t = 1.3s$. It can be seen that the Gaussian particle filter algorithm estimates the MAP diagnosis correctly, except for an uncertain area close to the transition. It can be seen that close to $t = 1.3s$, the probability of the correct diagnosis drops to around 0.5, indicating that at this time, approximately half of the tracked sequences correspond to the correct diagnosis at the fringe.

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 ⟨`has-ball=yes`,

0. `has-ball=no`, `actuator=ok`
1. `has-ball=yes`, `actuator=ok`
2. `has-ball=no`, `actuator=failed`
3. `has-ball=yes`, `actuator=failed`

Figure 19: Numbers corresponding to the different mode assignments

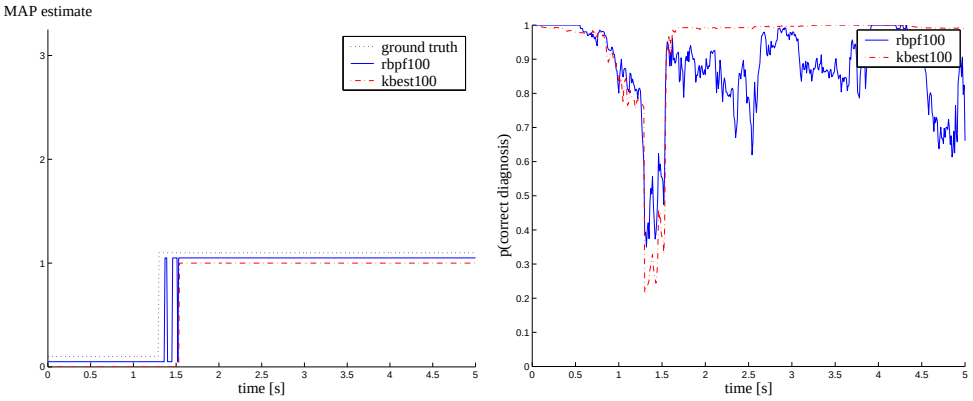

Figure 20: A single run for the ball capture scenario. Left: Maximum a posterior (MAP) estimate computed by the Gaussian particle filter (rbpf) and the k-best filter (kbest). Right: probability of the correct diagnosis `has-ball=yes` for $t \geq 1.3s$.

`actuator=ok`⟩ diagnosis builds up only over time. The low probability of a transition to ⟨`has-ball=yes`, `actuator=ok`⟩ biases the prior towards the ⟨`has-ball=no`, `actuator=ok`⟩ diagnosis. Since the system is noisy, only over time does the observation likelihood due to the correct diagnosis build up to compensate for this bias.

Figure 21 shows the results for the failure scenario, where the actuator loses the ability to exert any torque at $t = 1.8$ seconds. Again, it can be seen that after a delay, the Gaussian particle filter algorithm estimates the MAP diagnosis correctly as (`has-ball=no`, `actuator=failed`). 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.

In all cases, the Gaussian particle filter 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 22 shows the tracking of θ_1 and θ_2 for the nominal scenario. It can be seen that the tracked estimate is very close to the omniscient UKF, where the discrete state of the system is fully known.

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

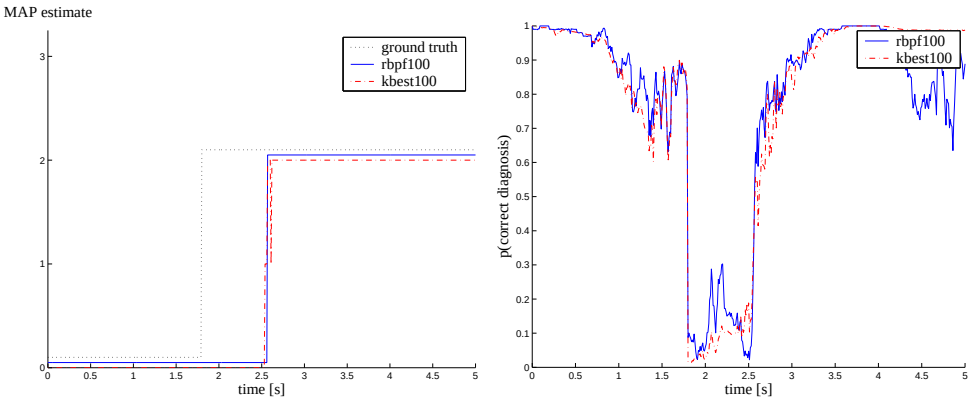

Figure 21: A single run for the actuator failure scenario. Left: Maximum a posterior (MAP) estimate computed by the Gaussian particle filter and the k-best filter. Right: probability of the correct diagnosis $\langle$`has-ball=no`, `actuator=failed`$\rangle$ for $t \geq 1.8s$.

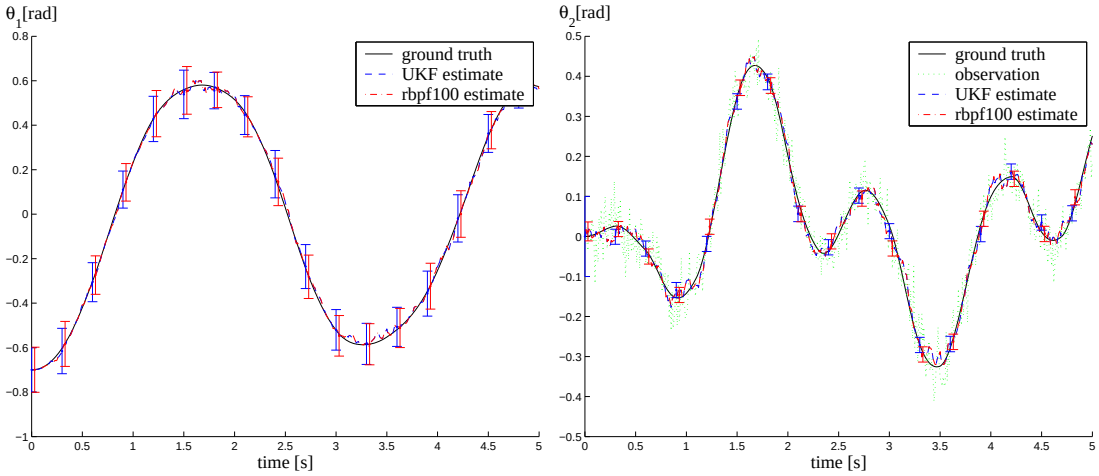

Figure 22: Filtered θ_1 and θ_2 for an execution of the nominal scenario.

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 $\frac{\#\ of\ wrong\ diagnoses}{\#\ time\ steps}$. Here, wrong diagnoses are defined as estimates of the discrete state at the fringe which do not correspond to the same discrete state 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 diagnosis, and $\mathbf{x}_{c,t}$ is the continuous state ground truth. This measure is averaged over all time steps and experiments.

These metrics compare the MAP diagnosis to the ground truth; however it is not necessarily the case that the true MAP hybrid estimate is the same as the ground truth. The scenarios we use were verified to be such that the posterior distribution is concentrated in a small number of hypotheses, and in such a way that the true MAP estimate is in fact the ground truth for the most part (exceptions to this were discussed in section 5.1.1). Hence these statistics are valid for the following experiments, and have also been employed in prior literature (Hutter & Dearden, 2003). An alternative would be to use as a measure the likelihood of the correct diagnosis for discrete estimates and the KL divergence from the omniscient Kalman Filter for the continuous estimates (Lerner, 2002).

5.1.3 Performance Comparison

Figures 23 through 25 show the percentage of diagnostic errors and the mean square tracking error for the three scenarios considered for the acrobot model. For each of the scenarios, the algorithms were run on 20 random observation sequences with fixed mode assignments.

Figure 23 shows that for the nominal scenario, the k-best algorithm makes almost no diagnostic errors. The Gaussian particle filter on the other hand, makes more than 30 per cent diagnostic errors on average for small k; this decays to zero as the number of tracked sequences increases.

The Gaussian particle filter gives a higher number of diagnostic errors because it approximates the true posterior of a given hypothesis, storing some of the information about the distribution in the particle count. Noise in the observations and variations in the random sampling can cause the particles corresponding to the wrong diagnosis to be weighted far greater than those corresponding the correct diagnosis. When this happens, if there are few particles being tracked, all of the particles corresponding to the correct hypothesis can be discarded. This process greatly approximates the true posterior distribution, and since the model has a low transition prior, the resulting diagnostic error persists.

The k-best filter, on the other hand, tracks k *distinct* sequences and calculates the posterior likelihood of these sequences exactly. In this experimental scenario, where there are relatively few mode sequences with non-negligible posterior likelihood, the k-best filter will continually track the sequence corresponding to the ground truth as well as alternative sequences. Hence the k-best filter does not make as many diagnostic errors as the Gaussian particle filter with the nominal scenario.

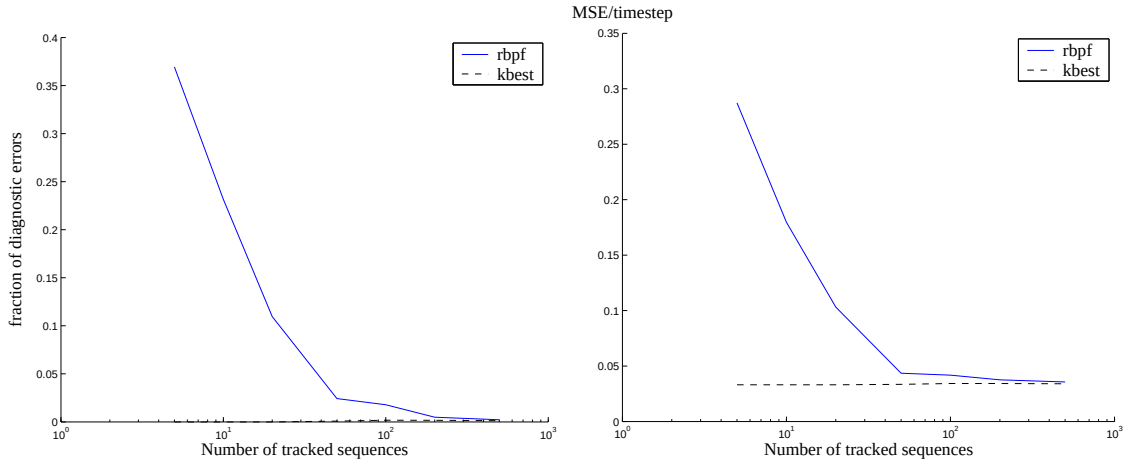

Figure 23: Performance for the nominal scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

Figure 24 shows the average results for the ball scenario. The k-best algorithm again consistently outperforms the Gaussian particle algorithm, although it now makes some diagnostic errors. Again, both algorithms appear to converge to a minimum number of diagnostic errors. Because the true posterior distribution is such that the MAP diagnosis is not the same as the ground truth for a small fraction of the time steps in the experiment, this minimum is non-zero.

Figure 25 shows a somewhat different trend for the failure scenario. For small k, the Gaussian particle filter now makes fewer diagnostic errors on average; for $k > 30$, however, the k-best algorithm outperforms the Gaussian particle filter as was the case with the ball and nominal scenarios. Also, whereas the performance of the Gaussian particle filter improves gradually as k increases, the k-best algorithm shows a large shift in performance between $k = 20$ and $k = 50$.

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.

In these scenarios, therefore, three interesting patterns emerge: Firstly, the k-best algorithm undergoes a phase shift in performance, depending on whether or not k is large enough for hypotheses similar to the ground truth to be tracked. The critical value of k depends on the concentration of the posterior distribution. If the posterior distribution is concentrated in a very small number of distinct sequences, as in the nominal and the ball scenarios, then the k-best method will perform well even for small k. In the failure scenario the posterior distribution was spread among a few more distinct sequences, hence the k-best method gave a large fraction of diagnostic errors for small k. Note that in this case the

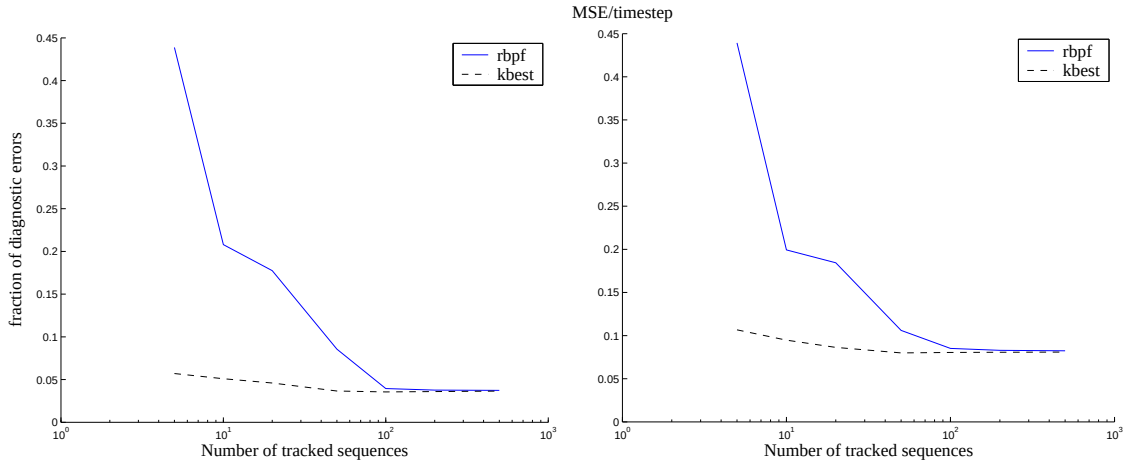

Figure 24: Performance for the ball scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

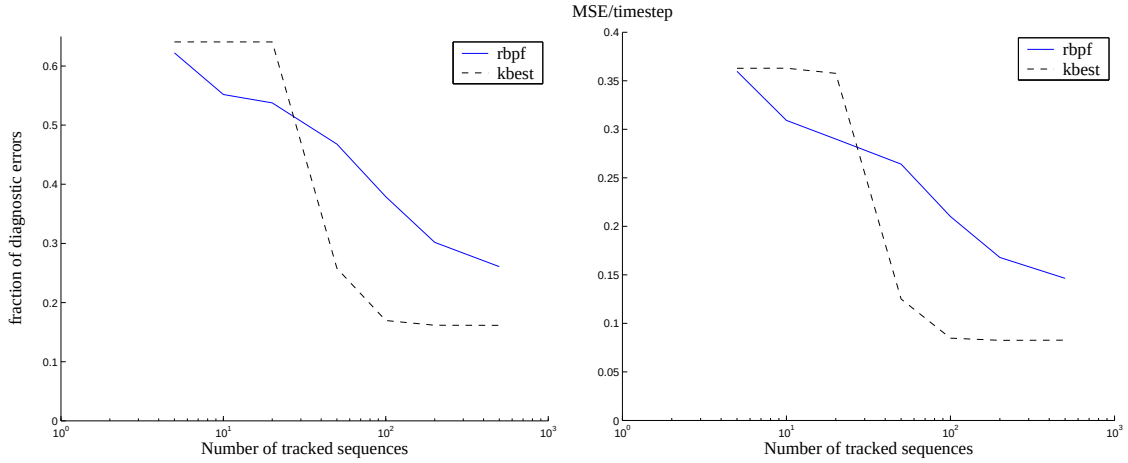

Figure 25: Performance for the actuator failure scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state.

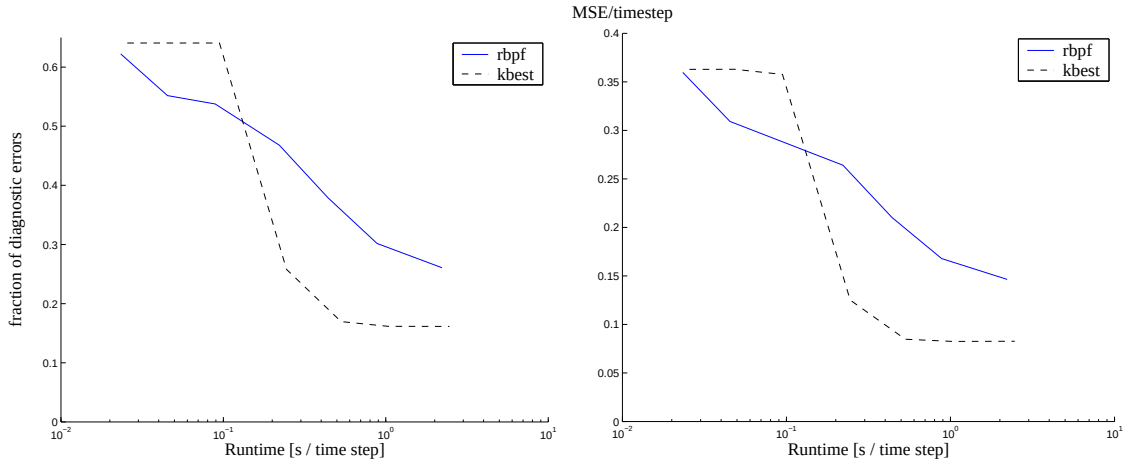

Figure 26: Performance for the failure scenario with respect to run time. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

k-best algorithm still only needed to track 50 sequences for the performance to converge to its optimum.

Secondly, as long as k is greater than the critical value, k-best clearly outperforms Gaussian particle filtering for the scenarios chosen. This is because in these scenarios, the posterior likelihoods of all but a handful of mode sequences are negligible; hence the posterior distribution can be approximated closely by tracking a small set of mode sequences. The k-best method performs well when this is the case because it calculates the exact posterior for a set of k distinct mode sequences. The Gaussian particle filter algorithm, on the other hand, approximates the posterior probability of a given mode sequence, representing some of the information in the number of samples for that mode sequence. It also uses some of its k particles to store duplicate mode sequences, meaning that alternative mode sequences tracked by the k-best enumeration methods are discarded by the particle filter.

Thirdly, the performance of the Gaussian particle filter converges less quickly than the k-best filter to a low fraction of diagnostic errors (See Figure 25). Since the Gaussian particle filter 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.

These results are the same when we compare the performance with respect to the run time of the algorithms. Figure 26 shows this comparison. We found that the run time of the k-best algorithm was no more than 50 per cent greater per time step than the Gaussian particle filter. This has a negligible effect on the log scales against which we compare the performance of the two algorithms.

These insights motivate a modified set of experiments. We would like to investigate situations where the posterior distribution is *not* concentrated in a small number of discrete sequences. In these cases, k-best enumeration techniques will not track a mode sequence which, although having a significant likelihood, is not within the k best mode sequences.

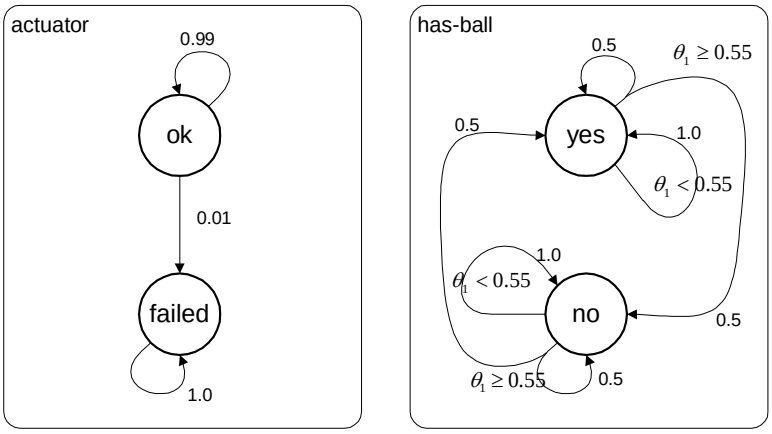

Figure 27: Probabilistic hybrid automata for the body and actuator of the modified acrobot example

Gaussian particle filtering on the other hand, which samples the mode transitions stochastically according their transition priors, may in fact track the correct mode sequence long enough for its observation probability to dominate. These experiments are discussed in the next section.

5.2 Acrobot Model with Dispersed Posterior

The modified acrobot model has the same components as before, however the model parameters are different. Most importantly, the probability of the acrobot catching a ball has changed from 0.05 to 0.5. An example of where this might be necessary is the case where we have no information about the prior likelihood of the acrobot capturing a ball. The increased transition probability means that whenever the transitions between `has-ball=no` and `has-ball=yes` are enabled, the number of mode sequences with a high prior grows exponentially. Secondly, the mass of the ball is decreased and the amplitude of the driving torque is increased. This means that the `has-ball=yes` has less of an impact on the continuous dynamics, while the loss of driving torque due to the `actuator=failed` mode has a more pronounced effect. Finally, the transition guard for the ball capture was changed to $\theta_1 \geq 0.55$. This means that transitions on the `body` component are enabled for fewer timesteps, which makes the results easier to interpret.

The overall effect of these modifications is that the posterior distribution is no longer concentrated in a small number of hypotheses. An exponential number of hypotheses with high priors grows whenever $\theta_1 \geq 0.55$. Since the mass of the ball has been decreased the observation probabilities for each of these sequences are not significantly different in a short time frame. Hence there are a very large number of distinct hypotheses to be tracked, all of which initially have a high posterior probability.

The modified acrobot model is an example where the fair sampling of the Gaussian particle filter can outperform the greedy search in the k-best filter. In a 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 k-best filter tracks the hypotheses in strictly best-first order and in this scenario the best-first enumeration is dominated by the prior probability. Now that there are a very large number of distinct hypotheses with high prior likelihoods, *no* sequences with transitions to the `actuator=failed` mode will be tracked if k is too small. Gaussian particle filtering, on the other hand, does not suffer from this problem since transitions are *sampled* stochastically. This method may 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.

5.2.1 Performance Comparison

The following experiments compare the performance of the k-best algorithm and the Gaussian particle filter when estimating the hybrid state of the modified acrobot model. A scenario was used where the actuator loses the ability to exert any torque at $t = 1.5$ seconds. Figure 28 shows the performance of the k-best and particle filtering algorithms for this scenario.

Figure 28 shows that for the acrobot model with a dispersed posterior distribution, the Gaussian particle filter clearly outperforms the k-best method both in terms of diagnostic errors and mean square estimation error. The performance of the Gaussian particle filter improves as the number of tracked sequences increases, while the the k-best method does not. This is because the exponential explosion of sequences with a high posterior prevents the strict enumeration of the k-best method from considering the initially less likely failure sequence even with large k.

Figure 29 shows the estimated MAP diagnosis and the probability of the ground truth diagnosis for a single experiment with $k = 500$. Even with this large number of tracked sequences, the k-best filter cannot detect the transition from `actuator=ok` to `actuator=failed` at $t = 1.5s$. Due to the exponential number of sequences with either the mode $\langle$`has-ball=no`, `actuator=ok`$\rangle$ or the mode $\langle$`has-ball=yes`, `actuator=ok`$\rangle$ at the fringe, the k-best filter does not track any sequences with the failure transition, and hence makes spurious diagnoses depending on which of these sequences happens to match the observations best. The Gaussian particle filter, by contrast, samples the actuator failure transition fairly. Whenever a particle samples the transition close to where it occurred in reality, the observation likelihood of that particle grows until it dominates the hypothesis space, leading to the correct diagnosis.

Figure 30 shows the continuous tracking of θ_1 and θ_2 for both k-best and Gaussian particle filter algorithms. The Gaussian particle filter 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. Observations of θ_2 are available to the filter, hence the tracking of this variable is close in both cases, however the tracking of θ_1 has large errors in the k-best case. This is because without an accurate estimate of the discrete mode sequence, the continuous dynamics of the system are unknown.

These results have therefore shown that for the modified acrobot model, the Gaussian particle filter algorithm greatly outperforms the k-best method, in sharp contrast to the original acrobot model tested in Section 5.1. By changing the parameters of the model, the

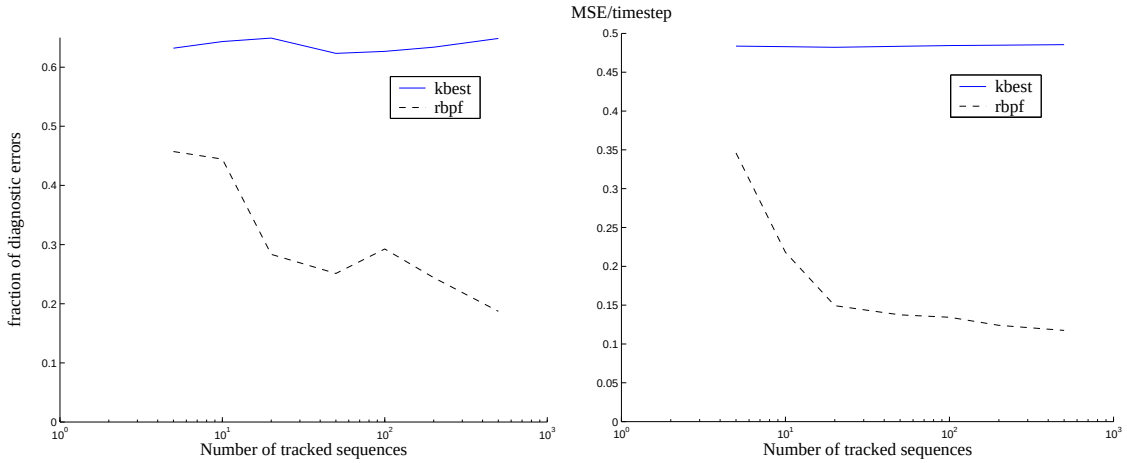

Figure 28: Performance for the failure scenario with the modified acrobot model. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state

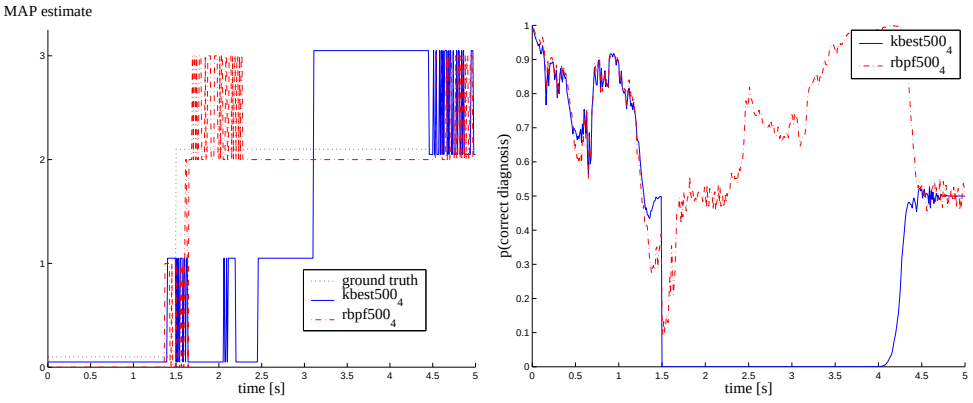

Figure 29: A single run for the failure scenario with the modified acrobot model. Left: Maximum a posterior (MAP) estimate computed by RBPF and k-best filtering algorithm. Right: probability of the correct diagnosis `upper-has-ball=yes` for $t \geq 1.8s$.

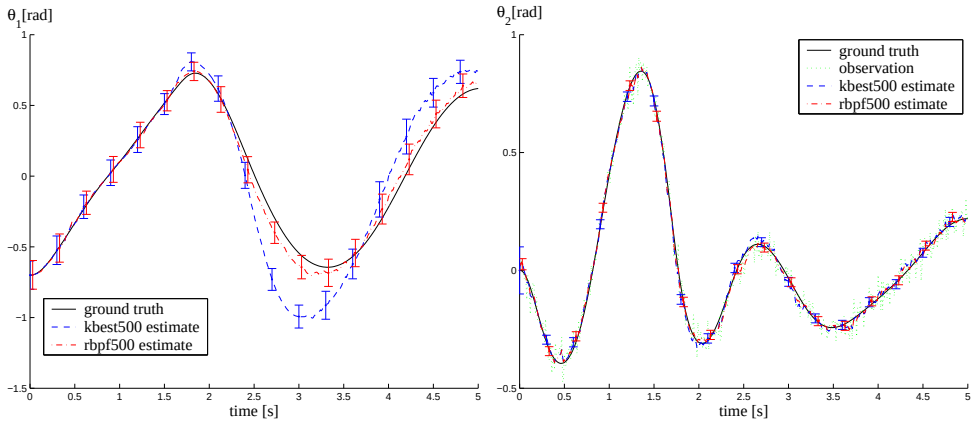

Figure 30: Continuous tracking of the MAP estimate with the modified acrobot model and failure scenario

posterior distribution changed from being concentrated in a small number of hypotheses to being dispersed over a number too great for a greedy enumeration scheme such as k-best to track. The Gaussian particle filter on the other hand was able to track the system with the dispersed posterior.

5.3 Sampling from the Posterior

We carried out experiments to compare the performance of the Gaussian particle filter when sampling from the posterior (Section 4.2) to the performance of the Gaussian particle filter when sampling from the prior (Figure 16). These tests used the original acrobot model, described in Section 2.2, and the failure scenario described in Section 5.1.

Figure 31 shows that for the failure scenario with the acrobot model, sampling from the posterior does not perform any better on average than sampling from the prior. The slight differences in diagnostic error visible in Figure 31 are noise due to the inherent randomness in any given experiment. Figure 31 also shows that sampling from the posterior is significantly more costly in terms of running time per time step, and hence has a worse performance for a given running time than sampling from the prior. Sampling from the posterior takes more time, because, in order to compute the proposal distribution, the filter needs to perform Kalman filter updates several times per particle, as discussed in Section 4.2.

Sampling from the posterior does not improve the diagnostic performance of the algorithm because the effect of the discrete mode on the observed continuous state is delayed. In the continuous acrobot dynamics, the mode of the system affects the accelerations $\ddot{\theta}_1$ and $\ddot{\theta}_2$ through the weight of the lower link (variable `has-ball`) or through the torque exerted by the actuator (variable `actuator`). In the discretized model used by the particle filter, this means that the mode at time t affects the angular velocities ω_1 and ω_2 at time t, which in turn affects the observed state θ_2 at time $t+1$. Hence the effect of the mode on the observed state is not manifested until the next time step later. This means that, in the case of the acrobot model, all successor modes of the sequence $x^{(i)}_{d,0:t-1}$ have the same

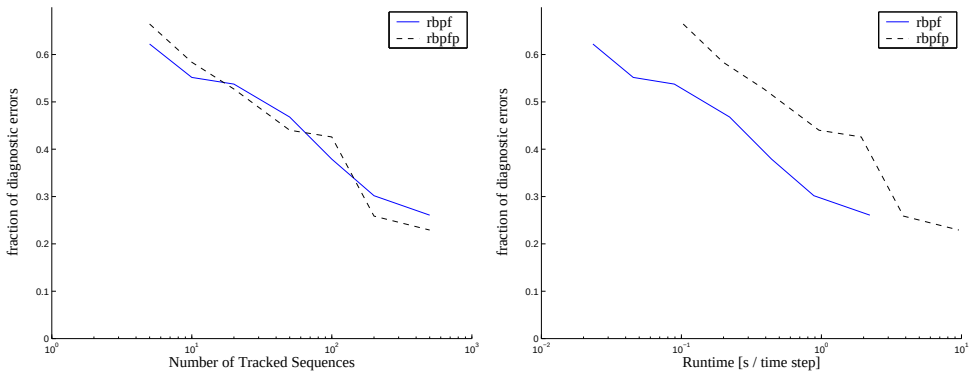

Figure 31: Percentage of diagnostic errors for the failure scenario, as a function of number of tracked mode sequences (left) and running time per time step (right).

observation likelihood, and hence sampling from the posterior by incorporating the most recent observation will not sample mode transitions more fairly than sampling from the prior does.

This results have therefore shown that sampling from the posterior affords us no significant advantage in a system where the effect of the discrete mode on the observed continuous state is delayed in the discretized form. This is the case when the discrete mode affects the derivative of the observed state, and not the observed state itself.

5.4 Discussion

We have carried out an experimental comparison of the k-best and Gaussian particle filter methods with two versions of the acrobot model introduced in Section 2.2. Scenarios were used where the acrobot swings normally, where it catches a ball, and where the actuator fails.

In scenarios where the posterior distribution was concentrated in a small number of distinct hypotheses, the k-best filter outperformed the Gaussian particle filter algorithm as long as k was greater than a critical value, which was between 20 and 50 for the failure scenario. In this case the fraction of diagnostic errors and the mean square error of the continuous estimate were significantly lower for k-best, and in addition the performance of the k-best filter converged much more quickly than for the Gaussian particle filter, exhibiting a marked shift in performance at a critical value of k. The reason for the difference in performance between the k-best method and the Gaussian particle filter is that the Gaussian particle filter approximates the posterior distribution of individual sequences, representing some of the information in the number of duplicated hypotheses, while the k-best algorithm stores the posterior probability of a given mode sequence exactly. When the scenario has a concentrated posterior, the k-best algorithm is able to track the relevant mode sequences exactly, and hence performs better than the Gaussian particle filter.

When the posterior distribution is not concentrated in a small number of distinct hypotheses, the Gaussian particle filter performs much better than the k-best algorithm. In

these cases, the k-best algorithm, by enumerating the different hypotheses in a strictly best-first order, fails to track any hypotheses corresponding to a transition with a low prior, such as a failure transition. The Gaussian particle filter samples such transitions according to their prior probability and hence is able to detect them much more effectively.

These insights motivate a new algorithm which combines the exact, greedy search of the k-best algorithm with the stochastic sampling of the Gaussian particle filter. This is analogous to the trade-off between exploitation and exploration present in many Artificial Intelligence techniques. This is a topic of ongoing research.

We have also presented an experimental comparison of the Gaussian particle filter algorithm using sampling from prior, and using sampling from the posterior. The results showed that sampling from the posterior is more costly in terms of computation time, and affords no performance benefit for systems where the effect of the discrete mode on the observed state is delayed by more than one time step.

6. Related work

Several algorithms have addressed the problem of the exponential growth of the 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, 2002), to the setting of hybrid dynamic Bayesian networks with SoftMax transitions, using numerical integration techniques instead of the Kalman Filter. Similarly to ours, their algorithm provides an any-time solution to the hybrid state estimation problem.

Hofbaur and Williams (2002a) introduced autonomous transitions to the models in the context of Concurrent Probabilistic Hybrid Automata. They introduced an any-time k-best filtering algorithm for concurrent systems, based on the k-best filtering algorithm for concurrent probabilistic constraint automata (Williams & Nayak, 1996). Their algorithm extracts a leading set of sequences in the order of their priors using a combination of branching and A* algorithm that exploits preferential independence and guarantees to find the next set of k leading sequences at each time step.

In the particle filtering community, several papers (Avitzour, 1995; Kitagawa, 1996) have proposed to use 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 $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$. Doucet (1997, 1998) 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, he proved sev-

eral properties on 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.

Similar to, but independent from our work, Hutter and Dearden (2003) combined the look-ahead Rao-Blackwellised particle filter with an Unscented Kalman Filter, in order to improve the accuracy of the continuous estimates. In our work (Funiak & Williams, 2003), we have introduced autonomous transitions and drew parallels to prior approaches in hybrid model-based reasoning.

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.

7. Conclusion

In this paper, we investigated the problem of estimating the state of system represented with probabilistic hybrid models. Our main accomplishment is an efficient Gaussian particle filtering algorithm, developed in Sections 3 and 4, that handles 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 reuse the continuous estimates associated with the tracked mode sequences.

In Section 3, we presented significant contributions related to discrete state transitions that depend on the continuous state (autonomous mode transitions). We extended the class of models, for which transition probabilities can be computed efficiently and explored the approximations that occur in the posterior of the continuous space when autonomous transitions are present. Due to the similarities in the theoretical development of both Gaussian particle filtering and k-best filtering, our results translate directly to prior k-best filtering algorithms that allow sharp transition guards in the models (Hofbaur & Williams, 2002a, 2002b).

Our contributions are, however, not merely theoretical. In Section 5, we demonstrated our algorithm on a simulated highly nonlinear system, and empirically compared its performance with k-best filter (Hofbaur & Williams, 2002a; Lerner et al., 2000). When the posterior is concentrated in a few nominal or single-fault sequences, a k-best filter is a clear winner. However, when the correct diagnosis is repeatedly left out from the leading set of mode sequences due to rapid branching and disparsity of the posterior, a Gaussian particle

filter is more successful. This development suggests that it may be possible to unify the two approaches in a stochastic search algorithm that shares the strengths of both methods.

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.

Cowell, R., Dawid, A., Lauritzen, S., & Spiegelhalter, D. (1999). *Probabilistic Networks and Expert Systems*. Springer, New York.

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. (1997). *Monte Carlo Methods for Bayesian Estimation of Hybrid Markov Models. Application to Radiation Signals*. Ph.D. thesis, Universit Paris-Sud, Orsay. (in French with Chapters 4 and 5 in English).

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., & Kroshnamurthy, 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.

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).

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.

Higuchi, T. (1996). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, *59*(1), 1–23.

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.

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*.

Kim, C.-J. (1994). Dynamic linear models with Markov switching. *Journal of Econometrics*, *60*, 1–22.

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

Koller, D., Lerner, U., & Angelov, D. (1999). A general algorithm for approximate inference and its application to hybrid bayes nets. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 324–333.

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

Lauritzen, S. L. (1992). Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, *87*(420), 1098–1108.

Lerner, U. (2002). *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.

Liu, J. S., & Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, *93*, 1032–1044.

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.

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

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*.

Russell, S. J., & Norvig, P. (Eds.). (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Shumway, R. H., & Stoffer, D. S. (1991). Dynamic linear models with switching. *Journal of the American Statistical Association*, *86*, 763–769.

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

Wan, E. A., & van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*.

Williams, B. C., Ingham, M., Chung, S., & Elliott, P. (2003). Model-based programming of intelligent embedded systems and robotic explorers. *invited paper in Proceedings of the IEEE, Special Issue on Modeling and Design of Embedded Software*, *9*(1), 212–237.

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