

# A combined stochastic and greedy hybrid estimation capability for concurrent hybrid models with autonomous mode transitions

Lars Blackmore<sup>a,\*</sup>, Stanislav Funiak<sup>b</sup>, Brian C. Williams<sup>a</sup>

<sup>a</sup>Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA 02139, United States

<sup>b</sup>Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, United States

Received 2 August 2006; received in revised form 27 June 2007; accepted 17 July 2007

Available online 26 July 2007

## Abstract

Probabilistic hybrid discrete/continuous models, such as Concurrent Probabilistic Hybrid Automata (CPHA) are convenient tools for modeling complex robotic systems. In this paper, we present a novel method for estimating the hybrid state of CPHA that achieves robustness by balancing greedy and stochastic search. To accomplish this, we (1) develop an efficient stochastic sampling approach for CPHA based on Rao–Blackwellised Particle Filtering, (2) perform an empirical comparison of the greedy and stochastic approaches to hybrid estimation and (3) propose a strategy for mixing stochastic and greedy search.

The resulting method handles nonlinear dynamics, concurrently operating components and autonomous mode transitions. We demonstrate the robustness of the mixed method empirically.

© 2007 Elsevier B.V. All rights reserved.

**Keywords:** Fault detection; Hybrid systems; Robust sensing; Graceful system degradation

## 1. Introduction

Robotic and embedded systems have become increasingly pervasive in a variety of applications. Space missions, such as Mars Science Laboratory [1] have increasingly ambitious science goals, such as operating for longer periods of time and with increasing levels of onboard autonomy. Here on Earth, robotic assistants, such as CMU's Pearl [2] directly benefit people in ways ranging from providing health care to performing rescue operations.

In order to act robustly in the physical world, robotic systems must handle the uncertainty and partial observability inherent in most real-world situations. Robotic systems often face unpredictable, harsh physical environments and must continue performing their tasks (perhaps at a reduced rate), even when some of their subsystems fail. For example, in land rover missions, such as MSL, the robot needs to detect when one or more of its wheel motors fail, which could jeopardize the

safety of the mission. The rover can detect the failure from a drift in its trajectory and then compensate for the failure, either by adjusting the torque to its other wheels or by replanning its path to the desired goal.

In our previous work we have developed methods for estimating the state of systems that evolve in a discrete manner, where system behavior is modeled by Concurrent Probabilistic Constraint Automata (CPCA), one automaton per component [3,4]. In many situations, however, a purely discrete model is insufficient. Probabilistic hybrid models therefore represent the system with both discrete and continuous state variables that evolve probabilistically according to a known distribution. The discrete state variables typically represent a *behavioral mode* of the system, while the continuous variables represent its *continuous dynamics*. Probabilistic hybrid models can be used to provide an appropriate level of modeling abstraction when purely discrete, qualitative models are too coarse, while purely continuous, quantitative models are too fine-grained.

In this paper, we investigate the problem of estimating the state of systems with probabilistic hybrid models. Given a sequence of control inputs and noisy observations, our goal

\* Corresponding author.

E-mail addresses: [larsb@mit.edu](mailto:larsb@mit.edu) (L. Blackmore), [sfuniak@cs.cmu.edu](mailto:sfuniak@cs.cmu.edu) (S. Funiak), [williams@mit.edu](mailto:williams@mit.edu) (B.C. Williams).

is to estimate the discrete and continuous state of the hybrid system. Probabilistic hybrid models are particularly useful for fault diagnosis, the problem of determining the health state of a system. With hybrid models, fault diagnosis can be framed as a state estimation problem, by representing the nominal and fault modes with discrete variables and the state of the system dynamics with continuous variables.

Our previous work [5] developed the HME (Hybrid Mode Estimation) system, which extended our work on discrete model-based diagnosis for Concurrent Probabilistic Constraint Automata, to reason about probabilistic hybrid models known as Concurrent Probabilistic Hybrid Automata (CPHA) [5]. As with CPCA, CPHA represent the system as a collection of concurrently operating automata, one automaton for each component in the system.

In this paper we present a novel method for hybrid estimation with CPHA that achieves robustness by balancing greedy and stochastic search. The key insight behind the new algorithm is that, in many AI and optimization methods, a combination of stochastic and greedy search methods can be effective in practice. This is analogous to the ‘exploration vs. exploitation’ tradeoff, which has been used with great success in Constraint Satisfaction Problems (CSP) [6] and reinforcement learning [7], for example. Previous methods for hybrid estimation have used either greedy search [8] or stochastic sampling [9]. In this paper we show empirically that these methods can have limited performance depending on whether the belief state is concentrated in a few mode sequences, or is relatively flat. The mixed greedy/stochastic method, by contrast, is robust to changes in the variance of the posterior distribution.

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

The paper is organized as follows. In Section 2 we consider existing methods for estimation with hybrid models. In the general case, inference in hybrid models is NP-hard [10]. Approximate inference using techniques in  $k$ -best Gaussian filtering [8,11] have been effective for hybrid state estimation. These methods represent the system state as a mixture of Gaussians that are enumerated in decreasing order of likelihood. Diagnostic errors can occur, however, if the correct diagnosis is not among the leading set of hypotheses. In this paper, we demonstrate this empirically. An alternative approach is to use stochastic sampling to allow the system to perform greater exploration, rather than performing a purely greedy search. An example of such a method is Particle Filtering, which approximates the posterior state distribution using a finite number of particles [12,13]. Examples of diagnosis systems include [9,14]. These particles are evolved stochastically and resampled based on an appropriately defined

importance weighting. The inefficiency of sampling in high-dimensional spaces has limited the effectiveness of Particle Filtering for state estimation in hybrid systems [15]. To avoid these problems, we therefore propose an efficient stochastic sampling approach for CPHA based on Rao–Blackwellised Particle Filtering. Rao–Blackwellised Particle Filtering exploits a tractable substructure in the underlying model by sampling only a subspace of the system state, while estimating the remainder using an efficient analytical method.

Rao–Blackwellised Particle Filtering has been used for hybrid state estimation in Switching Linear Dynamic Systems (SLDS) [13,16], 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  $d_t$ . In many domains, however, simple Markovian transitions  $p(d_t|d_{t-1})$  are not sufficiently expressive [5,17]. In these domains, the transitions of the discrete variables depend on the continuous state. Such transitions are called *autonomous*. Furthermore, many systems consist of several interconnected components, each of which is in its own behavioral mode. Representing the joint mode of all the components would be inefficient. In these cases, it is desirable to represent the mode with several mode variables. Finally, many systems have nonlinear dynamics. Each of these properties — autonomous transitions, interconnected components, and nonlinear dynamics, are expressed naturally in CPHA.

In Section 3, we extend Rao–Blackwellised Particle Filtering to handle autonomous mode transitions, nonlinearities and concurrency. Applying Rao–Blackwellisation schemes to models with autonomous transitions is difficult, since the discrete and continuous state spaces of these models are closely coupled. The key innovation in our algorithm is that it reuses the continuous state estimates in the importance sampling step of the Particle Filter. We extend the class of autonomous transitions that can be handled over our previous work in [5, 18] to multivariable linear transition guards, in the case of piecewise constant transition distributions, and to piecewise polynomial transition distributions of arbitrary order, in the case of single variables. Our RBPF algorithm handles nonlinear dynamics by using an Extended Kalman Filter [19] or an Unscented Kalman Filter [20]. We presented this innovation in [18], this was also proposed independently by [21].

These developments provide the foundation for a unified treatment of  $k$ -best enumeration and RBPF approaches to hybrid state estimation. Both these approaches represent the belief state by a mixture of Gaussians for a subset of mode trajectories traced by the discrete state. The former approach enumerates the trajectories in best-first order, while the latter evolves them through sampling. While prior work [21] has compared the performance of RBPF to other particle filters, there has been little empirical comparison of RBPF and  $k$ -best methods. Such an analysis is crucial to understanding the trade-offs between the two methods, and to developing a new approach that combines the strengths of both. In Section 6 we carry out the comparison and show that both approaches have limitations, depending on whether the posterior distribution

is concentrated in a few discrete mode trajectories, or is relatively flat across many different trajectories. These results demonstrate the need for a new algorithm that is robust to changes in the variance of the approximated posterior distribution.

In Section 5 we develop such an algorithm. The new algorithm uses Rao–Blackwellised Particle Filtering to generate, stochastically, additional candidates to add to  $k$ -best enumeration that would not have been tracked by a purely greedy approach. The algorithm maintains a set of particles, which are updated using RBPF, and a set of mode trajectories with the highest posterior probability, generated by both RBPF and  $k$ -best successor enumeration. This algorithm makes use of the efficient properties of both  $k$ -best enumeration and RBPF, while being probabilistically sound.

Using a simulated acrobatic robot, we demonstrate that the mixed algorithm is effective for both a concentrated and flat posterior distribution. The mixed algorithm shows a dramatic increase in robustness for a small performance penalty.

## 2. Models and state estimation methods

Probabilistic hybrid models and hybrid estimation methods date back to the 1970s [22] and are useful in many applications, including visual tracking [23] and fault diagnosis [5,24]. In this section, we review a formalism for modeling probabilistic hybrid systems, known as Concurrent Probabilistic Hybrid Automata. We then define the hybrid state estimation problem and outline the existing approach to hybrid state estimation for CPHA, based on greedy enumeration. This lays the groundwork for the new approach, based on Rao–Blackwellised Particle Filtering, which is described in Section 3.

### 2.1. Concurrent Probabilistic Hybrid Automata

We have previously developed Concurrent Probabilistic Hybrid Automata [8], a formalism for modeling engineered systems that consist of a large number of concurrently operating components with nonlinear dynamics. A CPHA model consists of a network of concurrently operating Probabilistic Hybrid Automata (PHA), connected through shared continuous input/output variables. Each PHA represents one component in the system and has both discrete and continuous hidden state variables. The automaton interacts with the other automata in the surrounding world through shared continuous variables, and its discrete state determines the evolution of its continuous variables.

**Definition 1.** A Probabilistic Hybrid Automaton is a tuple  $\langle \mathbf{x}, \mathbf{w}, F, T, X_0, \mathcal{X}_d, \mathcal{U}_d \rangle$  [5]:

- $\mathbf{x}$  denotes the *hybrid state* of the automaton.  $\mathbf{x} \triangleq \mathbf{x}_d \cup \mathbf{x}_c$  where  $\mathbf{x}_d$  denotes the discrete state variables  $\mathbf{x}_d \in \mathcal{X}_d$  and  $\mathbf{x}_c$  denotes the continuous state variables  $\mathbf{x}_c \in \mathbb{R}^{n_x}$ .<sup>1</sup>

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

The transition function  $T(\mathbf{d})$ , for some mode  $\mathbf{d}$ , specifies the transition distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1}, \mathbf{u}_t)$ . Each tuple  $\langle p_\tau, c \rangle \in T(\mathbf{d})$  defines the transition distribution as having a constant value of  $p_\tau$  in the regions satisfied by the guard  $c$ . The transition function can thus specify conditional distributions  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1} = \mathbf{d}, \mathbf{x}_{c,t-1}, \mathbf{u}_t)$  that are piecewise constant in the continuous state  $\mathbf{x}_{c,t-1}$ . While in keeping with previous work we retain this definition for PHA, in Section 4.2 we show that our efficient hybrid estimation method can handle piecewise *polynomial* transition functions.

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

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

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

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

We assume that  $\mathbf{x}_{d1,t}$  and  $\mathbf{x}_{d2,t}$  are independent, conditioned on  $\mathbf{x}_{c,t-1}$ . In other words, the transitions of each component are independent, conditioned on the continuous state.

The overall continuous evolution of the CPHA is determined by taking the union of algebraic and difference equations for

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

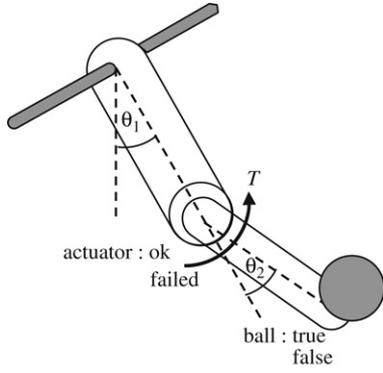


Fig. 1. A two-link acrobatic robot. The robot swings on a bar and may catch a ball of known mass, whenever it is on the right side ( $\theta_1 > 0.55$ ). The body of the robot can be modeled as a PHA, where the hidden continuous state consists of four variables, representing the angles and angular velocities at two joints.

each component PHA. The equations  $F(\mathbf{d})$  are then solved using Groebner bases [25] and causal analysis [26] into the standard form:

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

where  $\mathbf{v}_x$  and  $\mathbf{v}_y$  are the components of  $\mathbf{v}_c$  corresponding to *process noise* and *observation noise*, respectively. We assume that models of real-world systems lead to equations  $F(\mathbf{d})$  that permit a symbolic solver to arrive at the form given in (1); refer to [27] for a more extensive discussion of this topic. A number of conditional independency assumptions are implicit in the state-space form (1). These are shown graphically in Fig. 3. Note that we do *not* assume that discrete transitions are independent of the continuous state.

In this paper we consider the acrobatic robot shown in Figs. 1 through 5. The goal of hybrid estimation here is to filter out the acrobot's hybrid state from a sequence of noisy observations of  $\theta_2$ .

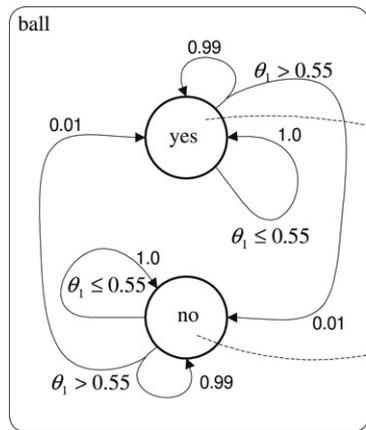


Fig. 2. A PHA for the two-link body of the acrobot system. Left: Transition model for the discrete state **ball** of the body, representing whether or not the robot carries a ball of mass  $m_{\text{ball}}$ . When  $\theta_1 > 0.55$ , there is a probability at each time step of 0.01 that the state of **ball** changes. When  $\theta_1 \leq 0.55$ , the state remains the same. Right: Evolution of the automaton's continuous state, one set of equations for each mode. The continuous dynamics for each mode can be derived using Lagrangian mechanics and turned into a set of discrete-time difference equations, using the Euler approximation. The equations for this system are given in Appendix A.2.

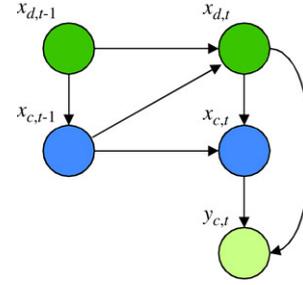


Fig. 3. Conditional dependencies in PHA among the state variables  $\mathbf{x}_c$ ,  $\mathbf{x}_d$  and the output  $\mathbf{y}$ , expressed as a dynamic Bayesian network [28]. The edge from  $\mathbf{x}_{c,t-1}$  to  $\mathbf{x}_{d,t}$  represents the dependence of  $\mathbf{x}_{d,t}$  on  $\mathbf{x}_{c,t-1}$ , that is, autonomous transitions. In hybrid models with Markovian mode transitions, this link is not present.

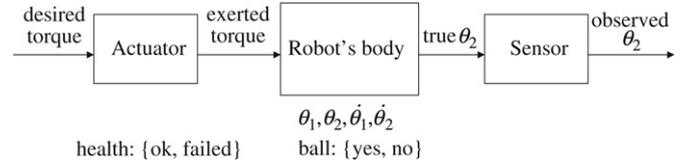


Fig. 4. A composed model for the acrobatic robot in Fig. 1. Each component is modeled with one Probabilistic Hybrid Automaton. The component automata are shown in rectangles, with their state variables shown beneath. Using composition, the PHA for an acrobot body can be augmented using a PHA model of the torque actuator, and a model of the angular position sensor, measuring  $\theta_2$ . The full discrete transition model is shown in Fig. 5. The actuator exerts the commanded torque in the **ok** mode, but exerts zero torque in the **failed** mode. The position sensor is modeled as adding Gaussian white noise to the true value of  $\theta_2$ .

This completes our review of Concurrent Probabilistic Hybrid Automata. We now consider the problem of state estimation for these models.

### 2.2. Hybrid state estimation

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

Continuous dynamics:

$$\theta_1, \theta_2, \omega_1, \omega_2$$

$$\theta_{1,t+1} = \theta_{1,t} + \omega_{1,t} \delta t + v_{\theta_1}$$

$$\theta_{2,t+1} = \theta_{2,t} + \omega_{2,t} \delta t + v_{\theta_2}$$

$$\omega_{1,t+1} = f_{1,\text{yes}}(\theta_{1,t}, \theta_{2,t}, \omega_{1,t}, \omega_{2,t}, T) \delta t + v_{\omega_1}$$

$$\omega_{2,t+1} = f_{2,\text{yes}}(\theta_{1,t}, \theta_{2,t}, \omega_{1,t}, \omega_{2,t}, T) \delta t + v_{\omega_2}$$

$$\theta_{1,t+1} = \theta_{1,t} + \omega_{1,t} \delta t + v_{\theta_1}$$

$$\theta_{2,t+1} = \theta_{2,t} + \omega_{2,t} \delta t + v_{\theta_2}$$

$$\omega_{1,t+1} = f_{1,\text{no}}(\theta_{1,t}, \theta_{2,t}, \omega_{1,t}, \omega_{2,t}, T) \delta t + v_{\omega_1}$$

$$\omega_{2,t+1} = f_{2,\text{no}}(\theta_{1,t}, \theta_{2,t}, \omega_{1,t}, \omega_{2,t}, T) \delta t + v_{\omega_2}$$

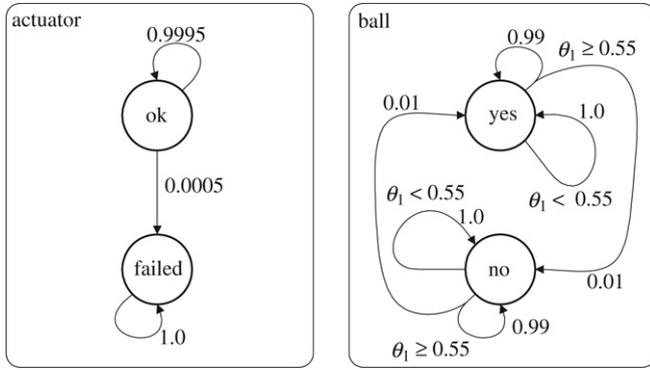


Fig. 5. Discrete transition model for the acrobot CPHA. If the actuator has failed, it exerts no torque. When the robot catches the ball, ball=yes, and the mass of the lower link increases.

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

In general, we can frame the hybrid state estimation problem as that of approximating the posterior distribution  $\langle \mathbf{x}_{d,t}, \mathbf{x}_{c,t} \rangle$  and use this distribution to compute characteristics of interest, such as the MAP mode estimate or continuous state estimate.

### 2.3. $K$ -best enumeration

Existing methods for hybrid estimation with CPHA models have used a  $k$ -best enumeration approach [8]. This section outlines the approach.

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

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

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

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

This decomposition leads to a natural representation of the belief state as a mixture of Gaussians, one for each reachable mode trajectory  $\mathbf{x}_{d,1:t}$ . Given  $\mathbf{x}_{d,1:t}$ , the second term can be approximated as a Gaussian, using a combination of a Kalman Filter and numerical integration techniques, such as Gaussian Quadrature and Exact Monomials [11]. The weight of each mixture component is then computed using the belief state update [18]:

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

In this equation,  $P_T \triangleq p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$  is the prior probability of transitioning to a state  $\mathbf{x}_{d,t}$ , given the

past mode trajectory and past observations; we refer to this as the *transition prior*.  $P_O \triangleq p(\mathbf{y}_t | \mathbf{x}_{d,0:t}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$  is the measurement update. Both  $P_T$  and  $P_O$  can be calculated approximately [8].

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 [10]. Nevertheless, in many domains, efficient inference is possible by employing two strategies: *pruning* (branching) and *collapsing* (merging). 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 [29,30].

One example of a pruning approach that has been shown to be particularly effective in both discrete and hybrid estimation is  $k$ -best filtering [8,30,31].  $K$ -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. Since in CPHA, the transitions of each component are independent conditioned on the continuous state (see Definition 2), an efficient solution is to frame the expansion as a search and solve it using a combination of branch and bound and A\* algorithms [8]. The pseudocode for this approach is shown in Figs. 6 and 7. The  $k$ -best enumeration approach has been shown empirically to be an efficient technique for hybrid state estimation in systems that exhibit autonomous mode transitions, nonlinear dynamics and concurrency [8].

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

## 3. Rao–Blackwellised particle filtering for CPHA

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

### 3.1. Overview of Rao–Blackwellised particle filtering for CPHA

The new approximate RBPF algorithm is our first step in developing a mixed exploitation/exploration strategy for CPHA, and complements our greedy  $k$ -best approach for CPHA. To achieve memory-efficiency, we use particles to represent Gaussian distributions conditioned on a particular mode sequence, and we develop a Gaussian particle filter for CPHA as an instance of a Rao–Blackwellised Particle Filter [13,22].

## 1. Initialization

- create a node  $s$  corresponding to each non-zero value of  $p(\mathbf{x}_{d,0})$
- initialize  $f(s) = -\ln(p(\mathbf{x}_{d,0}))$
- initialize the estimate mean  $\hat{\mathbf{x}}_{c,0}^{(s)} \leftarrow \mathbb{E}[\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(s)}]$
- initialize the estimate covariance  $\mathbf{P}_0^{(s)} \leftarrow Cov(\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(s)})$
- add all nodes  $\mathbf{x}_{d,0}^{(s)}$  to priority queue

2. For  $t = 1, 2, \dots$ 

## (a) A\* Search Step

- **While**  $size(new\_kbest) < k$  **do**
  - Remove node  $s$  from priority queue with lowest  $f = g + h$ :  $s \leftarrow pop\_from\_queue()$
  - **If**  $s$  is a leaf node (has full assignment to component modes and  $P_O$  computed), then
    - \* Add  $s$  to  $new\_kbest$
  - **Else**
    - \* Expand node  $s$  to successors:  $expanded\_nodes \leftarrow expand\_to\_successors(s)$
    - \* Add expanded nodes to priority queue:  $push\_onto\_queue(expanded\_nodes)$

## (b) Normalization Step

- Normalize new  $k$ -best nodes:  $normalize(new\_kbest)$

Fig. 6. Hybrid estimation for CPHA using  $k$ -best enumeration. The variables  $g$  and  $h$  denote the node cost and heuristic cost-to-go, respectively.1. Expand Node  $s$  to Successors

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

Fig. 7. Node expansion for  $k$ -best enumeration algorithm.

Our algorithm is illustrated in Fig. 8. In the spirit of prior approaches to RBPF [13,22], our algorithm exploits the structure in the estimation problem and samples only the discrete mode sequences. Conditioned on each sampled sequence, the new algorithm approximates the associated continuous state distribution as a Gaussian in closed form, using an Extended [19] or an Unscented Kalman Filter [20]. Each particle holds a sample trajectory  $\mathbf{x}_{d,0:t}^{(i)}$  and the corresponding continuous estimate  $(\hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)})$ . This is described in detail in Section 3.3.

The algorithm starts by taking a fixed number of random samples from the initial distribution over the mode variables  $p(\mathbf{x}_{d,0})$  (Step 1). For each sampled mode  $\mathbf{x}_{d,0}^{(i)}$ , the corresponding initial continuous distribution  $p(\mathbf{x}_{c,0}|\mathbf{x}_{d,0}^{(i)})$  is specified by the PHA model. At each time-step, the algorithm

then uses the model to expand the mode sequences of each particle and to update the corresponding continuous estimates (see Fig. 8, Step 2). This is done by first evolving each particle by taking one random sample  $\mathbf{x}_{d,t}^{(i)}$ , for each particle, from a suitably chosen proposal distribution  $q(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ . Intuitively, the proposal is a distribution close to the true distribution that we are trying to determine,  $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ ; that is, the posterior probability of a given mode sequence given all of the observations up to time  $t$ . The posterior distribution is difficult to calculate in closed form, so instead we sample from the proposal distribution, which we choose to be easy to calculate. We then compensate for the discrepancy between the proposal distribution and the true distribution by assigning an importance weight  $w_t^{(i)}$  for each new mode sequence  $\mathbf{x}_{d,0:t}^{(i)}$ . Finally, the resampling

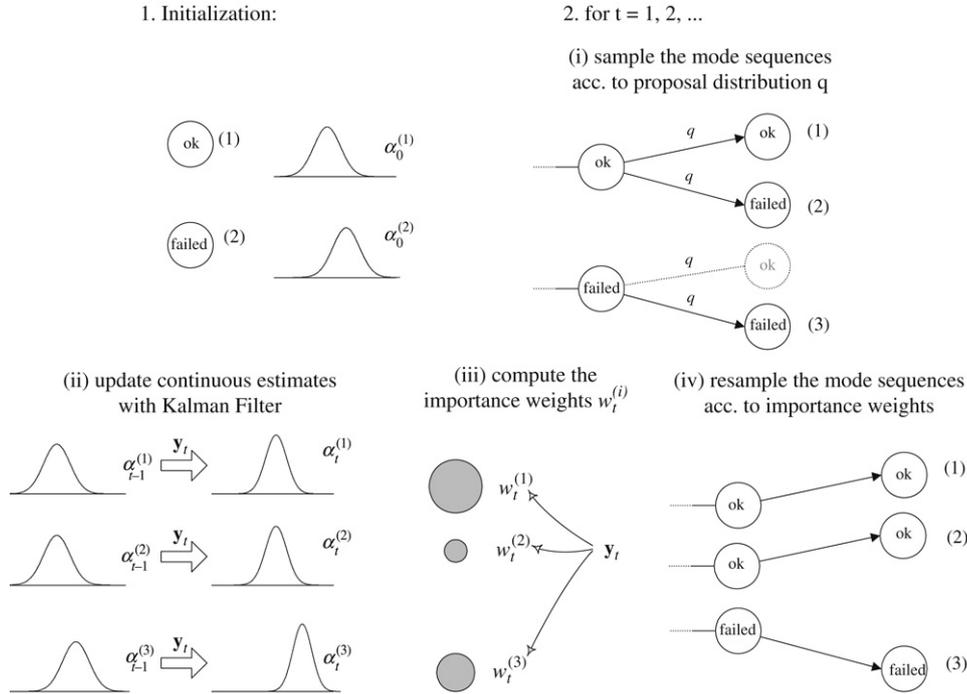


Fig. 8. Rao-Blackwellised Particle Filter for PHA.

step duplicates particles according to their weighting, thereby adjusting the number of particles wherever the proposal distribution does not match the desired distribution [12].

In order to instantiate this algorithm we must define the proposal distribution and the importance weight. For the proposal distribution we choose the distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ , which is the transition prior mentioned in Section 2.3. We show in Section 3.4 that this distribution can be calculated efficiently, even when the model includes autonomous mode transitions. The importance weights then correct for the discrepancy between the proposal and the posterior distribution by taking into account the latest observation  $\mathbf{y}_t$ . In Section 3.6 we show that this can be computed using the Kalman Filter innovation.

Finally, in Section 3.7 the algorithm outlined in Fig. 8 is extended to deal with Concurrent Probabilistic Hybrid Automata.

### 3.2. Rao-Blackwellised particle filtering

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

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

$$\bar{p}_N(\mathbf{x}_t) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_t^{(i)}). \quad (5)$$

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

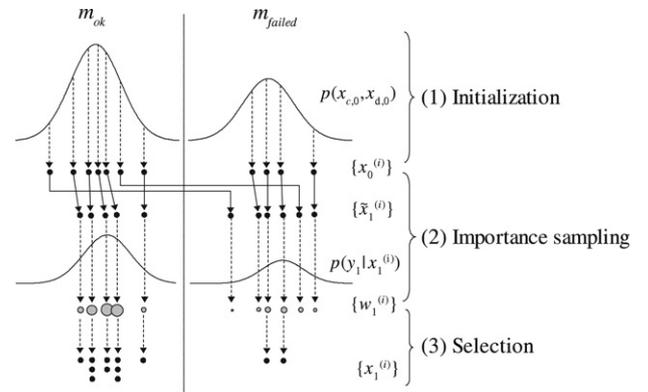


Fig. 9. The three steps of a simple particle filter for a PHA model with one discrete and one continuous variable.

steps, as illustrated in Fig. 9. In the first, initialization step, the algorithm samples the initial distribution  $p(\mathbf{x}_0)$ ; thus, effectively approximating the posterior at  $t = 0$ . Then, in each iteration, the particles  $\mathbf{x}_{0:t}^{(i)}$  are evolved by taking *one* random sample  $\mathbf{x}_t^{(i)}$  from an appropriately chosen *proposal distribution*, and by assigning importance weights that account for the differences between the proposal and the posterior distributions. The final step selects a number of offspring 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 [32] and abstraction [33]. One particularly effective method is Rao-Blackwellised Particle Filtering [12,22,34]. This method

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

Fig. 10. Generic RBPF algorithm [35].

is based on a fundamental observation that with some estimation problems, a particular part of the desired distribution can be determined efficiently without using a sampling approach. By factoring out this portion, we obtain a more efficient approach that only samples the remaining variables.

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

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) &= p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \\ &= p(\mathbf{s}_{0:t}|\mathbf{r}_{0:t}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})p(\mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}). \end{aligned} \quad (6)$$

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 both  $\mathbf{s}_{0:t}$  and  $\mathbf{r}_{0:t}$ . Intuitively, far fewer particles will be needed in this way to reach a given precision of the estimate, since for each sampled sequence  $\mathbf{r}_{0:t}$ , the corresponding state space  $\mathbf{s}$  is covered by an analytical solution, rather than a finite number of samples.

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

$$p(\mathbf{s}_{0:t}, \mathbf{r}_{0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t}) \approx \sum_1^N \alpha_t(i) \delta(\mathbf{r}_{0:t} - \mathbf{r}_{0:t}^{(i)}). \quad (7)$$

A generic RBPF method is outlined in Fig. 10, and, except for the initialization and the addition of the exact step, it is identical to the particle filter, illustrated in Fig. 9. Under weak

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

assumptions, the Rao–Blackwellised estimate converges to the estimated value as  $N \rightarrow +\infty$ , with a variance smaller than the non-Rao–Blackwellised particle filtering method with the same number of particles [15]. Hence the RBPF is superior given a fixed number of particles; however, the run-time performance of the filter will depend on the cost of the exact update for  $\alpha_t^{(i)}$ .

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

### 3.3. Rao–Blackwellisation for hybrid estimation with PHA

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

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

The continuous estimate for each new mode sequence  $\mathbf{x}_{d,0:t}^{(i)}$  is updated as shown in Fig. 8(ii). Since in a PHA, each mode assignment  $\mathbf{d}$  over the variables  $\mathbf{x}_d$  is associated with transition and observation functions:

$$\begin{aligned} \mathbf{x}_{c,t} &= \mathbf{f}(\mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}, \mathbf{d}) + \mathbf{v}_x(\mathbf{d}) \\ \mathbf{y}_t &= \mathbf{g}(\mathbf{x}_{c,t}, \mathbf{u}_t, \mathbf{d}) + \mathbf{v}_y(\mathbf{d}), \end{aligned} \quad (8)$$

we update each estimate  $\langle \hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)} \rangle$  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  $\langle \hat{\mathbf{x}}_{c,t}^{(i)}, P_t^{(i)} \rangle$ .

We now consider the approximation error introduced by using an Extended or Unscented Kalman Filter in the presence of autonomous mode transitions. Such transitions can be used to model severe nonlinearities, and it is possible to construct cases where the error between the Kalman Filter approximation and the true distribution is large. In this case, the only alternative approach is to use a non-Rao–Blackwellised Particle Filter at greatly increased computational cost. However under relatively weak assumptions, the Kalman Filter approach developed in this section is an effective compromise that introduces relatively small error at low computational cost. Extensive empirical validation by [21,36] has shown that for fault diagnosis in real-world systems, with autonomous mode transitions, a Rao–Blackwellised approach has a dramatically lower error rate than a normal Particle Filter operating with the same computational resources. In addition to this, in Appendix A.1 we analyze the Kullback–Liebler divergence between the true distribution, and the Kalman Filter approximation, and show that the divergence is reasonably small in most cases.

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

### 3.4. Proposal distribution

In particle filtering, the proposal distribution is chosen to be one that is close to the desired distribution, but that can also be calculated efficiently in closed form. In this section we specify the proposal distribution  $q(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$  and show how it is calculated efficiently, while taking into account autonomous mode transitions.

An autonomous mode transition is a form of guarded transition for which the transition distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$  depends explicitly on the continuous state  $\mathbf{x}_{c,t-1}$ . In PHA, the transition distribution is specified as a finite set of guard conditions  $c$  and their associated transition probabilities  $p_\tau$ . Each guard condition specifies a region over the continuous state and automaton’s input/output variables, for which  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) = p_\tau$ . Hence the transition distribution is piecewise constant over  $\mathbf{x}_{c,t-1}$  (see Fig. 11).

We choose the proposal distribution to be  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1} = \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ . This distribution expresses the probability of the transition from the mode  $\mathbf{x}_{d,0:t-1}^{(i)}$  to each mode  $\mathbf{x}_{d,t} \in \mathcal{X}_d$  and is similar in its form to the transition distribution  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$  in a Markov process. However, it is conditioned on a complete discrete state sequence and all previous observations and control actions, rather than simply on the previous state. This is because  $\{\mathbf{x}_{d,t}\}$  alone is *not* an HMM process: due to the autonomous transitions, knowing only  $\mathbf{x}_{d,t-1}$  does not tell us what the distribution of  $\mathbf{x}_{d,t}$  is. The distribution of  $\mathbf{x}_{d,t}$  is known only when conditioned on the mode *and* the continuous state for the previous time step (see Fig. 3). Since the continuous state must be estimated, rather than being observed directly, autonomous transitions make calculation of the proposal challenging.

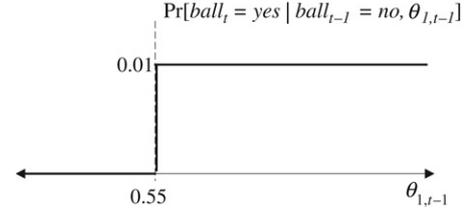


Fig. 11. Probability of a mode transition  $\text{ball=no}$  to  $\text{ball=yes}$  as a function of  $\theta_{1,t-1}$  for the  $\text{ball}$  PHA in Fig. 2.

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

$$\begin{aligned} p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) &= \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) d\mathbf{x}_{c,t-1} \\ &= \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{x}_{c,t-1}, \mathbf{u}_{0:t}) \\ &\quad \times p(\mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) d\mathbf{x}_{c,t-1} \\ &= \int_{\mathbf{x}_{c,t-1}} p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) \\ &\quad \times p(\mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}_{c,t-1}. \end{aligned} \quad (9)$$

Here, the third equality comes from the independence assumptions made in the model; as shown in Fig. 3, the distribution of  $\mathbf{x}_{d,t}$  is independent of the observations  $\mathbf{y}_{1:t-1}$  and mode assignments prior to time  $t-1$ , given the state at time  $t-1$ .

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

Typically, when performing Rao–Blackwellisation, the integral in (9) is difficult to evaluate efficiently [35]. In this section we show that for PHA, however, efficient evaluation of this integral is possible.

For the piecewise constant transition distribution in PHA, the left term in the integral in (9),  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$  takes on only a finite number of values  $p_{\tau_j}(\mathbf{x}_{d,t})$ . Hence we can split the integral domain into the sets  $X_j$  that satisfy the guard condition  $c_j$  and factor out the transition probability  $p_{\tau_j}$ :

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

$$\begin{aligned}
&= \sum_j \int_{X_j} p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1}) \\
&\quad \times p(\mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}_{c,t-1} \\
&= \sum_j p_{\tau_j}(\mathbf{x}_{d,t}) \\
&\quad \times \int_{X_j} p(\mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}_{c,t-1} \\
&= \sum_j p_{\tau_j}(\mathbf{x}_{d,t}) \Pr_{\alpha_{t-1}^{(i)}}[X_j]. \tag{10}
\end{aligned}$$

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

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

### 3.5. Evaluating the probability of satisfying transition guards

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

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

$$\begin{aligned}
\Pr_{\alpha_t^{(i)}}[X_j] &\approx \frac{1}{(2\pi)^{n_c/2} |\mathbf{P}_{t-1}^{(i)}|^{1/2}} \\
&\quad \times \int_{X_j} e^{-\frac{1}{2}(\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})^T \mathbf{P}_{t-1}^{(i)-1} (\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})} d\mathbf{x}_c. \tag{11}
\end{aligned}$$

When the guard conditions are of the form  $x < c$  or  $x \leq c$ , for some constant  $c$ , such as  $\theta_1 < 0.55$ , the integral in (11) simplifies to evaluating the cumulative density function  $D(c)$  of the normal variable  $\mathcal{N}(\mu, \sigma^2)$ , where  $\mu = (\hat{\mathbf{x}}_{c,t-1}^{(i)})_x$  is the mean of variable  $x$  in  $\hat{\mathbf{x}}_{c,t-1}^{(i)}$  and  $\sigma^2 = (\mathbf{P}_t^{(i)})_x$  is its variance Fig. 12:

$$D(c) \triangleq \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^c e^{-(x-\mu)^2/(2\sigma^2)} dx. \tag{12}$$

The cumulative density function  $D(c)$  is evaluated using

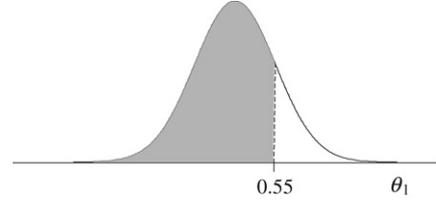


Fig. 12. Evaluating single-variate guard conditions.

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  $[l, u]$ ,<sup>4</sup> 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,  $D(u) - D(l)$ .

We have previously used this technique with our  $k$ -best enumeration approach for CPHA [5]. Using this approach for RBPF, we are able to calculate the proposal distribution in Section 3.4 efficiently for a particular class of autonomous mode transitions; those whose guard conditions are intervals over a single variable. In Section 4.1 we generalize this method to apply to CPHA with multi-variate linear guard conditions.

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

### 3.6. Importance weights

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

<sup>3</sup> Guard conditions involving discrete or continuous input variables, such as  $c_c(\mathbf{x}_c) \wedge c_d(\mathbf{u}_d)$ , are handled by setting  $\Pr_{\alpha_{t-1}^{(i)}}[X_j] \equiv 0$  whenever  $c_d(\mathbf{u}_d)$  is not satisfied. More complex guards are transformed to a number of simpler guards using elementary rules of logic.

<sup>4</sup> 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.

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

$$w_t^{(i)} \triangleq \frac{p(\mathbf{y}_t | \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{0:t-1}) p(\mathbf{x}_{d,t}^{(i)} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{0:t-1})}{q(\mathbf{x}_{d,t}^{(i)}, \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{0:t})} \\ = p(\mathbf{y}_t | \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t}). \quad (13)$$

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

In the Kalman Filter predict/measurement cycle, the Gaussian distribution  $\alpha_{t-1}^{(i)} = \mathcal{N}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)})$  is propagated through the continuous transition and observation functions for mode  $\mathbf{x}_{d,t}^{(i)}$  (8). For SLDS models, this gives a Gaussian distribution for the observed value  $\mathbf{y}_t$ , with mean  $\mathbf{y}_p$  and covariance  $\mathbf{S}_t^{(i)}$ . The Kalman Filter *innovation*,  $\mathbf{r} = \mathbf{y}_t - \mathbf{y}_p$ , is defined as the difference between the expected observation and the actual observation. For SLDS models, the observation likelihood in (13) is calculated *exactly* as [29]:

$$w_t^{(i)} = \frac{1}{(2\pi)^{N/2} |\mathbf{S}_t^{(i)}|^{1/2}} e^{-0.5 \mathbf{r}^T (\mathbf{S}_t^{(i)})^{-1} \mathbf{r}}. \quad (14)$$

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  $\alpha_{t-1}^{(i)} = p(\mathbf{x}_{c,t-1} | \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{0:t-1}, \mathbf{u}_{0:t})$  is no longer strictly Gaussian. Nevertheless, if we approximate it with the estimated Gaussian  $\mathcal{N}(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)})$ , as we have done in the previous subsections, we can compute the weight in (13) from the Extended [19] or the Unscented [20] 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  $\mathbf{x}_{d,t}^{(i)}$ :

$$\hat{\mathbf{x}}_{c,t}^{(i-)} = \mathbf{f}(\hat{\mathbf{x}}_{c,t-1}^{(i-)}, \mathbf{u}_{t-1}) \quad (15)$$

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_c} |_{\hat{\mathbf{x}}_{c,t}^{(i-)}} \quad (16)$$

$$\mathbf{P}_t^{(i-)} = \mathbf{A} \mathbf{P}_{t-1}^{(i-)} \mathbf{A}^T + \mathbf{Q}, \quad (17)$$

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

$$\mathbf{y}_p = \mathbf{g}(\hat{\mathbf{x}}_{c,t}^{(i-)}, \mathbf{u}_t) \quad (18)$$

$$\mathbf{C} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}_c} |_{\hat{\mathbf{x}}_{c,t}^{(i-)}} \quad (19)$$

$$\mathbf{S}_t^{(i)} = \mathbf{C} \mathbf{P}_t^{(i-)} \mathbf{C}^T + \mathbf{R}, \quad (20)$$

where  $\mathbf{R} = \text{cov}(\mathbf{v}_y(\mathbf{x}_{d,t}^{(i)}))$  is the observation noise in mode  $\mathbf{x}_{d,t}^{(i)}$ . The observation likelihood in (13) can then be approximated with normal p.d.f.

$$w_t^{(i)} = \frac{1}{(2\pi)^{N/2} |\mathbf{S}_t^{(i)}|^{1/2}} e^{-0.5 \mathbf{r}^T (\mathbf{S}_t^{(i)})^{-1} \mathbf{r}}. \quad (21)$$

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

### 3.7. Rao–Blackwellised Particle Filtering for CPHA

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

In CPHA, component transitions are conditionally independent, given the current discrete and continuous state (Definition 2). Therefore, it is possible to compute the transition probabilities  $P_{\mathcal{T}}^{(i)}$  for each tracked mode sequence componentwise [5,37]. 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.

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

$$w_t^{(i)} = p(\mathbf{y}_t | \mathbf{x}_{d,0:t}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t}) \triangleq P_{\mathcal{O},t}^{(i)}. \quad (22)$$

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  $\mathcal{X}_{d,j}$ , and obtain the joint transition distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$  as a product of component transition distributions  $\prod_j p(\mathbf{x}_{d,j,t} | \mathbf{x}_{d,j,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$ , for all components  $j$  in the model (see Section 3.4).

The final algorithm is shown in Fig. 13.<sup>5</sup> To summarize, using a Rao–Blackwellised Particle Filtering approach, this

<sup>5</sup> Note that, compared to the generic RBPF algorithm in Fig. 10, the importance weight is now calculated after the exact step, because the innovation mean and covariance, computed in the Kalman Filter update step, are used to compute the importance weight.

1. Initialization
  - For  $i = 1, \dots, N$ 
    - 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  $t = 1, 2, \dots$ 
  - (a) Importance sampling step
    - For  $i = 1, \dots, N$ 
      - For each component  $k$ 
        - \* compute the transition distribution  $p(\mathbf{x}_{dk,t} | \mathbf{x}_{dk,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
        - \* sample  $\mathbf{x}_{d,t}^{(i)} \sim p(\mathbf{x}_{dk,t} | \mathbf{x}_{dk,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$
      - let  $\mathbf{x}_{d,0:t}^{(i)} \leftarrow (\mathbf{x}_{d,0:t-1}^{(i)}, (\mathbf{x}_{d,t}^{(i)}))$
    - (b) Exact step
      - For  $i = 1, \dots, N$ 
        - perform a KF update:  $\hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)}, \mathbf{r}_t^{(i)}, \mathbf{S}_t^{(i)} \leftarrow UKF(\hat{\mathbf{x}}_{c,t-1}^{(i)}, \mathbf{P}_{t-1}^{(i)}, \mathbf{x}_{d,t}^{(i)})$
        - 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  $\{(\mathbf{x}_{d,0:t}^{(i)}, \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)})\}$  according to the normalized weights  $\{w_t^{(i)}\}$  to obtain particles  $\{(\mathbf{x}_{d,0:t}^{(i)}, \hat{\mathbf{x}}_{c,t}^{(i)}, \mathbf{P}_t^{(i)})\}$

Fig. 13. Rao–Blackwellised Particle Filter for CPHA.

algorithm is able to estimate efficiently the hybrid state of Concurrent Probabilistic Hybrid Automata, which have autonomous mode transitions, nonlinear dynamics, and many concurrently operating components.

#### 4. Generalizing autonomous mode transitions

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

##### 4.1. Multi-variate guard conditions

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

In the case of CPHA, this probability can be expressed as a sum over a finite number of terms, where each term is a product of the probability  $\Pr_{\alpha_{t-1}^{(i)}} [X_j]$  of a guard condition  $c_j$  being satisfied, and the corresponding transition probability  $p_{\tau_j}$  (10). The remaining challenge is to calculate  $\Pr_{\alpha_{t-1}^{(i)}} [X_j]$ . Our previous work showed how this can be carried out efficiently for guard conditions that are single-variate intervals [8,18]. In this section we show that this can be generalized to rectangular and linear multi-variate guard conditions.

Multi-variate guard conditions are often needed to represent more complex constraints on transitions than can be handled

by single-variate interval conditions. In general, the rectangular multi-variate guard conditions will take the form  $\bigwedge_{i=1}^{n_x} (x_i \in [l_i, u_i])$ , where  $x_i$  are distinct continuous state variables in  $\mathbf{x}_c$ . Evaluating the probability of such a multi-variate guard condition amounts to evaluating the multi-dimensional (hyper)rectangular integral over a Gaussian distribution:

$$\Pr_{\alpha_t^{(i)}} [X_j] \approx \frac{1}{(2\pi)^{n_x/2} |\mathbf{P}_{t-1}^{(i)}|^{1/2}} \times \int_{l_{i_1}}^{u_{i_1}} \int_{l_{i_2}}^{u_{i_2}} \dots \int_{l_{i_n}}^{u_{i_n}} e^{-\frac{1}{2}(\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})^T \mathbf{P}_{t-1}^{(i)-1} (\mathbf{x}_c - \hat{\mathbf{x}}_{c,t-1}^{(i)})} d\mathbf{x}_c. \quad (23)$$

Rectangular integrals over Gaussian distributions are evaluated efficiently using numerical methods, such as those presented in [38–40]. The method of [38] achieves errors of the order of  $10^{-5}$  with computation times of the order of  $10^{-4}$  s for integrals of dimension 5. This efficiency justifies the use of such algorithms in a Rao–Blackwellisation scheme.

Sometimes, transition guards are best represented by a linear combinations of continuous variables. For example, in a two-tank system, the direction of the flow between the two tanks depends on the heights in the two tanks. Hence, the mode variable for the flow direction would be guarded by the linear guards  $h_1 - h_2 > 0$  and  $h_1 - h_2 < 0$  (see Fig. 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 by one dimension, and would make the covariance matrix singular. A singular covariance matrix prevents the Kalman Filter update essential to the Rao–Blackwellised Particle Filter from being carried out. Instead, we apply a linear transform to the Gaussian distribution, thus reducing the computation to

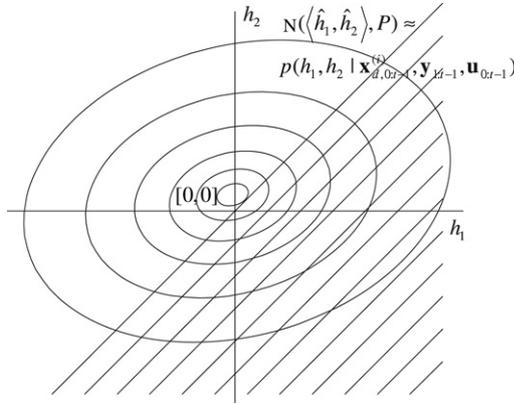


Fig. 14. Linear guard condition  $h_2 < h_1$  over the Gaussian approximation of the posterior density of  $h_1$  and  $h_2$ .

an instance of the rectangular multi-variate integral in (23). Suppose that the guard condition  $c$  is expressed as a conjunction of clauses  $\bigwedge_{i=1}^{n_x} l_i < \mathbf{a}_i \mathbf{x}_c < u_i$ , where  $\mathbf{a}_i$  is the vector of *guard coefficients* that specify condition  $c_i$ . Such guard conditions correspond to a convex space that is formed as an intersection of hyper-planes  $l_i < \mathbf{a}_i \mathbf{x}_c$  and  $\mathbf{a}_i \mathbf{x}_c < u_i$ . Let  $\mathbf{A} \triangleq [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_{n_x}]^T$  be the square matrix of the guard coefficients and define  $\mathbf{z} \triangleq \mathbf{A} \mathbf{x}_{c,t-1}$  as the derived vector with  $n_x$  elements. Then the guard condition  $c \triangleq \bigwedge_{i=1}^{n_x} l_i < \mathbf{a}_i \mathbf{x}_c < u_i$  is equivalent to the guard condition  $\bigwedge_{i=1}^{n_x} l_i < \mathbf{z}_i < u_i$ . The probability of the guard condition  $c$  being satisfied can thus be evaluated as an integral

$$\int_{l_1}^{u_1} \int_{l_2}^{u_2} \dots \int_{l_{n_x}}^{u_{n_x}} p(\mathbf{z}_t | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}_c, \quad (24)$$

over the rectangular region  $[l_1, u_1] \times [l_2, u_2] \times \dots \times [l_{n_x}, u_{n_x}]$ . The covariance of  $\mathbf{z}$  is given by  $\mathbf{A} \mathbf{P}_{t-1}^{(i)} \mathbf{A}^T$ , and furthermore this covariance is nonsingular as long as  $\mathbf{P}_{t-1}^{(i)}$  is nonsingular and  $\mathbf{A}$  is nonsingular; we need consider only cases where this is true.<sup>6</sup> Note that this approach is restricted to the case where the number of guard conditions is at most the same as the number of continuous state variables  $n_x$ .<sup>7</sup>

Therefore, the key result is that the linear guard conditions can once again be evaluated as a rectangular integral over a Gaussian distribution.

#### 4.2. Polynomial transition distributions

The PHA formalism specifies a finite set of guarded transitions between discrete modes, each with a constant transition probability, given that the guard condition is satisfied. Hence the transition distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$  is piecewise constant in  $\mathbf{x}_{c,t-1}$  (Fig. 11). We now present a method for calculating  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$ , i.e. the transition prior,

<sup>6</sup> A singular  $\mathbf{A}$  matrix corresponds to the case where there are redundant guard conditions, in which case an equivalent nonsingular  $\mathbf{A}$  can be calculated.

<sup>7</sup> A number of guard conditions greater than  $n_x$  leads to a singular covariance for  $\mathbf{z}$ . Fewer guard conditions than  $n_x$  is handled trivially by setting  $l_i = -\infty$  and  $u_i = \infty$  for unused guard conditions.

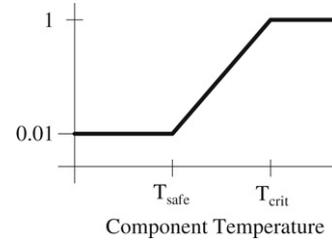


Fig. 15. Probability of transition into failed mode for a simple component as a function of temperature. Below the safe temperature,  $T_{\text{safe}}$ , there is a small failure probability. Above this, the failure probability increases linearly until  $T_{\text{crit}}$ , at which point failure is guaranteed.

when the transition distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, \mathbf{x}_{c,t-1}, \mathbf{u}_{t-1})$  is *not* piecewise constant. In particular, we show that the integral in (9) can be calculated efficiently when the transition distribution is described by a piecewise polynomial function of arbitrary order, for transition distributions defined over a single variable. An example of a piecewise polynomial transition distribution is shown in Fig. 15. This contribution greatly expands the class of models about which our hybrid estimation approaches can reason, since piecewise polynomial functions of arbitrary order can approximate any piecewise smooth function to arbitrary accuracy.

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

$$p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) = \int_x p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) \times p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) dx. \quad (25)$$

As in Section 3.5 we approximate  $p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1})$  using a Gaussian distribution with mean  $\mu_i$  and covariance  $\sigma_i$ . As before, we assume the cumulative distribution function  $D(c)$ , given by (12), can be calculated efficiently.

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

$$p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) = a_{0,j} + a_{1,j}x \quad x \in X_j. \quad (26)$$

When calculating the transition prior in (25), the integral can be split into a finite number of integrals of the following form:

$$\begin{aligned} & \int_x p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) \\ & \times p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) dx \\ & = \frac{1}{\sigma_i \sqrt{2\pi}} \sum_j \int_{X_j} (a_{0,j} + a_{1,j}x) e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx. \end{aligned} \quad (27)$$

The integral for each guard condition can be rewritten as:

$$\int_{X_j} a_{1,j}(x - \mu_i) e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx + \int_{X_j} (a_{0,j} + a_{1,j}\mu_i) e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx. \quad (28)$$

We now use the result that the integral

$$\int_l^u (x - \mu_i) e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx = \left[ -\sigma_i^2 e^{-(x-\mu_i)^2/(2\sigma_i^2)} \right]_l^u, \quad (29)$$

can be calculated in closed form using a substitution of the form  $v = (x - \mu_i)^2$ . Assuming the guard region  $X_j$  takes the form  $x \in [l, u]$  this means the first term in (28) can be calculated in closed form. The second term in (28) is an interval integral over the Gaussian p.d.f. for  $x$ , and this is evaluated using the cumulative distribution function  $D(c)$ , as in (12). Hence for linear transition functions, the transition prior can be calculated efficiently.

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

$$p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) = \sum_{q=0}^{n_p} a_{q,j} x^q \quad x \in X_j, \quad (30)$$

where  $n_p$  is the order of the polynomial. The transition prior in (25) can now be decomposed further, according to each term of the polynomial:

$$\int_x p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) \times p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) dx = \sum_j \sum_i \int_{X_j} a_{q,j} x^q p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) dx. \quad (31)$$

The polynomial in (30) can be rewritten as:

$$\sum_{q=0}^{n_p} a_{q,j} x^q = \sum_{q=0}^{n_p} a'_{q,j} (x - \mu_i)^q. \quad (32)$$

The translated coefficients  $a'_{q,j}$  are calculated by comparing coefficients in the polynomial (32) to give  $n_p + 1$  equations of the form:

$$a_q = \sum_{s=0}^{n_p-q} (-\mu_i)^s \binom{s+q}{q} a'_{s+q} \quad q = 0, \dots, n_p. \quad (33)$$

These equations are solved quickly online by direct substitution starting from  $q = n_p$  and ending at  $q = 0$ .<sup>8</sup> Using (32) the transition prior (31) can be written as:

$$\int_x p(\mathbf{x}_{d,t} | \mathbf{x}_{d,t-1}^{(i)}, x, \mathbf{u}_{t-1}) p(x | \mathbf{x}_{d,0:t-1}^{(i)}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t-1}) dx = \sum_j \sum_q \frac{1}{\sigma_i \sqrt{2\pi}} \int_{X_j} a'_{q,j} (x - \mu_i)^q e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx. \quad (34)$$

<sup>8</sup> To see this, note that, for  $q = n_p$ , the right hand side of (33) is exactly  $a'_{n_p}$ , making calculation of  $a'_{n_p}$  trivial. For  $q = n_p - 1$ , the right hand side of (33) has only terms with  $a'_{n_p}$  and  $a'_{n_p-1}$ , allowing us to solve for  $a'_{n_p-1}$ ; this process continues until  $q = 0$ .

By repeated integration by parts, the more general form of the integral in (29),

$$\int_l^u (x - \mu_i)^q e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx = \left[ -\sigma_i^2 (x - \mu_i)^{q-1} e^{-(x-\mu_i)^2/(2\sigma_i^2)} \right]_l^u + \int_l^u (q-1)(x - \mu_i)^{q-2} \sigma_i^2 e^{-(x-\mu_i)^2/(2\sigma_i^2)} dx, \quad (35)$$

can be reduced to a summation of closed form terms, plus either the integral in (29), in the case of odd  $q$ , or the integral in (12), in the case of even  $q$ . In the first case the integral is evaluated entirely in closed form, while in the second case the only non-closed form term is the integral over the p.d.f., which is evaluated using  $D(c)$ . Furthermore, all of the terms that cannot be evaluated in closed form are scaled versions of the same integral over the p.d.f.; hence,  $D(c)$  needs only to be evaluated at the boundaries of the regions  $X_j$ .

Thus, for transition distributions over single variables, the transition prior is calculated efficiently for transition distributions that are piecewise polynomial. This contribution greatly expands the class of models about which both Rao–Blackwellised particle filtering and existing  $k$ -best methods for CPHA can reason, since piecewise polynomial functions of arbitrary order can approximate any piecewise smooth function to arbitrary accuracy. Note that this approach requires  $\mathcal{O}(n_p^3)$  simple operations and at most two single-variate Gaussian c.d.f. table lookups for each region  $X_j$ ; for reasonably small  $n_p$  this is far fewer operations than would be required in evaluating (25) through direct numerical integration to a suitable level of accuracy.<sup>9</sup>

## 5. Hybrid estimation using a mixed stochastic/greedy method

### 5.1. A unified treatment of $k$ -best enumeration and RBPF

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

<sup>9</sup> While direct numerical integration would enable polynomial transition distributions in *multiple* variables, the authors are not aware of any results analogous to those used in the piecewise constant case (Section 4.1) that enable efficient evaluation of such integrals for general piecewise polynomial functions.

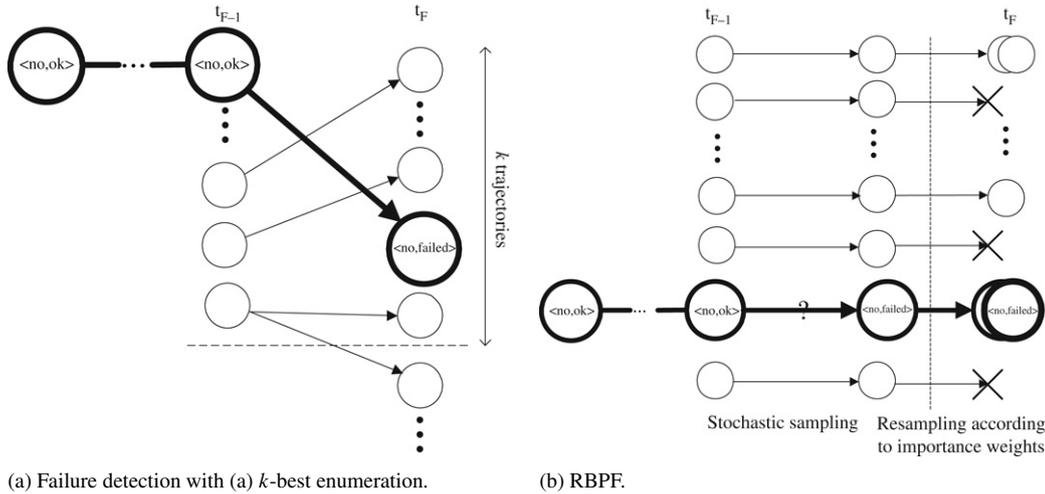


Fig. 16. In (a), the trajectories are shown ordered in terms of their posterior probability, with the most likely at the top. Trajectories below the dashed line are discarded. The true mode trajectory is shown in bold. In (b), the question mark indicates that the generation of the successor corresponding to the true trajectory is generated stochastically.

Filter. In addition, both methods use the same approach to evaluating the transition prior  $p(\mathbf{x}_{d,t}|\mathbf{x}_{d,0:t-1}, \mathbf{y}_{1:t-1}, \mathbf{u}_{0:t})$  in the presence of autonomous mode transitions.

Finally, both methods approximate the posterior belief state by tracking only a subset of the reachable mode trajectories. The key difference between the two methods is the approach used to select which mode trajectories to track;  $k$ -best enumeration greedily selects the  $k$  trajectories with the highest posterior probability  $p(\mathbf{x}_{d,0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ , while RBPF uses stochastic sampling. To gain insight into the relative behavior of these two methods, and their appropriate combination, consider the acrobatic robot introduced in Section 2. Suppose that the robot does not carry a ball, and its actuator is functional, up to time step  $t_F$ , when a failure occurs. As illustrated in Fig. 16(a), the  $k$ -best enumeration algorithm maintains the nominal trajectory, and will detect the failure, as long as a trajectory with the fault transition is among the set of leading trajectories at (or near) the time when the actual fault occurs. By contrast, with the RBPF, mode transitions are sampled stochastically (Fig. 16(b)). Since the transition into the failure state has a low prior, many particles will be needed in order to detect the fault.

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

*other candidate trajectories*. This behavior makes the RBPF more robust to local maxima.

However by representing the posterior probability  $p(\mathbf{x}_{d,0:t}|\mathbf{y}_{1:t}, \mathbf{u}_{0:t})$  using the number of repeated particles, RBPF introduces additional approximation and typically reduces the number of distinct trajectories tracked at any time. This means that for a relatively concentrated posterior,  $k$ -best enumeration will typically outperform RBPF. These insights are validated empirically in Section 6 using the simulated acrobatic robot.

These observations motivate the development of a new algorithm that combines  $k$ -best enumeration and RBPF in a greedy and stochastic search. In Section 5.2 we introduce the new algorithm and show that it exploits the differences between  $k$ -best enumeration and RBPF in order to increase the robustness of the algorithm to changes in the variance of the posterior distribution.

## 5.2. A novel combined greedy/stochastic algorithm

In this section we present a novel algorithm that combines the greedy and stochastic approaches, of  $k$ -best enumeration and RBPF respectively, to explore the discrete mode trajectories of the system (Fig. 17). The algorithm maintains two sets of trajectories: one set of stochastically generated trajectories, updated with RBPF, and a separate set of leading trajectories, enumerated according to their posterior probability. The key idea is to generate successors to the leading  $k$  trajectories through *both* previously deterministically generated successors to the current  $k$  leading trajectories and the candidates generated by the RBPF. In this manner, the true trajectory that was discarded by simple  $k$ -best enumeration on the basis of having a lower posterior probability than other trajectories can still be tracked in the RBPF particle set and included in the deterministic set at a later time. In Section 6 we show empirically that the new algorithm is more robust than  $k$ -best and RBPF taken individually, with only a minor performance penalty.

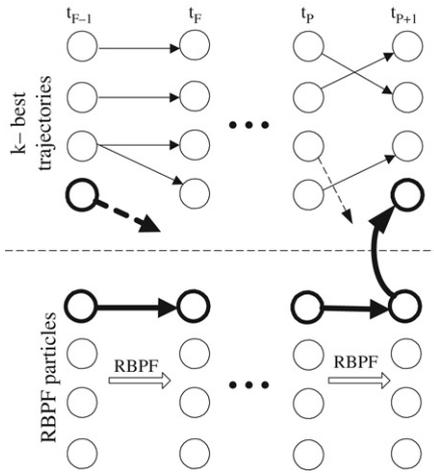


Fig. 17. Failure detection with our mixed method. At time  $t_F$  the true trajectory (bold line) is no longer among the trajectories with the  $k$  highest posteriors. However, the true trajectory is sampled stochastically and becomes a member of the RBPF particle set. At time  $t_{p+1}$ , the posterior of the true trajectory becomes large enough for it to be included in the leading  $k$  trajectories.

#### 1. Initialization

- Initialize Rao-Blackwellised Particle Filter with  $N - k$  particles
- Initialize  $k$ -best trajectories

#### 2. For $t = 1, 2, \dots$

##### (a) RBPF update

- Do RBPF update and add candidates to priority queue

##### (b) A\* search for successors with highest posterior

- Add current trajectories to priority queue
- **While**  $\text{size}(\text{new\_kbest}) < k$  **do**
  - Remove candidate from priority queue
  - If candidate is a goal then
    - \* Pop candidates until a unique one is found
    - \* Add unique candidate to  $\text{new\_kbest}$
  - **Else**
    - \* Expand candidate to its successors
    - \* Add successors to priority queue
- Normalize new  $k$ -best trajectories

Fig. 18. Belief state update using greedy and stochastic search.

The combination of greedy and stochastic search introduces two challenges. First, in order to enable the comparison of the two sets of trajectories, based on the posterior probability, each particle needs to be augmented with  $b(\mathbf{x}_{d,0:t})$ , the posterior probability of a mode sequence  $\mathbf{x}_{d,0:t}$ . This can be updated with (4), by reusing the values  $P_T$  and  $P_O$ , computed in the importance sampling step of the RBPF.

Second, trajectories generated by both greedy and stochastic search must be combined to give the belief state maintained by the mixed algorithm. Both techniques maintain a number of trajectories and a Gaussian distribution over the continuous state, conditioned on each trajectory  $\mathbf{x}_{d,0:t}$ . The belief state representation in the new algorithm is a mixture of Gaussians, obtained by summation over the  $k$  trajectories with the highest  $b(\mathbf{x}_{d,0:t})$  from both RBPF and greedy successor enumeration.

The pseudocode for the resulting algorithm is shown in Fig. 18. Part 2(b) of the algorithm implements an A\* search for

the  $k$  successors with the highest posterior. As in Section 2.3, partial paths in the search tree correspond to partial assignments of modes to components. A goal candidate is one that has a full assignment of modes, and for which the posterior probability  $b(\mathbf{x}_{d,0:t})$  has been calculated. The two key additions to the method in Section 2.3 are as follows:

- (1) *Addition of RBPF particles to search queue:* In 2(a), the RBPF update step is carried out as described in Section 3.7, and the resulting particles are added to the search queue as candidates. To ensure soundness of the A\* search, candidates are added with the unnormalized observation function used by [41].
- (2) *Checking for uniqueness:* Many identical candidates are generated by RBPF and added to the search queue. The set  $\text{new\_kbest}$ , however, holds *unique* trajectories. Part 2(b) ensures that unique trajectories only are added to the set of  $k$  best.

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

We have therefore proposed a new algorithm that mixes greedy and stochastic search in hybrid estimation. In Section 6 we show that this approach is more robust than  $k$ -best and RBPF taken individually.

## 6. Simulation results

In this section we present three main contributions. First, using individual estimation runs, we demonstrate empirically the insight in Section 5.1, that both  $k$ -best enumeration and Rao-Blackwellised Particle Filtering can perform poorly depending on whether the posterior distribution is concentrated in relatively few mode trajectories, or is flat across many. Second, we perform a detailed empirical analysis of the relative performance of  $k$ -best enumeration and RBPF that confirms this result, and illustrates the need for our robust algorithm, which balances the complementary approaches of greedy exploitation and stochastic exploration. Finally, we demonstrate that the new mixed algorithm is significantly more robust than either approach alone. We do not compare the Rao-Blackwellised approach with a standard particle filtering approach, since this analysis has been carried out extensively by [13,21,42]; this analysis revealed that Rao-Blackwellisation increases the efficiency of the particle filtering approach dramatically.

We consider the acrobatic robot introduced in Section 2 and shown in Figs. 1 through 5. Recall that the goal of hybrid estimation in this case is to filter out the acrobot's hybrid state from a sequence of noisy observations of  $\theta_2$ . We consider the following three scenarios for hybrid estimation with the acrobot model:

- (1) In the *nominal* scenario, the robot remains in the nominal mode ( $\text{ball} = \text{no}$ ,  $\text{actuator} = \text{ok}$ ) for the duration of the experiment.

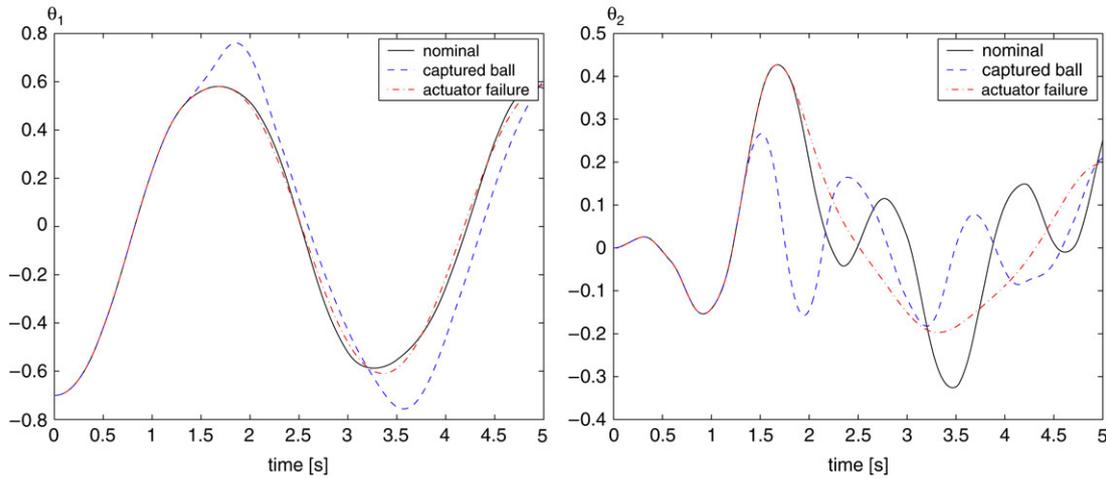


Fig. 19. The evolution of  $\theta_1$  (left) and  $\theta_2$  (right) for the acrobot model scenarios.

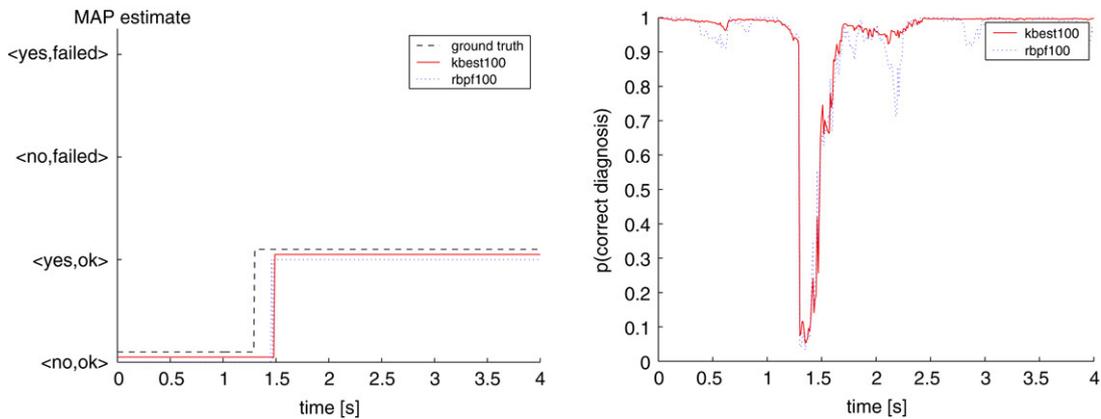


Fig. 20. A single run for the ball capture scenario with concentrated posterior. Left: Maximum a posteriori (MAP) mode estimate computed by the Rao–Blackwellised Particle Filter (rbpf) and the  $k$ -best filter (kbest). Right: probability of the correct diagnosis ( $\text{ball} = \text{yes}, \text{actuator} = \text{ok}$ ) for  $t \geq 1.3$  s. There is a delay between the ball transition and when it is identified as the most likely diagnosis because the posterior probability of the true mode trajectory builds up only over time. The low probability of a transition to ( $\text{ball} = \text{yes}, \text{actuator} = \text{no}$ ) biases the prior towards the ( $\text{ball} = \text{no}, \text{actuator} = \text{ok}$ ) diagnosis.

- (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 Fig. 19.
- (3) In the *failure* scenario, the robot's actuator breaks at  $t = 1.5$ . This event causes the actuator to stop exerting any torque, and alters the robot's trajectory, as shown in Fig. 19.

### 6.1. Individual estimation runs

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

#### 6.1.1. Concentrated posterior

First, we examine the original acrobatic robot model, shown in Figs. 2 and 5. In this model, the probability of a ball transition is 0.01 and the probability of an actuator failure is 0.0005. Since all transitions have low priors, the true posterior distribution

is concentrated in a relatively small number of discrete mode trajectories. The mass of the ball is 4 kg, and each link has mass 1 kg. The upper link has length 2 m and the lower link has length 1 m. Sensor noise is modeled as additive white Gaussian noise with standard deviation 0.04 rad. Additive Gaussian white process noise with standard deviation 0.01 is applied to each element of the state vector, with the noise for each state vector element being uncorrelated with all other elements.

Fig. 20 shows the maximum a posteriori (MAP) estimate of the discrete state by  $k$ -best and RBPF estimation filters for the ball scenario, for a single simulation run with 100 tracked sequences. Both  $k$ -best and RBPF algorithms estimate the MAP diagnosis correctly, and track the continuous state closely, except for an uncertain area close to the transition. Fig. 21 shows the tracking of  $\theta_1$  for the ball scenario.

Figs. 22 and 23 show the results for a single simulation run with the failure scenario. In this case the  $k$ -best algorithm is able to diagnose the fault after a delay, but the Rao–Blackwellised Particle Filter does not diagnose the correct mode, even after several seconds. As described in Section 5.1, this is because

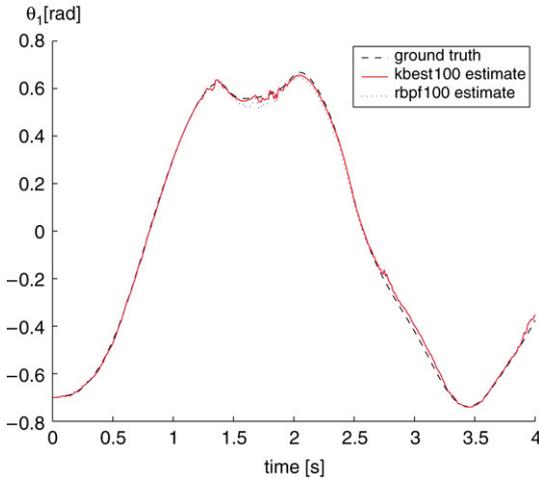


Fig. 21. Filtered  $\theta_1$  for an execution of the ball scenario.

the low prior of the fault transition makes it unlikely that the true mode trajectory is sampled unless many more particles are used. The true mode trajectory is therefore discarded. By contrast,  $k$ -best enumeration is able to retain the true mode sequence, since the posterior distribution is concentrated in relatively few mode sequences.

6.1.2. Flat posterior

We now consider single simulation results for a modified model, in which the posterior distribution is spread out across many distinct mode trajectories. The modified acrobot model has the same components as before, however the model parameters are different. The new transition priors are shown in Fig. 24. Note that the probability of the acrobot catching a ball has changed from 0.05 to 0.5. Also, the mass of the ball is decreased to 1 kg. All other parameters are unchanged. The increased transition probability means that whenever  $\theta_1 \geq 0.55$  the number of mode sequences with a high prior grows exponentially. Because the effect of a transition on observations is initially small, a large number of distinct trajectories will have high posterior probability.

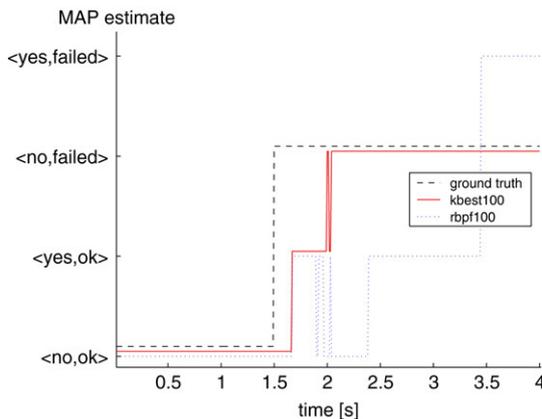


Fig. 22. A single run for the actuator failure scenario with concentrated posterior. Left: Maximum a posteriori (MAP) estimate computed by the RBPf and the  $k$ -best filter. Right: probability of the correct diagnosis (ball = no, actuator = failed) for  $t \geq 1.5$  s. The  $k$ -best algorithm is able to diagnose the fault after a delay; the probability assigned to the ground truth drops close to zero immediately after the fault, but gradually increases as the observations reveal that this is in fact the most likely diagnosis. The RBPf algorithm fails to sample the true mode trajectory due to its low prior, and hence the probability of the ground truth falls to zero and remains there.

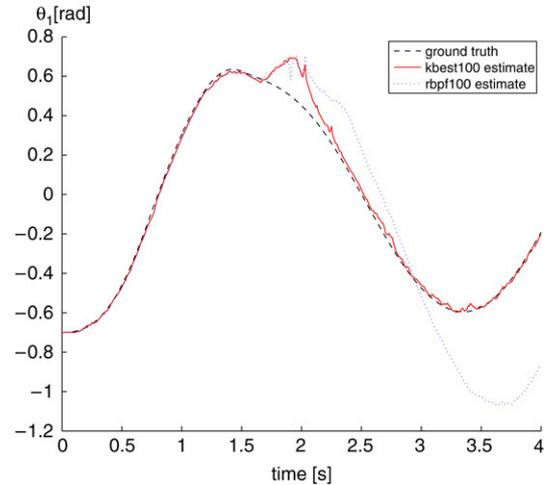


Fig. 23. Filtered  $\theta_1$  for an execution of the failure scenario with concentrated posterior. The  $k$ -best algorithm significantly outperforms the RBPf algorithm.

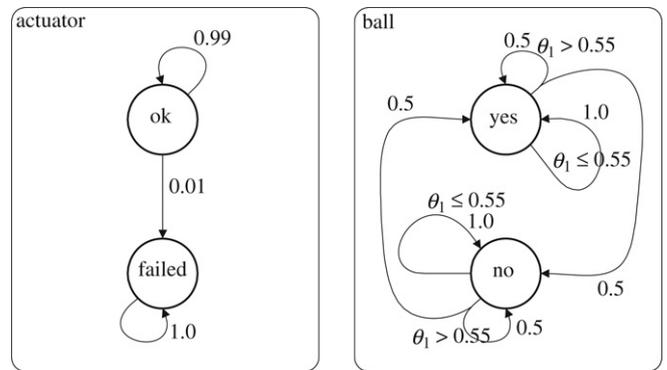
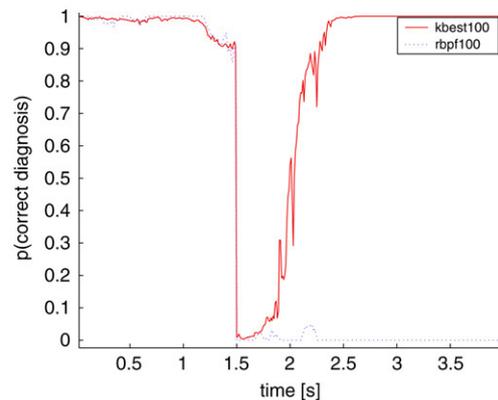


Fig. 24. Probabilistic hybrid automata for the body and actuator of the modified acrobot example.

The modified acrobot model is an example where the fair sampling of the Rao–Blackwellised Particle Filter can outperform the greedy search in the  $k$ -best filter, as discussed in Section 5.1. Figs. 25 and 26 show the results for a single hybrid estimation run on the failure scenario, this time with



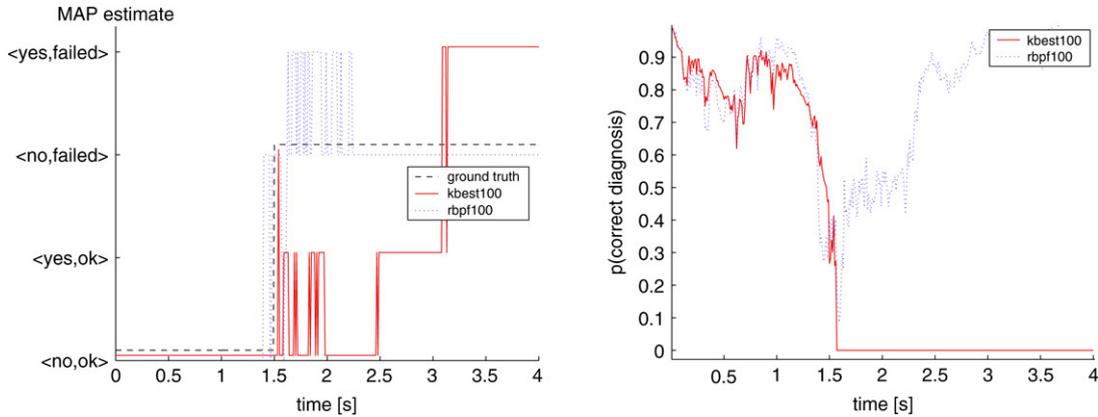


Fig. 25. A single run for the actuator failure scenario with flat posterior. Left: Maximum a posteriori (MAP) estimate computed by the RBPf and the  $k$ -best filter. Right: probability of the correct diagnosis (ball = no, actuator = failed) for  $t \geq 1.5$  s.

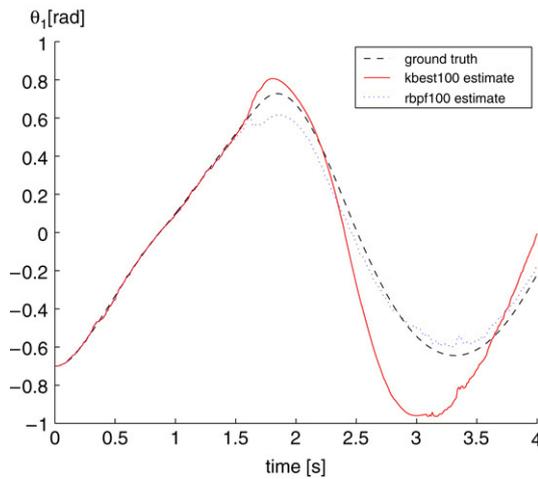


Fig. 26. Continuous tracking of  $\theta_1$  with the failure scenario for a flat posterior. The RBPf has much more accurate continuous tracking than the  $k$ -best algorithm for this example. The state variable  $\theta_1$  is not observed directly, hence without an accurate estimate of the discrete mode sequence, the continuous dynamics of the system are unknown, leading to large estimation errors in the continuous state.

the flat posterior distribution. The  $k$ -best enumeration approach fails to diagnose the fault correctly because very many distinct hypotheses with high posterior probability grow whenever the ball transition is enabled. The stochastic sampling approach of Rao–Blackwellised Particle Filtering, on the other hand, samples the failure transition fairly. When a particle samples the transition close to where it occurred in reality, the observation likelihood of that particle grows until it dominates the trajectory space, giving the correct diagnosis. Note that even at times  $t \in [1.5, 2.0]$  when the RBPf diagnosis is uncertain, RBPf determines that the mode is one of two approximately equally likely values, one of which is the true mode.  $K$ -best, on the other hand, assigns a probability of zero to the true mode.<sup>10</sup>

<sup>10</sup> At such times the correct diagnosis could be considered to be ‘unknown mode’. While the MAP error criterion used in this section does not reflect this, the MAP criterion *does* give a better score to diagnoses that are uncertain, but chatter between a set of modes containing the true value, than to diagnoses that do not contain the true value. Hence the criterion *is* effective at comparing relative performance.

With a flat posterior therefore, RBPf clearly outperforms  $k$ -best enumeration.

These experimental results, therefore, provide empirical validation of the insight described in Section 5.1. While these are examples of single simulation runs only, in Section 6.2 we carry out an extensive performance comparison that confirms this result and provides empirical motivation for a mixed method that balances greedy and stochastic approaches to hybrid estimation.

## 6.2. Performance comparison

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

### 6.2.1. Performance metrics

One of the biggest obstacles to evaluating the performance of hybrid state estimation algorithms is that inference with hybrid models is, in general, NP-hard [10], and it is very difficult to obtain the true posterior distribution  $p(\mathbf{x}_{c,t}, \mathbf{x}_{d,t} | \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ . Sometimes, this distribution can be approximated by a particle filter with a large number of samples; however, the accuracy of such approximations may not be bounded tightly enough.

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

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

Each algorithm was run on 20 random observation sequences with fixed mode assignments. The results given here

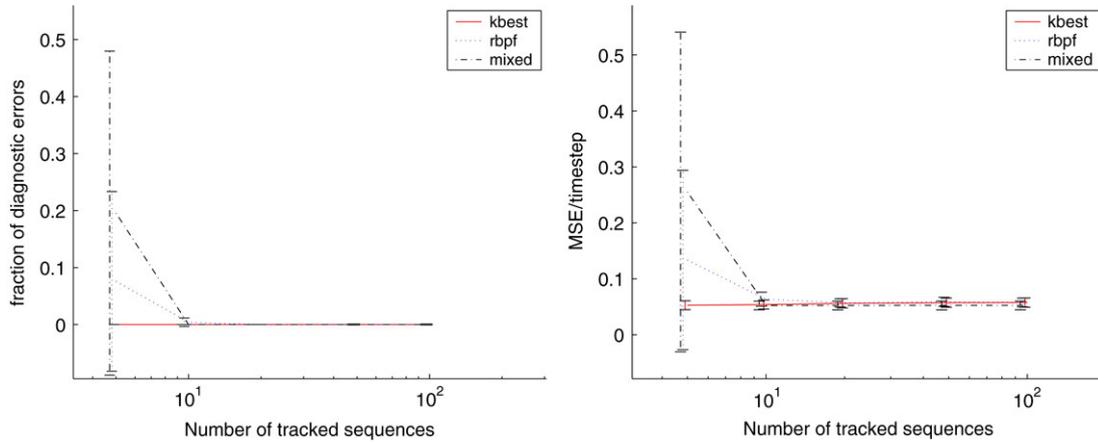


Fig. 27. Performance for the nominal scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state.

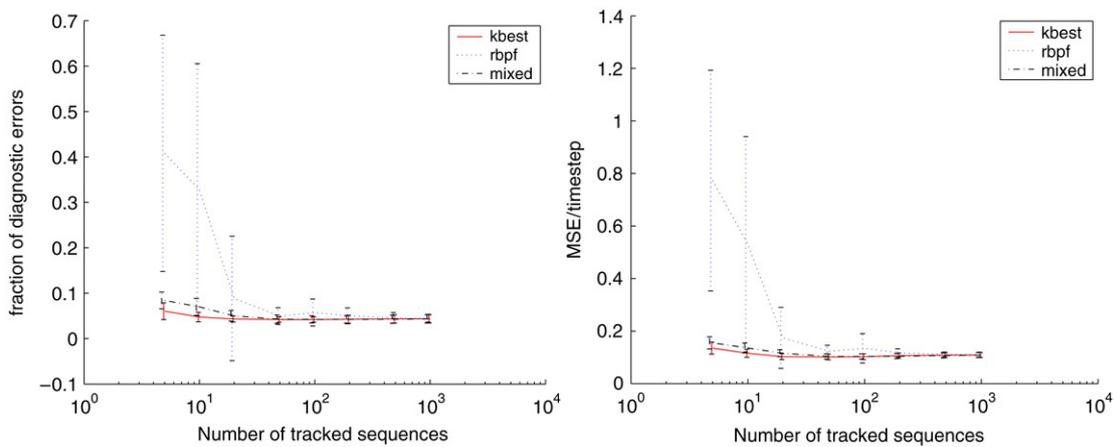


Fig. 28. Performance for the ball scenario. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state. The  $k$ -best algorithm makes almost no diagnostic errors. The Rao–Blackwellised Particle Filter on the other hand, makes more than 40 per cent diagnostic errors on average for small  $k$ ; this decays to a minimum value as the number of tracked sequences increases. In addition, there is far greater variance in the case of the RBPF; in an application such as fault diagnosis, this is particularly undesirable, since reliable performance is essential.

show the mean and standard deviation (shown as error bars) of the performance metrics for these runs. For the mixed method, half of the trajectories were evolved using the RBPF.

### 6.2.2. Concentrated posterior

Figs. 27 through 29 show the percentage of diagnostic errors and the mean square tracking error for the three scenarios considered for the acrobot model with concentrated posterior.

Overall, the Rao–Blackwellised Particle Filter gives a higher number of diagnostic errors than  $k$ -best for the reasons discussed in Section 5.1. The mixed method performs almost as well as  $k$ -best enumeration. These results are the same when we compare the performance with respect to the run time of the algorithms. Fig. 30 shows this comparison. We found that the run times of the three algorithms were different by a factor less than 1.5.

### 6.2.3. Flat posterior

We now compare the performance of the  $k$ -best algorithm and the RBPF when estimating the hybrid state of the modified acrobot model, which has a flat posterior distribution. Fig. 31 shows the performance of the  $k$ -best and RBPF algorithms for

the failure scenario. In this case, the RBPF clearly outperforms the  $k$ -best method both in terms of diagnostic errors and mean square estimation error. The large number of sequences with high posterior prevents the strict enumeration of the  $k$ -best method from considering the initially less likely failure sequence, except with very large  $k$ , while the RBPF samples the actuator failure transition fairly. The mixed method exploits this stochastic approach, performing only slightly worse than the RBPF alone.

### 6.3. Discussion of results

In the experimental results an interesting pattern emerges: the  $k$ -best algorithm undergoes a phase shift in performance, depending on whether or not  $k$  is large enough for trajectories similar to the ground truth to be tracked. By contrast, the performance of the Rao–Blackwellised Particle Filter converges less quickly than the  $k$ -best filter to a low fraction of diagnostic errors (see Fig. 29). Since the RBPF approximates the true posterior and duplicates high likelihood hypotheses, increasing the number of particles simply makes the approximation closer to the true posterior. Hence there is a gradual convergence and not the phase shift seen for  $k$ -best.

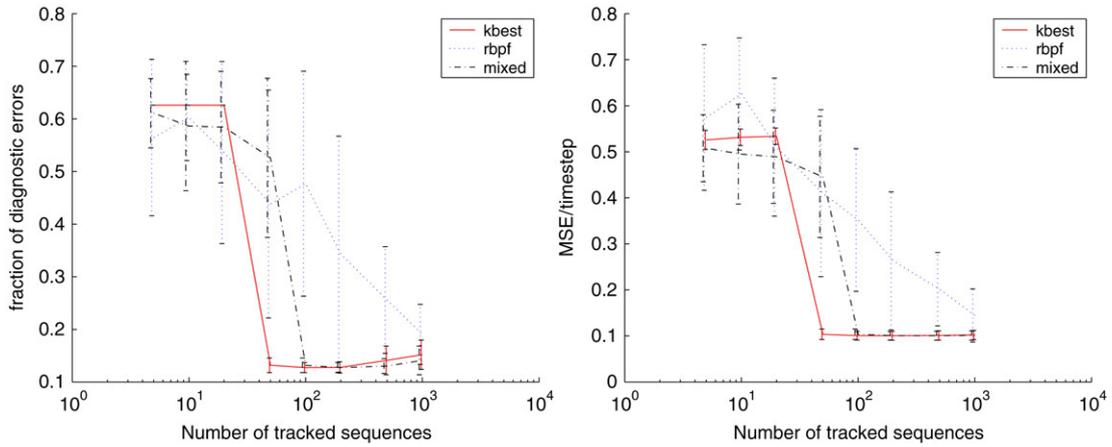


Fig. 29. Performance for the actuator failure scenario with concentrated posterior. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state. For small  $k$ , both RBPf and  $k$ -best have a high proportion of diagnostic errors; for  $k > 20$ , however, the  $k$ -best algorithm outperforms the RBPf. Also, whereas the performance of the RBPf improves gradually as  $k$  increases, the  $k$ -best algorithm shows a large shift in performance between  $k = 20$  and  $k = 50$ . The mixed method behaves in a similar manner to  $k$ -best enumeration, except that the shift in performance occurs at approximately double the number of tracked sequences, since only half of the sequences are being evolved greedily.

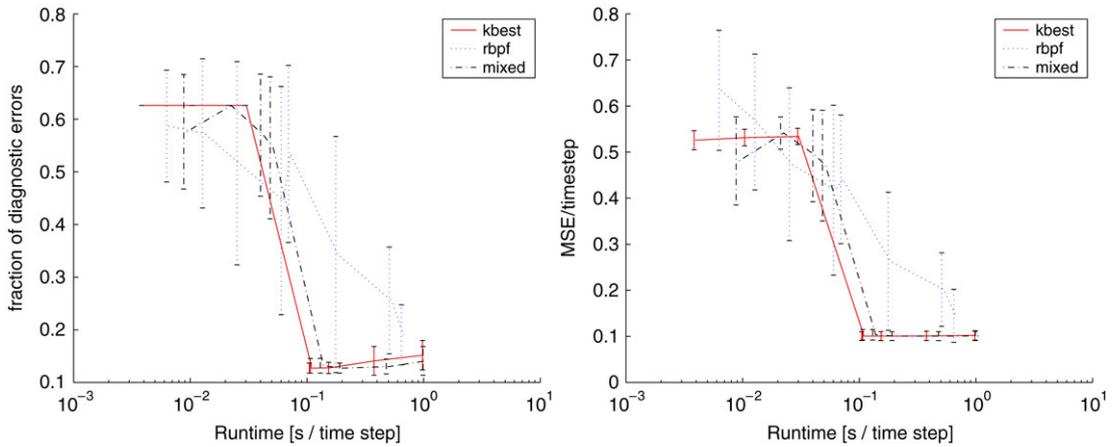


Fig. 30. Performance for the failure scenario for concentrated posterior with respect to run time. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state.

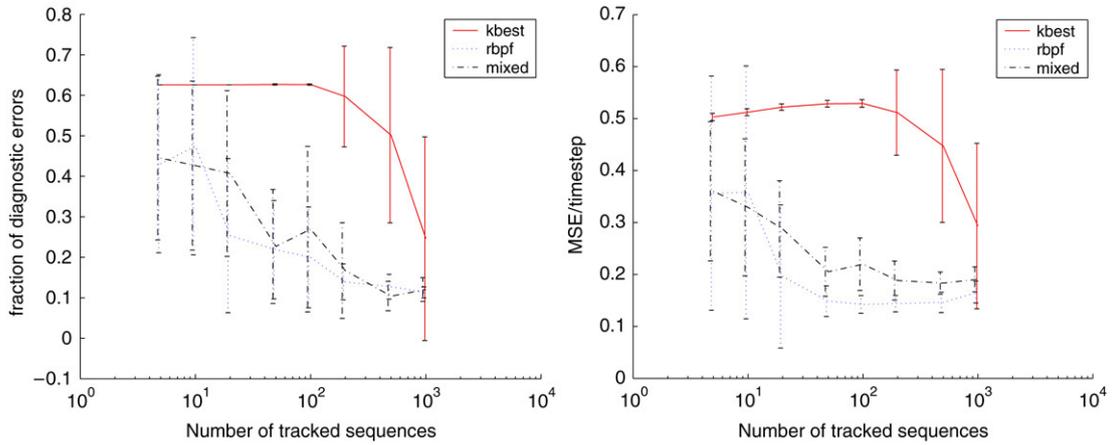


Fig. 31. Performance for the failure scenario with flat posterior. Left: Percentage of diagnostic errors. Right: Mean square estimation error of the continuous state. The  $k$ -best method performs very poorly, except for a very large number of tracked sequences, while the RBPf performs well. The mixed method has similar performance to the RBPf.

The critical value of  $k$  that greedy enumeration needs to track in order to perform well, depends on the concentration of the posterior distribution. If the posterior distribution is concentrated in a relatively small number of distinct sequences

the  $k$ -best method will perform well even for small  $k$ . In these cases,  $k$ -best enumeration typically outperforms RBPF, since it does not duplicate hypotheses, leading to unnecessary approximation. For a flat posterior, on the other hand, the critical value of  $k$  is so large when detecting rare events that many trajectories will need to be tracked, to reliably detect the fault. In these cases, the RBPF will perform better for small  $k$ , since the distribution is sampled fairly.

We have shown, empirically, that the mixed method combines the benefits of the two methods. While its performance is marginally worse than the RBPF or  $k$ -best in their best-case scenario, by balancing greedy and stochastic search, the new method is much more robust to the choice of model parameters than the RBPF and  $k$ -best individually.

## 7. Related work

Several algorithms have addressed the problem of hybrid estimation in linear systems with Markovian mode transitions. One class of solutions are multiple-model estimation schemes, which maintain a pre-determined number of Gaussians. These include the generalized pseudo-Bayesian algorithm (GPB) [43], the Detection Estimation Algorithm (DEA) [44], the Interacting Multiple Model (IMM) algorithm [29], and the residual correlation Kalman filter bank [45]. [46] derived a number of analytic results regarding the convergence of the IMM and Kalman filter bank algorithms and proposed an extension, based on these results, called the Residual-Mean Interacting Multiple Model (RMIMM) algorithm. [30] 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 [11], to the setting of hybrid dynamic Bayesian networks with SoftMax transitions, using numerical integration techniques instead of the Kalman Filter. As is the case with our prior work in [5], their algorithm provides an any-time solution to the hybrid state estimation problem and can handle autonomous mode transitions. Later work by [47,48] also considers autonomous mode transitions. In a similar manner to [5], [47] propose a Monte Carlo approach to evaluating the probability of guard satisfaction, while [48] propose an analytic approach that can handle the special case of transition distributions with Gaussian forms.

In the particle filtering community, several papers [14,49–51] have proposed using the bootstrap particle filter to perform state estimation in hybrid models. An early application of the Rao–Blackwellisation method to reducing the variance of sampling in SLDS models was introduced by [22]. Their algorithm, named the Random Sampling Algorithm (RSA), sampled the sequences of mode assignments using the distribution  $p(\mathbf{x}_{d,t} | \mathbf{x}_{d,0:t-1}, \mathbf{y}_{1:t}, \mathbf{u}_{0:t})$ . [52,53] introduced the Selection Step, which is crucial for the convergence of sequential Monte Carlo methods and framed the problem in the general particle filtering framework. In addition, they proved several properties regarding the convergence and variance reduction of Rao–Blackwellisation schemes. [49]

further extended this work and described an algorithm for fixed-lag smoothing with MCMC steps. Finally, [13] introduced a procedure, called one-step look-ahead, which computes the total probability for the sequences stemming from a given sample and moves the selection step before the importance sampling step, at the cost of evaluating the Kalman Filter residual for all successor modes. All of these techniques were designed for linear switching models without autonomous transitions.

Concurrently with our work published in [18], the combination of a Rao–Blackwellised particle filter with an Unscented Kalman Filter for fault detection was proposed independently by [21]. In addition, they incorporated the look-ahead step proposed by [13].

Two complementary approaches for improving the performance of particle filters were proposed by [33,54]. 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.

Related work in the Artificial Intelligence community includes [55,56]. [55] use Qualitative Reasoning techniques to address the problem of hybrid estimation for systems that do not have stochastic noise, and whose transitions are triggered by exogenous events. [56] extended this work to consider systems with stochastic noise, but again did not consider stochastic jumps in the discrete state.

## 8. Conclusion

In this paper, we investigated the problem of estimating the state of a system represented with probabilistic hybrid models. We presented an efficient Rao–Blackwellised particle filtering algorithm, developed in Section 3, that handles the autonomous mode transitions, concurrency, and nonlinearities present in Concurrent Probabilistic Hybrid Automata (CPHA). We extended the class of autonomous mode transitions that can be handled by both  $k$ -best enumeration and Rao–Blackwellised Particle Filtering for CPHA to multivariable linear transition guards, in the case of piecewise constant transition distributions, and to polynomial transition distributions of arbitrary order, in the case of single variables.

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

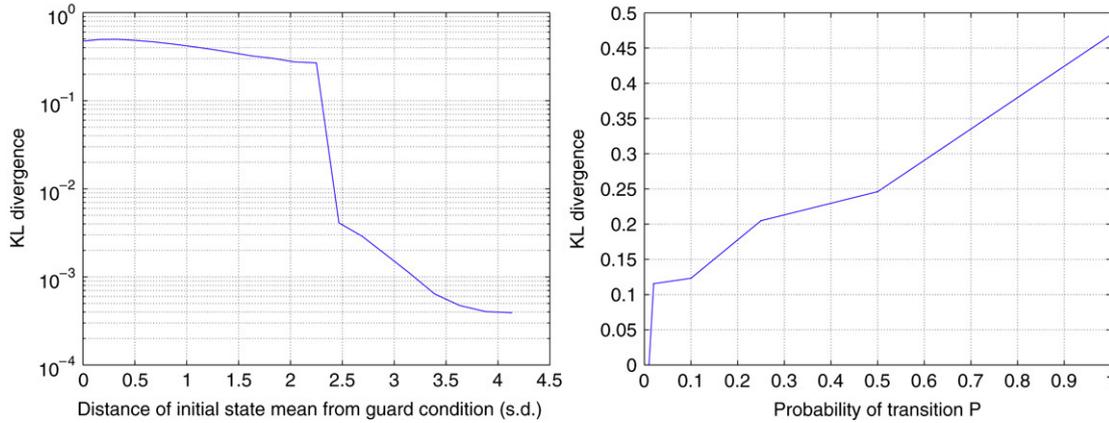


Fig. A.1. Kullback–Liebler divergence between Kalman Filter based approximation and true distribution with autonomous mode transition.

approach of the Rao–Blackwellised particle filter is more successful.

We therefore developed a new algorithm that combines greedy and stochastic search by tracking two sets of mode trajectories with  $k$ -best and Rao–Blackwellised particle filters, and uses both sets to generate the set of leading trajectories at the next time step. Simulations showed that the algorithm is more robust than  $k$ -best and RBPF taken individually, with only a minor performance penalty.

## Appendix A

### A.1. Kalman Filter approximation error

In this section we investigate the error introduced by autonomous mode transitions. Consider a simple system with one continuous state variable and continuous dynamics given by:

$$\mathbf{x}_{c,t+1} = a\mathbf{x}_{c,t} + \mathbf{v}_{x,t}. \quad (\text{A.1})$$

There are two modes, and at time  $t$  we are in Mode 1 with probability 1. For Mode 1,  $a = 1$ , while for Mode 2,  $a = -1$ . The distribution of  $\mathbf{x}_{c,t}$  is Gaussian with mean  $\hat{\mathbf{x}}_{c,t}$  and variance 2. The variance of  $\mathbf{v}_x$  is 0.5. There are two guard conditions; for  $\mathbf{v}_x > 10$ , we remain in Mode 1 with probability 1; for  $\mathbf{v}_x \leq 10$ , we transition to Mode 2 with probability  $P$ . When  $P$  is close to zero, the discrete dynamics are unaffected by satisfaction of the guard  $\mathbf{v}_x \leq 10$ . When  $P$  is close to one, guard satisfaction has a large effect on the discrete dynamics. We have chosen this particular system for two reasons. First, the autonomous mode transition introduces a severe nonlinearity; if the state is just above 10, it remains there, while just below 10 it becomes  $-10$  at the next time step. Note that the acrobot model is much more benign in this sense, however we wish to examine a ‘worst-case’ scenario for autonomous transitions. Second, since the system is linear, we can analyze the error introduced by autonomous mode transitions in isolation.

We now compare the true distribution  $p(\mathbf{x}_{c,t+1}|\mathbf{y}_{c,0:t})$  to the approximation generated using the Kalman Filter approach employed in the RBPF algorithm. We determine a close approximation to the true distribution using  $10^7$

non-Rao–Blackwellised particles, and calculate the sum-of-Gaussians RBPF approximation using 100 particles. Fig. A.1 shows the Kullback–Liebler (KL) divergence (calculated approximately using numerical integration) between the two distributions as a function of  $\hat{\mathbf{x}}_{c,t}$  and  $P$ . This plots show that in the worst case, the KL divergence is significant, but reasonable,<sup>11</sup> and that as *either* the state mean moves away from the guard condition, *or* the probability  $P$  decreases, the KL divergence decreases to insignificant levels. In these cases, while there is still significant error in the distribution of  $\mathbf{x}_{c,t+1}$  conditioned on a mode sequence having occurred, the contribution of this conditional distribution to the marginal distribution of  $\mathbf{x}_{c,t}$ , and hence the KL divergence, is small. Note also that the error in the distribution analyzed here will reduce as observations are incorporated. These results show that the error introduced by autonomous mode transitions when using Kalman Filter approximations is only large if the following conditions hold. First, guard satisfaction must have a significant effect on the discrete dynamics. For example, discrete mode changes could be very likely if a guard is satisfied, and very unlikely otherwise. Second, there must significant probability mass either side of the guard threshold. In other words, the expected continuous state must be close to the guard threshold.

We conclude, that, while in the worst case autonomous mode transitions lead to large approximation errors, in most cases the Kalman Filter approximation used in the new RBPF and mixed algorithms is an effective approach that brings large computational savings compared to standard Particle Filter approaches. Furthermore, in Section 6 we show that the new algorithms based on Kalman Filter approximations perform well in terms of mean squared estimation error and MAP mode estimation error for a number of scenarios.

### A.2. Acrobot dynamic equations

Here we present the discretized dynamic equations of the acrobot model in full. The state vector consists of the angular

<sup>11</sup> For comparison, two Gaussians with identical variance but separated by one standard deviation have a KL divergence of 0.5.

positions and velocities of the two joints:

$$\mathbf{x} = [\theta_1 \ \theta_2 \ \omega_1 \ \omega_2]^\top. \quad (\text{A.2})$$

The masses of the upper and lower links are given by  $m_1$  and  $m_2$ , respectively. The lengths are given by  $l_1$  and  $l_2$ , respectively. The difference equations defining the system dynamics are as follows:

$$\theta_{1,t+1} = \theta_{1,t} + \omega_{1,t}\delta t + v_{\theta 1} \quad (\text{A.3})$$

$$\theta_{2,t+1} = \theta_{2,t} + \omega_{2,t}\delta t + v_{\theta 2} \quad (\text{A.4})$$

$$\omega_{1,t+1} = \omega_{1,t} + B_1\delta t + v_{\omega 1} \quad (\text{A.5})$$

$$\omega_{2,t+1} = \omega_{2,t} + B_2\delta t + v_{\omega 2}, \quad (\text{A.6})$$

where:

$$B_1 = (T_2 D_{12} - T_1 D_{22} - \omega_{1,t}^2 D_{122} D_{12} + \omega_{2,t}^2 D_{122} D_{22} + 2\omega_{1,t}\omega_{2,t} D_{122}(D_{22} - D_{12}) + D_{22} D_1 - D_{11} D_2) / D_3 \quad (\text{A.7})$$

$$B_2 = -(T_2 D_{11} - T_1 D_{12} - \omega_{1,t}^2 D_{122} D_{11} + \omega_{2,t}^2 D_{122} D_{12} + 2\omega_{1,t}\omega_{2,t} D_{122}(D_{12} - D_{11}) + D_{12} D_1 - D_{11} D_2) / D_3,$$

and  $v_{\theta 1}$ ,  $v_{\theta 2}$ ,  $v_{\omega 1}$ ,  $v_{\omega 2}$  are noise processes. The  $D$  terms in the above equations are given by:

$$D_1 = (m_1 + m_2)gl_1 \sin \theta_{1,t} + m_2 gl_2 \sin(\theta_{1,t} + \theta_{2,t})$$

$$D_2 = m_2 gl_2 \sin(\theta_{1,t} + \theta_{2,t})$$

$$D_3 = D_{12}^2 - D_{11} D_{22}$$

$$D_{11} = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos \theta_{2,t} \quad (\text{A.8})$$

$$D_{22} = m_2 l_2^2$$

$$D_{12} = m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_{2,t}$$

$$D_{122} = -m_2 l_1 l_2 \sin \theta_{2,t}.$$

$T_1$  and  $T_2$  are the torques at each of the joints, which are given by:

$$T_1 = -\omega_{1,t} \quad (\text{A.9})$$

$$T_2 = -\omega_{2,t} + T,$$

where  $T$  is the torque applied by the actuator; this is zero if the actuator has failed.

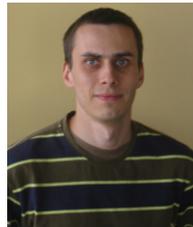
## References

- [1] Mars science laboratory, <http://mars.jpl.nasa.gov/msl/>, 2006.
- [2] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, V. Verma, Experiences with a mobile robotic guide for the elderly, in: Proceedings of the AAAI National Conference on Artificial Intelligence, AAAI, Edmonton, Canada, 2002.
- [3] B.C. Williams, S. Chung, V. Gupta, Mode estimation of model-based programs: Monitoring systems with complex behavior, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2001.
- [4] O.B. Martin, B.C. Williams, M.D. Ingham, Diagnosis as approximate belief state enumeration for probabilistic concurrent constraint automata, in: 20th National Conference on Artificial Intelligence, 2005.
- [5] M. Hofbaur, B.C. Williams, Mode estimation of probabilistic hybrid systems, in: Intl. Conf. on Hybrid Systems: Computation and Control, 2002.
- [6] C.P. Gomes, B. Selman, H. Kautz, Boosting combinatorial search through randomization, in: Proceedings of the 15th National Conference on Artificial Intelligence, 1998.
- [7] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.
- [8] M.W. Hofbaur, B.C. Williams, Hybrid estimation of complex systems, IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics (2004) 2178–2191.
- [9] V. Verma, J. Langford, R. Simmons, Non-parametric fault identification for space rovers, in: International Symposium on Artificial Intelligence and Robotics in Space, iSAIRAS, 2001.
- [10] U. Lerner, R. Parr, Inference in hybrid networks: Theoretical limits and practical algorithms, in: Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence, UAI-01, Seattle, WA, 2001, pp. 310–318.
- [11] U. Lerner, Hybrid bayesian networks for reasoning about complex systems, Ph.D. Thesis, Stanford University, October 2002.
- [12] A. Doucet, On sequential simulation-based methods for bayesian filtering, Tech. Rep. CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering, 1998.
- [13] R. Morales-Menéndez, N. de Freitas, D. Poole, Real-time monitoring of complex industrial processes with particle filters, in: Proceedings of Neural Information Processing Systems, NIPS, Vancouver, 2002.
- [14] R. Dearden, D. Clancy, Particle filters for real-time fault detection in planetary rovers, in: Proceedings of the 13th International Workshop on Principles of Diagnosis, DX02, Semmering, Austria, 2002, pp. 1–6.
- [15] A. Doucet, N. de Freitas, K. Murphy, S. Russell, Rao–Blackwellised particle filtering for dynamic bayesian networks, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI, 2000.
- [16] N. de Freitas, Rao–Blackwellised particle filtering for fault diagnosis, in: Proceedings of the IEEE Aerospace Conference, vol. 4, 2002, pp. 1767–1772.
- [17] X. Koutsoukos, J. Kurien, F. Zhao, Monitoring and diagnosis of hybrid systems using particle filtering methods, in: Fifteenth International Symposium on Mathematical Theory of Networks and Systems, Notre Dame, IN, 2002.
- [18] S. Funiak, B.C. Williams, Multi-modal particle filtering for hybrid systems with autonomous mode transitions, in: Proceedings of SafeProcess (also published in DX-2003), 2003.
- [19] B. Anderson, J. Moore, Optimal Filtering, Information and System Sciences Series, Prentice Hall, 1979.
- [20] S.J. Julier, J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in: Proceedings of AeroSense: The 11th Symposium on Aerospace/Defense Sensing, Simulation and Controls, 1997.
- [21] F. Hutter, R. Dearden, The gaussian particle filter for diagnosis of nonlinear systems, in: Proceedings of the 14th International Conference on Principles of Diagnosis, DX'03, Washington, DC, USA, 2003, pp. 65–70.
- [22] H. Akashi, H. Kumamoto, Random sampling approach to state estimation in switching environments, Automatica 13 (1977) 429–434.
- [23] V. Pavlovic, J. Rehg, T.-J. Cham, K. Murphy, A dynamic Bayesian network approach to figure tracking using learned dynamic models, in: Proceedings of the International Conference on Computer Vision, 1999.
- [24] S. Narasimhan, G. Biswas, G. Karasai, F. Zhao, Building observers to address fault isolation and control problems in hybrid dynamic systems, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2000, 2000, pp. 2393–2398.
- [25] B. Buchberger, F. Winkler (Eds.), Gröbner Bases and Applications, Cambridge Univ. Press, 1998.
- [26] P. Nayak, Automated Modelling of Physical Systems, in: Lecture Notes in Artificial Intelligence, Springer, 1995.
- [27] M. Hofbaur, Hybrid estimation and its role in automation, Habilitationsschrift, Faculty of Electrical Engineering, Graz University of Technology, Austria, September 2003.
- [28] T. Dean, K. Kanazawa, A model for reasoning about persistence and causation, Computational Intelligence 5 (3) (1989) 142–150.
- [29] H. Blom, Y. Bar-Shalom, The interacting multiple model algorithm for systems with markovian switching coefficients, IEEE Transactions on Automatic Control 33 (1988) 780–783.

- [30] U. Lerner, R. Parr, D. Koller, G. Biswas, Bayesian fault detection and diagnosis in dynamic systems, in: Proc. of the 17th National Conference on A. I., Austin, TX, 2000, pp. 531–537. URL: <http://www.citeseer.nj.nec.com/lerner00bayesian.html>.
- [31] B.C. Williams, P. Nayak, A model-based approach to reactive self-configuring systems, in: Proceedings of AAAI-96, 1996, pp. 971–978.
- [32] B. Ng, L. Peshkin, A. Pfeffer, Factored particles for scalable monitoring, in: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, 2002.
- [33] V. Verma, S. Thrun, R. Simmons, Variable resolution particle filter, in: Proceedings of International Joint Conference on Artificial Intelligence, 2003.
- [34] G. Casella, C.P. Robert, Rao-Blackwellisation of sampling schemes, *Biometrika* 83 (1) (1996) 81–94.
- [35] K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian networks, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001, pp. 499–515 (Chapter 24).
- [36] N. de Freitas, R. Dearden, F. Hutter, R. Morales-Menendez, J. Mutch, D. Poole, Diagnosis by a waiter and a mars explorer, in: *Sequential State Estimation*, Proceedings of the IEEE 92 (3) (2004) 455–468 (special issue).
- [37] P. Nayak, B.C. Williams, Fast context switching in real-time propositional reasoning, in: Proceedings of AAAI-97, 1997, pp. 50–56.
- [38] H. Joe, Approximations to multivariate normal rectangle probabilities based on conditional expectations, *Journal of the American Statistical Association* 90 (431) (1995) 957–964.
- [39] A. Genz, K.-S. Kwong, Numerical evaluation of singular multivariate normal distributions, *Journal of Statistical Computation and Simulation* 68 (2000) 1–21.
- [40] A. Genz, Numerical computation of rectangular bivariate and trivariate normal and t probabilities, *Statistics and Computing* 14 (2004) 151–160.
- [41] P. Maybeck, R. Stevens, Reconfigurable flight control via multiple model adaptive control methods, *IEEE Transactions on Aerospace and Electronic Systems* 27 (3) (1991) 470–480.
- [42] A. Doucet, N.J. Gordon, V. Krishnamurthy, Particle filters for state estimation of jump Markov linear systems, *IEEE Transactions on Signal Processing* 49 (3) (2001) 613–624.
- [43] G. Ackerson, K. Fu, On state estimation in switching environments, *IEEE Transactions on Automatic Control* 15 (1970) 10–17.
- [44] J. Tugnait, Detection and estimation for abruptly changing systems, *Automatica* 18 (1982) 607–615.
- [45] P. Hanlon, P. Maybeck, Multiple-model adaptive estimation using a residual correlation kalman filter bank, *IEEE Transactions on Aerospace and Electronic Systems* 36 (2) (2000) 393–406.
- [46] I. Hwang, H. Balakrishnan, C. Tomlin, State estimation for hybrid systems: Applications to aircraft tracking, *IEEE Proceedings on Control Theory and Applications* 153 (5) (2006) 556–566.
- [47] C.E. Seah, I. Hwang, Hybrid estimation algorithm using state-dependent mode transition matrix for aircraft tracking, in: Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, CO, 2006.
- [48] I. Hwang, C.E. Seah, An estimation algorithm for stochastic linear hybrid systems with continuous-state-dependent mode transitions, in: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 2006.
- [49] A. Doucet, N. de Freitas, N.J. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.
- [50] D. Avitzour, A stochastic simulation Bayesian approach to multitarget tracking, *IEEE Proceedings on Radar Sonar and Navigation* 142 (2) (1995) 41–44.
- [51] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *Journal of Computational and Graphical Statistics* 5 (1996) 1–25.
- [52] N. Metropolis, S. Ulam, The Monte Carlo method, *American Statistical Association* 44 (1949) 335–341.
- [53] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings - F* 140 (2) (1993) 107–113.
- [54] S. Thrun, J. Langford, V. Verma, Risk sensitive particle filters, in: *Neural Information Processing Systems*, NIPS, 2001.
- [55] S. McIlraith, G. Biswas, D. Clancy, V. Gupta, Hybrid systems diagnosis, in: *Proceedings of Hybrid Systems: Computation and Control*, in: *Lecture Notes in Computer Science*, vol. 1790, Springer-Verlag, 2000, pp. 282–295.
- [56] S. McIlraith, Diagnosing hybrid systems: A Bayesian model selection approach, in: Proceedings of the 11th International Workshop on Principles of Diagnosis, DX00, 2000, pp. 140–146.



**Lars Blackmore** has B.A. and M.Eng. degrees from the University of Cambridge (UK) and is currently a Ph.D. candidate at MIT in the department of Aeronautics and Astronautics. He has published nine papers in refereed conference proceedings. His research interests include estimation and control of robotic and autonomous systems under stochastic uncertainty.



**Stanislav Funiak** holds B.S. and M.Eng. degrees from MIT and is currently a Ph.D. student in Robotics at Carnegie Mellon University. He pursues research in probabilistic reasoning, machine learning, and distributed systems. He is interested in algorithms and modeling formalisms that enable robust solutions to real-world problems in sensor networks.



**Brian C. Williams** is a professor at MIT in the department of Aeronautics and Astronautics. He received his Ph.D. from MIT in 1989. He pioneered multiple fault, model-based diagnosis at the Xerox Palo Alto Research Center, and at NASA Ames Research Center formed the Autonomous Systems Area. He co-invented the Remote Agent model-based autonomous control system, which flew on the NASA Deep Space One probe in 1999. He has served as guest editor of the *Artificial Intelligence Journal* and has been on the editorial board of the *Journal of Artificial Intelligence Research*. He is currently a member of the Advisory Council of the NASA Jet Propulsion Laboratory at Caltech. Prof. Williams' research concentrates on model-based autonomy — the creation of long-lived autonomous systems that are able to explore, command, diagnose and repair themselves using fast, commonsense reasoning.