

# Mode Estimation of Probabilistic Hybrid Systems

Michael W. Hofbaur<sup>1,2\*</sup> and Brian C. Williams<sup>1</sup>

<sup>1</sup> MIT Space Systems and Artificial Intelligence Laboratories  
77 Massachusetts Ave., Rm-37-381, Cambridge MA 02139, USA,  
williams@mit.edu

<sup>2</sup> Graz University of Technology, Department of Automatic Control,  
Inffeldgasse 16c/2, A-8010 Graz, Austria,  
hofbaur@irt.tu-graz.ac.at

**Abstract.** Model-based diagnosis and mode estimation capabilities excel at diagnosing systems whose symptoms are clearly distinguished from normal behavior. A strength of mode estimation, in particular, is its ability to track a system's discrete dynamics as it moves between different behavioral modes. However, often failures bury their symptoms amongst the signal noise, until their effects become catastrophic.

We introduce a hybrid mode estimation system that extracts mode estimates from subtle symptoms. First, we introduce a modeling formalism, called *concurrent probabilistic hybrid automata* (cPHA), that merge hidden Markov models (HMM) with continuous dynamical system models. Second, we introduce *hybrid estimation* as a method for tracking and diagnosing cPHA, by unifying traditional continuous state observers with HMM belief update. Finally, we introduce a novel, any-time, any-space algorithm for computing approximate hybrid estimates.

## 1 Introduction

The year 2000 was kicked off with two missions to Mars, following on the heels of the highly successful Mars Pathfinder mission. Mars Climate Orbiter burned up in the Martian atmosphere. After extensive investigation it was found that a units error in a small forces table introduced a small, but indiscernible fault that, over a lengthy time period, caused the loss of the orbiter. The problem of misinterpreting a system's dynamics was punctuated later in the year when the Mars Polar Lander vanished without a trace. After months of analysis, the failure investigation team concluded that the vehicle most likely crashed into Mars, because it incorrectly shutdown its engine at 40 meters above the surface. This failure, like the orbiter, resulted from a misinterpretation of the vehicle's dynamics, in this case, due to a faulty software monitor.

The above case study is a dramatic instance of a common problem – increasingly complex systems are being developed, whose failure symptoms are nearly

---

\* Supported by NASA under contract NAG2-1388.

indiscernible, up until a catastrophic result occurs. To tackle this problem we address two issues. First, these failures are manifest through a coupling between a system's continuous dynamics, and its evolution through different behavior modes. We address this issue by developing hybrid monitoring and diagnosis capabilities that are able to track a system's behavior, along both its continuous state evolution and its discrete mode changes. Second, failures may generate symptoms that are initially on the same scale as sensor and actuator noise. To discover these symptoms, we use statistical methods to separate the noise from the true dynamics.

We address this challenge by extending the deductive mode estimation capabilities of the Livingstone system[1], to reason about continuous dynamics, using classical methods for state estimation[2]. After a motivating example, we discuss traditional methods for separately estimating discrete and continuous behaviors. We then introduce a modeling formalism, called *concurrent probabilistic hybrid automata* (cPHA), that merges hidden Markov models (HMM) with continuous dynamical system models. A cPHA provides a modeling framework that captures probabilistic mode transitions. This is unlike most traditional hybrid modeling frameworks, for example, [3,4,5,6], which define mode transitions to be deterministic, or do not explicitly specify probabilities for transitions.

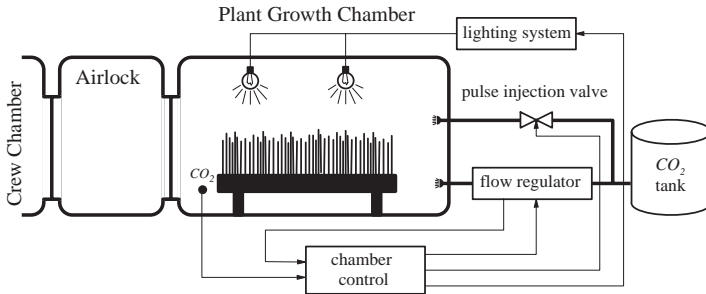
Next, we introduce a method, called *hybrid mode estimation*, that tracks and diagnoses cPHA, by creating a hybrid HMM observer. The observer uses the results of continuous state estimates to estimate a system's mode changes, and coordinates the actions of a set of continuous state observers. This approach is similar to work pursued in multi-model estimation[7,8,9]. However, we provide a novel any-time, any-space algorithm for computing approximate hybrid estimates, which allows us to track concurrent automata that have a large number of possible modes.

Several approaches have been recently introduced for hybrid system diagnosis, including approaches that bridge methods from model-based diagnosis and multi-model filtering (e.g. [10,11,12]), and less traditional methods, such as dynamic Bayesian networks[13] or particle filters[14]. Most of these methods, however, do not deal with autonomous mode transitions of the system under investigation. Our paper remedies this situation and provides one possible path towards a unified framework for monitoring and diagnosing hybrid systems.

## 2 Example: BIO-Plex

Our application is the BIO-Plex Test Complex at NASA Johnson Space Center, a five chamber facility for evaluating biological and physiochemical life support technologies. It is an artificial, biosphere-type, closed environment, which must robustly provide all the air, water, and most of the food for a crew of four without interruption. Plants are grown in plant growth chambers, where they provide food for the crew, and convert the exhaled  $CO_2$  into  $O_2$ . In order to maintain a closed-loop system, it is necessary to control the resource exchange between the chambers, without endangering the crew. For the scope of this paper, we restrict

our evaluation to the sub-system dealing with  $CO_2$  control in the plant growth chamber (PGC), shown in Fig. 1.



**Fig. 1.** BIO-Plex plant growth chamber

The system is composed of several components, such as a flow regulator that provides continuous  $CO_2$  supply, a pulse injection valve that provides a means for increasing the  $CO_2$  concentration rapidly, a lighting system and the plant growth chamber, itself. The control system maintains a plant growth optimal  $CO_2$  concentration of 1200 ppm during the day phase of the system (20 hours/day). This  $CO_2$  level is unsuitable for humans, hence the gas concentration is lowered to 500 ppm, whenever crew members request to enter the chamber for harvesting, re-planting or other service activities. Safety regulations require that the system inhibit high volume gas injection via the pulse-injection path, while crew members are in the PGC. Sensors are available to record entry and exit of crew members. However, sensors are known to fail. In this paper we demonstrate how hybrid estimation provides a robust backup strategy that detects the presence of crew members, based on the slight change in the gas balance that is caused by the exhaled  $CO_2$ .

Hybrid estimation schemes are key to tracking system operational modes, as well as, detecting subtle failures and performing diagnoses. For instance, a partial lighting failure has impact on the gas conversion rate due to lower photosynthesis activity. The failure leads to behavior that is similar to crew members entering the PGC. A hybrid estimation scheme should correctly discriminate among different operational and failure modes of each system component, based on the overall system operation.

### 3 Traditional Estimation

To model a hybrid system, we start by using a *hidden Markov model (HMM)* to describe discrete stochastic changes in the system. We then fold in the continuous dynamics, by associating a set of continuous dynamical equations with each HMM state. To avoid confusion in terminology, we refer to the HMM state as the

system's mode, and reserve the term state to refer to the state of a probabilistic hybrid automaton. We develop a hybrid estimation capability by generalizing traditional methods for estimating HMM states and continuous state-variables.

### 3.1 Estimating HMMs

For an HMM, estimation is framed as a problem of belief-state update, that is, the problem of determining the probability distribution  $b_{(k)}$  over modes  $\mathcal{M}$  at time-step  $k$ . The probability of being in a mode  $m_i$  at time-step  $k$  is denoted  $b_{(k)}[m_i]$ .

**Definition 1.** A *Hidden Markov Model (HMM)* can be described by a tuple  $\langle \mathcal{M}, \mathcal{Y}_d, \mathcal{U}_d, P_\Theta, P_T, P_O \rangle$ .  $\mathcal{M}$ ,  $\mathcal{Y}_d$  and  $\mathcal{U}_d$  denote finite sets of *feasible modes*  $m_i$ , *observations*  $y_{di}$  and *control values*  $u_{di}$ , respectively. The *initial state function*,  $P_\Theta[m_i]$ , denotes the probability that  $m_i$  is the initial mode. The *mode transition function*,  $P_T(m_i|u_d, m_j)$ , describes the probability of transitioning from mode  $m_{j,(k-1)}$  to  $m_{i,(k)}$  at time-step  $k$ , given a discrete control action  $u_{d,(k-1)}$ . The *observation function*  $P_O(y_d|m_i)$  describes the probability that a discrete value  $y_d$  is observed, given the mode  $m_i$ .

Standard belief update for an HMM is an incremental process that determines the belief-state  $b_{(k)}$  at the current time-step, given the current observations  $y_{d,(k)}$ , and the belief-state  $b_{(k-1)}$  and discrete control action  $u_{d,(k-1)}$  from the previous time-step. Belief update is a two step process. First, it uses the previous belief-state and the probabilistic transition function to predict the belief-state, denoted  $b_{(\bullet k)}[m_i]$ . Then it adjusts this prediction to account for the current observations at time-step  $k$ , resulting in the final belief-state  $b_{(k)}[m_i]$ :

$$b_{(\bullet k)}[m_i] = \sum_{m_j \in \mathcal{M}} P_T(m_i|u_{d,(k-1)}, m_j) b_{(k-1)}[m_j] \quad (1)$$

$$b_{(k)}[m_i] = \frac{b_{(\bullet k)}[m_i] P_O(y_{d,(k)}|m_i)}{\sum_{m_j \in \mathcal{M}} b_{(\bullet k)}[m_j] P_O(y_{d,(k)}|m_j)}, \quad (2)$$

### 3.2 Estimating Continuous Variables

The state of a continuous dynamic system is traditionally estimated using a state observer. In this paper we use a discrete-time model for the continuous dynamics and estimate the behavior with discrete-time extended Kalman filters[2]. This model selection is motivated by our overall goal: building a hybrid estimator for a supervisory control system that operates on a discrete-time basis.

**Definition 2.** We describe a *discrete-time model (DTM)* as a tuple  $\langle \mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}_s, \mathbf{v}_o, \mathbf{f}, \mathbf{g}, T_s, \mathbf{Q}, \mathbf{R} \rangle$ .  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{u}$  denote the vectors of *independent state-variables*  $x_1, \dots, x_n$ , *observed variables*  $y_1, \dots, y_{m_i}$  and *control variables*  $u_1, \dots, u_{m_o}$  respectively. The function  $\mathbf{f}$  specifies the dynamic evolution  $\mathbf{x}_{(k)} = \mathbf{f}(\mathbf{x}_{(k-1)}, \mathbf{u}_{(k-1)}) + \mathbf{v}_{s,(k-1)}$  and the function  $\mathbf{g}$  determines the observed variables  $\mathbf{y}_{(k)} = \mathbf{g}(\mathbf{x}_{(k)},$

$\mathbf{u}_{(k)}) + \mathbf{v}_{o,(k)}$ . The exogenous inputs  $\mathbf{v}_s$  and  $\mathbf{v}_o$  represent additive state disturbances and measurement noise. We assume that these disturbances can be modeled as a random, uncorrelated sequence with zero-mean and Gaussian distribution, and specify them by the covariance matrices  $E[\mathbf{v}_{s,(k)} \mathbf{v}_{s,(k)}^T] =: \mathbf{Q}$  and  $E[\mathbf{v}_{o,(k)} \mathbf{v}_{o,(k)}^T] =: \mathbf{R}$ .  $T_s$  denotes the *sampling-rate*, so that the time-step,  $k$ , denotes the time point,  $t_k = kT_s$ , assuming the initial time point  $t_0 = 0$ .

The disturbances and imprecise knowledge about the initial state  $\mathbf{x}_{(0)}$  make it necessary to estimate the state<sup>1</sup> by its mean  $\hat{\mathbf{x}}_{(k)}$  and covariance matrix  $\mathbf{P}_{(k)}$ . We use an extended Kalman filter for this purpose, which updates its current state, like an HMM observer, in two steps. The first step uses the model to predict the state  $\hat{\mathbf{x}}_{(\bullet,k)}$  and its covariance  $\mathbf{P}_{(\bullet,k)}$ , based on the previous estimate  $\langle \hat{\mathbf{x}}_{(k-1)}, \mathbf{P}_{(k-1)} \rangle$ , and the control input  $\mathbf{u}_{(k-1)}$ :

$$\hat{\mathbf{x}}_{(\bullet,k)} = \mathbf{f}(\hat{\mathbf{x}}_{(k-1)}, \mathbf{u}_{(k-1)}) \quad (3)$$

$$\mathbf{A}_{(k-1)} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{(k-1)}, \mathbf{u}_{(k-1)}} \quad (4)$$

$$\mathbf{P}_{(\bullet,k)} = \mathbf{A}_{(k-1)} \mathbf{P}_{(k-1)} \mathbf{A}_{(k-1)}^T + \mathbf{Q}. \quad (5)$$

This one-step ahead prediction leads to a prediction residual  $\mathbf{r}_{(k)}$  with covariance matrix  $\mathbf{S}_{(k)}$

$$\mathbf{r}_{(k)} = \mathbf{y}_{(k)} - \mathbf{g}(\hat{\mathbf{x}}_{(\bullet,k)}, \mathbf{u}_{(k)}) \quad (6)$$

$$\mathbf{C}_{(k)} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{(\bullet,k)}, \mathbf{u}_{(k)}} \quad (7)$$

$$\mathbf{S}_{(k)} = \mathbf{C}_{(k)} \mathbf{P}_{(\bullet,k)} \mathbf{C}_{(k)}^T + \mathbf{R}. \quad (8)$$

The second filter step calculates the Kalman filter gain  $\mathbf{K}_{(k)}$ , and refines the prediction as follows:

$$\mathbf{K}_{(k)} = \mathbf{P}_{(\bullet,k)} \mathbf{C}_{(k)}^T \mathbf{S}_{(k)}^{-1} \quad (9)$$

$$\hat{\mathbf{x}}_{(k)} = \hat{\mathbf{x}}_{(\bullet,k)} + \mathbf{K}_{(k)} \mathbf{r}_{(k)} \quad (10)$$

$$\mathbf{P}_{(k)} = [\mathbf{I} - \mathbf{K}_{(k)} \mathbf{C}_{(k)}] \mathbf{P}_{(\bullet,k)}. \quad (11)$$

The output of the extended Kalman filter is a sequence of mean/covariance pairs  $\langle \hat{\mathbf{x}}_{(k)}, \mathbf{P}_{(k)} \rangle$  for  $\mathbf{x}_{(k)}$ .

## 4 Concurrent Probabilistic Hybrid Automata

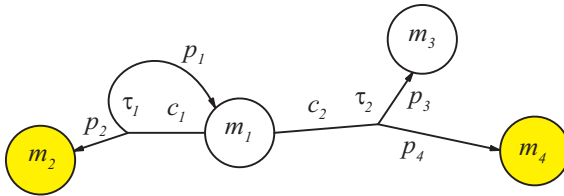
We extend hidden Markov models by incorporating discrete-time difference equations and algebraic equations for each mode to capture the dynamic evolution of the system. This leads to a hybrid model with probabilistic transitions. More specifically, we define our automaton model as:

<sup>1</sup> Throughout this paper we assume that the discrete-time models under investigation are observable in the sense of control theory.

**Definition 3.** A *discrete-time probabilistic hybrid automaton (PHA)*  $\mathcal{A}$  can be described as a tuple  $\langle \mathbf{x}, \mathbf{w}, F, T, \mathcal{X}_d, \mathcal{U}_d, T_s \rangle$ :

- $\mathbf{x}$  denotes the hybrid *state variables* of the automaton,<sup>2</sup> composed of  $\mathbf{x} = \{x_d\} \cup \mathbf{x}_c$ . The discrete variable  $x_d$  denotes the *mode* of the automaton and has finite domain  $\mathcal{X}_d$ . The *continuous state variables*  $\mathbf{x}_c = \{x_{c1}, \dots, x_{cn}\}$  capture the dynamic evolution of the automaton with domain  $\mathbb{R}^n$ .  $\mathbf{x}$  denotes the *hybrid state* of the automaton, while  $\mathbf{x}_c$  denotes the *continuous state*.
- The set of *I/O variables*  $\mathbf{w} = \mathbf{u}_d \cup \mathbf{u}_c \cup \mathbf{y}_c$  of the automaton is composed of disjoint sets of discrete input variables  $\mathbf{u}_d = \{u_{d1}, \dots, u_{dm_d}\}$  (called *command variables*), continuous *input variables*  $\mathbf{u}_c = \{u_{c1}, \dots, u_{cm_i}\}$ , and continuous *output variables*  $\mathbf{y}_c = \{y_{c1}, \dots, y_{cm_o}\}$ . The I/O variables have domain  $\mathcal{U}_d$ ,  $\mathbb{R}^{m_i}$  and  $\mathbb{R}^{m_o}$ , respectively.
- $F : \mathcal{X}_d \rightarrow F_{DE} \cup F_{AE}$  specifies the *continuous evolution* of the automaton in terms of *discrete-time difference equations*  $F_{DE}$  and *algebraic equations*  $F_{AE}$  for each mode  $x_d \in \mathcal{X}_d$ .  $T_s$  denotes the sampling period of the discrete-time difference equations.
- The finite set,  $T$ , of *transitions* specifies the probabilistic discrete evolution of the automaton in terms of tuples  $\langle \tau_i, c_i \rangle \in T$ . Each *transition function*  $\tau_i$  has an associated Boolean *guard condition*  $c_i : \mathbb{R}^n \times \mathcal{U}_d \rightarrow \{\text{true}, \text{false}\}$  and specifies the probability mass function over target modes  $x_d \in \mathcal{X}_d$ .

Fig. 2 visualizes the probabilistic transitions  $T = \{\langle \tau_1, c_1 \rangle, \langle \tau_2, c_2 \rangle\}$  for a PHA with 4 modes  $\mathcal{X}_d = \{m_1, m_2, m_3, m_4\}$ , where  $m_2$  and  $m_4$  represent failure modes. The transition function  $\tau_2$  specifies a transition from mode  $m_1$  to mode  $m_3$  with probability  $p_3$  or to mode  $m_4$  with probability  $p_4$ , whenever its associated guard  $c_2$  is satisfied.



**Fig. 2.** Probabilistic mode transition

Complex systems are modeled as a composition of concurrently operating PHA that represent the individual system components. A *concurrent probabilistic hybrid automata (cPHA)* specifies this composition as well as its interconnection to the outside world:

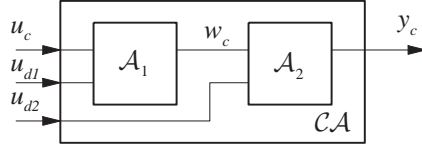
<sup>2</sup> When clear from context, we use lowercase bold symbols, such as  $\mathbf{v}$ , to denote a *set* of variables  $\{v_1, \dots, v_l\}$ , as well as a *vector*  $[v_1, \dots, v_l]^T$  with components  $v_i$ .

**Definition 4.** A concurrent probabilistic hybrid automaton (cPHA)  $\mathcal{CA}$  can be described as a tuple  $\langle A, \mathbf{u}, \mathbf{y}_c, \mathbf{v}_s, \mathbf{v}_o, N_x, N_y \rangle$ :

- $A = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_l\}$  denotes the finite set of PHAs that represent the components  $\mathcal{A}_i$  of the cPHA (we denote the components of a PHA  $\mathcal{A}_i$  by  $x_{di}, \mathbf{x}_{ci}, \mathbf{u}_{di}, \mathbf{u}_{ci}, \mathbf{y}_{ci}, F_i$ , etc.).
- The *input variables*  $\mathbf{u} = \mathbf{u}_d \cup \mathbf{u}_c$  of the automaton consists of the sets of discrete input variables  $\mathbf{u}_d = \mathbf{u}_{d1} \cup \dots \cup \mathbf{u}_{dl}$  (command variables) and continuous input variables  $\mathbf{u}_c \subseteq \mathbf{u}_{c1} \cup \dots \cup \mathbf{u}_{cl}$ .
- The *output variables*  $\mathbf{y}_c \subseteq \mathbf{y}_{c1} \cup \dots \cup \mathbf{y}_{cl}$  specify the observed output variables of the cPHA.
- The observation process is subject to (mode dependent<sup>3</sup>) additive Gaussian *sensor noise*.  $N_y : \mathcal{X}_d \rightarrow \mathbb{R}^{m \times m}$  specifies the disturbance  $\mathbf{v}_o$  in terms of the covariance matrix  $\mathbf{R}$ .
- $N_x$  specifies (mode dependent) additive Gaussian *disturbances* that act upon the continuous state variables  $\mathbf{x}_c = \mathbf{x}_{c1} \cup \dots \cup \mathbf{x}_{cl}$ .  $N_x : \mathcal{X}_d \rightarrow \mathbb{R}^{n \times n}$  specifies the disturbance  $\mathbf{v}_s$  in terms of the covariance matrix  $\mathbf{Q}$ .

**Definition 5.** The *hybrid state*  $\mathbf{x}_{(k)}$  of a cPHA at time-step  $k$  specifies the mode assignment  $\mathbf{x}_{d,(k)}$  of the mode variables  $\mathbf{x}_d = \{x_{d1}, \dots, x_{dl}\}$  and the continuous state assignment  $\mathbf{x}_{c,(k)}$  of the continuous state variables  $\mathbf{x}_c = \mathbf{x}_{c1} \cup \dots \cup \mathbf{x}_{cl}$ .

Interconnection among the cPHA components  $\mathcal{A}_i$  is achieved via shared continuous I/O variables  $w_c \in \mathbf{u}_{ci} \cup \mathbf{y}_{ci}$  only. Fig. 3 illustrates a simple example composed of 2 PHAs.



**Fig. 3.** Example cPHA composed of two PHAs

A cPHA specifies a mode dependent discrete-time model for a plant with command inputs  $\mathbf{u}_d$ , continuous inputs  $\mathbf{u}_c$ , continuous outputs  $\mathbf{y}_c$ , mode  $\mathbf{x}_d$ , continuous state variables  $\mathbf{x}_c$  and disturbances  $\mathbf{v}_s, \mathbf{v}_o$ . The continuous evolution of  $\mathbf{x}_c$  and  $\mathbf{y}_c$  can be described by

$$\mathbf{x}_{c,(k)} = \mathbf{f}_{(k)}(\mathbf{x}_{c,(k-1)}, \mathbf{u}_{c,(k-1)}) + \mathbf{v}_{s,(k-1)} \quad (12)$$

$$\mathbf{y}_{c,(k)} = \mathbf{g}_{(k)}(\mathbf{x}_{c,(k)}, \mathbf{u}_{c,(k)}) + \mathbf{v}_{o,(k)}. \quad (13)$$

The functions  $\mathbf{f}_{(k)}$  and  $\mathbf{g}_{(k)}$  are obtained by symbolically solving the set of equations  $F_1(x_{d1,(k)}) \cup \dots \cup F_l(x_{dl,(k)})$  given mode  $\mathbf{x}_{d,(k)} = [x_{d1,(k)}, \dots, x_{dl,(k)}]^T$ .

<sup>3</sup> E.g. sensors can experience different magnitudes of disturbances for different modes.

Consider the cPHA in Fig. 3 with  $\mathcal{A}_1 = \langle \{x_{d1}, x_{c1}\}, \{u_{d1}, u_c, w_c\}, F_1, T_1, \{m_{11}, m_{12}\}, \dots \rangle$  and  $\mathcal{A}_2 = \langle \{x_{d2}\}, \{u_{d2}, w_c, y_c\}, F_2, T_2, \{m_{21}, m_{22}\}, \dots \rangle$ , where  $F_1$  and  $F_2$  provide for a cPHA mode  $\mathbf{x}_{d,(k)} = [m_{11}, m_{21}]^T$  the equations  $F_1(m_{11}) = \{x_{c1,(k)} = -0.5x_{c1,(k-1)} + u_{c,(k-1)}, w_c = x_{c1}\}$  and  $F_2(m_{21}) = \{w_c = 4y_c\}$ . This leads to the discrete-time model

$$x_{c1,(k)} = -0.5x_{c1,(k-1)} + u_{c,(k-1)} + v_{s,(k-1)} \quad (14)$$

$$y_{c,(k)} = 0.25x_{c1,(k)} + v_{o,(k)}. \quad (15)$$

**Definition 6.** A *trajectory* of a cPHA  $\mathcal{CA}$  for a given input sequence  $\{\mathbf{u}_{(0)}, \mathbf{u}_{(1)}, \dots, \mathbf{u}_{(k)}\}$  is represented by a sequence of hybrid states  $\{\mathbf{x}_{(0)}, \mathbf{x}_{(1)}, \dots, \mathbf{x}_{(k)}\}$  and can be observed in terms of the sequence of observations  $\{\mathbf{y}_{c,(0)}, \mathbf{y}_{c,(1)}, \dots, \mathbf{y}_{c,(k)}\}$ .

## 5 Hybrid Estimation

To detect the onset of subtle failures, it is essential that a monitoring and diagnosis system be able to accurately extract the hybrid state of a system from a signal that may be hidden among disturbances, such as measurement noise. This is the role of a hybrid observer. More precisely:

**Hybrid Estimation Problem:** Given a cPHA  $\mathcal{CA}$ , a sequences of observations  $\{\mathbf{y}_{c,(0)}, \mathbf{y}_{c,(1)}, \dots, \mathbf{y}_{c,(k)}\}$  and control inputs  $\{\mathbf{u}_{(0)}, \mathbf{u}_{(1)}, \dots, \mathbf{u}_{(k)}\}$ , estimate the most likely hybrid state  $\hat{\mathbf{x}}_{(k)}$  at time-step  $k$ .

A *hybrid state estimate*  $\hat{\mathbf{x}}_{(k)}$  consists of a *continuous state estimate*, together with the associated *mode*. We denote this by the tuple  $\hat{\mathbf{x}}_{(k)} := \langle \mathbf{x}_{d,(k)}, \hat{\mathbf{x}}_{c,(k)}, \mathbf{P}_{(k)} \rangle$ , where  $\hat{\mathbf{x}}_{c,(k)}$  specifies the mean and  $\mathbf{P}_{(k)}$  the covariance for the continuous state variables  $\mathbf{x}_c$ . The likelihood of an estimate  $\hat{\mathbf{x}}_{(k)}$  is denoted by the *hybrid belief-state*  $h_{(k)}[\hat{\mathbf{x}}]$ .

The hybrid observer is composed of two components. The first component, an extended Kalman filter bank, maintains several continuous state estimates. The second component, a hybrid Markov observer, controls the filter bank by selecting trajectory candidates for estimation and ranking the estimated trajectories according to their belief-state. In the following we specify this operation in more detail and show (a) how the filter bank uses information from the hybrid Markov observer to guide continuous state estimation, and (b) how results from the filter bank guide the hybrid Markov observer.

### 5.1 Continuous State Estimation

Continuous state estimation is performed by a bank of extended Kalman filters which track the set of trajectories under consideration and provides an estimate  $\langle \hat{\mathbf{x}}_{i,(k)}, \mathbf{P}_{i,(k)} \rangle$  and residual  $\langle \hat{\mathbf{r}}_{i,(k)}, \mathbf{S}_{i,(k)} \rangle$  for each trajectory. The filter bank is controlled by the hybrid Markov observer that performs the mode estimation.

## 5.2 Mode Estimation - Hybrid Markov Observer

To extend HMM-style belief update to hybrid estimation, we must account for two ways in which the continuous dynamics influences the system's discrete modes. First, mode transitions depend on changes in continuous state variables (autonomous transitions), as well as discrete events injected via  $\mathbf{u}_d$ . To account for this influence we specify the hybrid probabilistic transition function  $P_{\mathcal{T}}$  to depend on continuous state  $\mathbf{x}_c$ . Second, the observation of the output variables  $\mathbf{y}_{c,(k)}$  offers important evidence that can significantly shape the hybrid state probabilities. To account for this influence we specify the hybrid probabilistic observation function  $P_{\mathcal{O}}$  to depend on  $\hat{\mathbf{x}}$  and  $\mathbf{u}_c$ .

A major difference between hybrid estimation and an HMM-style belief-state update, as well as multi-model estimation, is, however, that hybrid estimation tracks a set of trajectories, whereas standard belief-state update and multi-model estimation aggregate trajectories which share the same mode. This difference is reflected in the first of the following two recursive functions which define our hybrid estimation scheme:

$$h_{(\bullet k)}[\hat{\mathbf{x}}_i] = P_{\mathcal{T}}(\mathbf{m}_i | \hat{\mathbf{x}}_{j,(k-1)}, \mathbf{u}_{d,(k-1)}) h_{(k-1)}[\hat{\mathbf{x}}_j] \quad (16)$$

$$h_{(k)}[\hat{\mathbf{x}}_i] = \frac{h_{(\bullet k)}[\hat{\mathbf{x}}_i] P_{\mathcal{O}}(\mathbf{y}_{c,(k)} | \hat{\mathbf{x}}_{i,(k)}, \mathbf{u}_{c,(k)})}{\sum_j h_{(\bullet k)}[\hat{\mathbf{x}}_j] P_{\mathcal{O}}(\mathbf{y}_{c,(k)} | \hat{\mathbf{x}}_{j,(k)}, \mathbf{u}_{c,(k)})} \quad (17)$$

$h_{(\bullet k)}[\hat{\mathbf{x}}_i]$  denotes an intermediate hybrid belief-state, based on transition probabilities only. Hybrid estimation determines for each  $\hat{\mathbf{x}}_{j,(k-1)}$  at the previous time-step  $k - 1$  the possible transitions, thus specifying candidate trajectories to be tracked by the filter bank. Filtering provides the new hybrid state  $\hat{\mathbf{x}}_{i,(k)}$  and adjusts the hybrid belief-state  $h_{(k)}[\hat{\mathbf{x}}_i]$  based on the hybrid probabilistic observation function  $P_{\mathcal{O}}(\mathbf{y}_{c,(k)} | \hat{\mathbf{x}}_{i,(k)}, \mathbf{u}_{c,(k)})$ .

The next three subsections complete the story by outlining techniques for calculating the hybrid probabilistic transition function  $P_{\mathcal{T}}$  and the hybrid probabilistic observation function  $P_{\mathcal{O}}$ , as well as providing a tractable algorithmic formulation for hybrid estimation.

## 5.3 Hybrid Probabilistic Transition Function

A mode transition  $\mathbf{m}_j \rightarrow \mathbf{m}_i$  involves transitions  $\langle \tau_{l\eta}, c_{l\eta} \rangle \in \mathcal{T}_l$  for each component  $\mathcal{A}_l$  of the cPHA<sup>4</sup>. Given that the automaton component is in mode  $x_{dl,(k-1)} = m_j$ , the probability  $P_{\mathcal{T}l\eta}$  that it will take a transition to  $x'_{dl,(k-1)} = m_i$  is the probability that its guard  $c_{l\eta}$  is satisfied times the probability of transition  $\tau_{l\eta}(m_i)$ , given that the guard  $c_{l\eta}$  is satisfied. We assume independence of component transitions, therefore, we obtain  $P_{\mathcal{T}}$  by taking the product of all components'  $P_{\mathcal{T}l\eta}$ .

For a PHA  $\mathcal{A}_l$ , the guard  $c_{l\eta}$  is a constraint over continuous variables  $\mathbf{x}_{cl}$  and the discrete command inputs  $\mathbf{u}_{dl}$ . The guard  $c_{l\eta}$  is of the form  $[b^- \leq q_{cl\eta}(\mathbf{x}_{cl}) <$

<sup>4</sup> For symmetry, we also treat the non-transition  $m_j \rightarrow m_j$  of a PHA  $\mathcal{A}_l$  as a transition.

$b^+], where  $q_{cl\eta}(\cdot)$  is a nonlinear function,  $b^-$  and  $b^+$  denote two boundary values, and  $q_{dl\eta}(\cdot)$  is a propositional logic formula. Assuming independence of  $\mathbf{x}_{cl,(k)}$  and  $\mathbf{u}_{dl,(k)}$  allows us to determine both constraints separately as follows: The probability  $P(c_{cl\eta})$  that the guard inequality  $b^- \leq q_{cl\eta}(\mathbf{x}_{cl}) < b^+$  is satisfied, can be expressed by the volume integral over the multi-variable Gaussian distribution of the continuous state estimate  $\{\hat{\mathbf{x}}_{cl}, \mathbf{P}_l\}$  for component  $\mathcal{A}_l$$

$$P(c_{cl\eta}) = \frac{|\mathbf{P}_l|^{-1/2}}{(2\pi)^{n/2}} \int_{\mathcal{Q}} \dots \int e^{-(\mathbf{z}-\hat{\mathbf{x}}_{cl})^T \mathbf{P}_l^{-1} (\mathbf{z}-\hat{\mathbf{x}}_{cl})/2} dz_1 \dots dz_n \quad (18)$$

where  $\mathcal{Q} \subset \mathbb{R}^n$  denotes the domain of states that satisfy the guard inequality. Our current implementation calculates this cumulative distribution using a Monte Carlo[15] approach that checks the guard inequality on a sufficiently large set of normally distributed state samples with mean  $\hat{\mathbf{x}}_{cl}$  and covariance matrix  $\mathbf{P}_l$ . An open research issue is to compute  $P(c_{cl\eta})$  more efficiently, through a combination of restricting and approximating the function  $q_{cl\eta}$ .

The discrete constraint  $q_{dl\eta}(\mathbf{u}_d)$  has probability  $P(c_{dl\eta}) = 1$  or  $P(c_{dl\eta}) = 0$ , according to its truth value. The assumed independence of the continuous state and the discrete input leads to

$$P_{\mathcal{T}l\eta} = P(c_{cl\eta})P(c_{dl\eta})\tau_{l\eta}(m_i) \quad (19)$$

## 5.4 Hybrid Probabilistic Observation Function

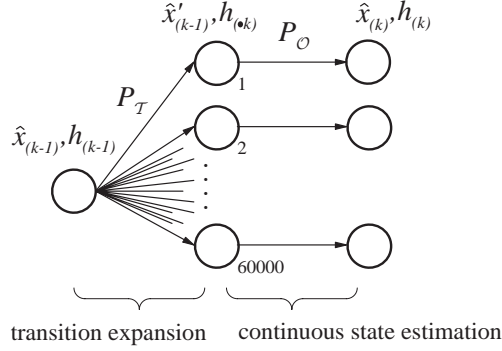
The extended Kalman filters calculate state estimates for the continuous state  $\hat{\mathbf{x}}_{c,(k)}$ . This involves calculating the measurement residual  $\mathbf{r}_{(k)}$  (Eq. 6) and its associated covariance matrix  $\mathbf{S}_{(k)}$  (Eq. 8). From this estimate we can calculate  $P_{\mathcal{O}}(\mathbf{y}_{(k)}|\hat{\mathbf{x}}_{(k)}, \mathbf{u}_{c,(k)})$  using the standard relation for the multi-variable Gaussian probability density function without normalization<sup>5</sup>:

$$P_{\mathcal{O}}(\mathbf{y}_{(k)}|\hat{\mathbf{x}}_{(k)}, \mathbf{u}_{c,(k)}) = e^{-\mathbf{r}_{(k)}^T \mathbf{S}_{(k)}^{-1} \mathbf{r}_{(k)}/2} \quad (20)$$

## 5.5 Tracking the Most Likely Trajectories

Tracking all possible trajectories of a system is almost always intractable because the number of trajectories becomes too large after only a few time steps. As an example consider a cPHA with 10 components. The components have on average 5 modes and each mode has on average 3 successor states. This cPHA represents an automaton with  $5^{10} \approx 10000000$  modes and hybrid estimation, as formulated above, lead to  $(3^{10})^k$  trajectories to be tracked at time-step  $k$ . Fig. 4 visualizes this blowup for a single hybrid estimation step.

<sup>5</sup> We omit the normalization term  $\frac{|\mathbf{S}_{(k)}|^{-1/2}}{(2\pi)^{n/2}}$  as our any-time any-space estimation algorithm requires  $0 \leq P_{\mathcal{O}} \leq 1$ . Normalization is already ensured by Eq. 17. Furthermore, [7] explicitly suggest this change based on the observation that the normalization term has nothing to do with the identification of the correct mode.



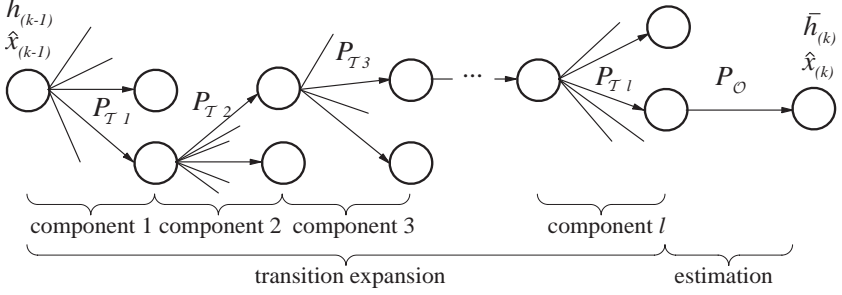
**Fig. 4.** Hybrid Estimation at time-step  $k$  with single estimate  $\hat{\mathbf{x}}_{(k-1)}$ .

The problem of exponential growth is a well known drawback of full hypothesis multi-model estimation and lead to the development of approximative schemes, such as the generalized pseudo-Bayesian and the interacting multiple model algorithm[8] and adaptive extensions[9,16]. These algorithms slow the effect of the exponential growth by merging the estimates. However, in general they are not effective enough to cope with many hybrid estimation problems as the number of hypotheses at each time-step is still beyond their scope (e.g.  $3^{10} \approx 60000$  in the “small” example above).

We address this problem with an any-time, any-space solution that dynamically adjusts the number of trajectories tracked in order to fit within the processor’s computational and memory limits. The Livingston system[1] successfully utilized such a focusing scheme for model-based diagnosis and configuration management. This approach succeeds because a small subset of the set of possible modes of a system is typically sufficient to cover most of the probability space. For hybrid estimation we adopt an analogous scheme that enumerates a focused subset of the possible trajectories by framing hybrid estimation as *beam search* that maintains the fringe  $\mathbf{X}_{(k)} = \{\hat{\mathbf{x}}_{1,(k)}, \dots, \hat{\mathbf{x}}_{m,(k)}\}$  of the  $m$  most likely trajectories. Key to this approach is an any-time, any-space enumeration scheme that provides, at each time-step  $k$ , the focussed subset of most likely states  $\{\hat{\mathbf{x}}_{1,(k)}, \dots, \hat{\mathbf{x}}_{m,(k)}\}$ , without having to calculate a prohibitively large number of hybrid state estimates  $\hat{\mathbf{x}}_{i,(k)}$ .

In our modeling framework, we assume that the mode transitions of the system’s components are independent of each other. Therefore, we consider possible mode transitions for a hybrid estimate  $\hat{\mathbf{x}}_{i,(k-1)}$  and the discrete (command) input  $\mathbf{u}_{(k-1)}$  componentwise. This allows us to formulate enumeration as best-first search, using A\* (Fig. 5 shows the corresponding search tree for a fringe size of 1). Best-first search expands search tree nodes in increasing order of their utility  $f(n)$ . We define the utility of a node  $n$  as  $f(n) = f_1(n) + f_2(n)$ , where

$$f_1(n) = -\ln(h_{(k-1)}[\hat{\mathbf{x}}_j]) - \sum_{i=1}^{\nu} \ln(P_i), \quad P_i = \begin{cases} P_{\tau i} & i = 1, \dots, l \\ P_{\mathcal{O}} & i = l + 1 \end{cases} \quad (21)$$



**Fig. 5.** Search Tree for Hybrid Estimation at time-step  $k$

denotes the cost of the path from a root node (state estimate  $\hat{\mathbf{x}}_{j,(k-1)}$  with initial cost  $-\ln(h_{(k-1)}[\hat{\mathbf{x}}_j])$ ) to a node within the search tree (e.g. considering transitions in components  $1, \dots, \nu$ , whenever  $\nu \leq l$ ).  $f_2(n)$  estimates the cost for the best path from node  $n$  to a goal node by considering the best transitions for the remaining components, more specifically<sup>6</sup>

$$f_2(n) = \sum_{i=\nu+1}^l -\ln(\max_{\eta} P_{T_{i\eta}}). \quad (22)$$

The search is optimal and complete, as  $f_2(n)$  *never overestimates* the cost to reach a goal (i.e.  $f_2$  is a *admissible heuristic*). It provides, at each time-step  $k$ , the most likely successor states  $\{\hat{\mathbf{x}}_{1,(k)}, \hat{\mathbf{x}}_{2,(k)}, \dots\}$  of the fringe  $\mathbf{X}_{(k-1)}$  together with the un-normalized hybrid belief  $\bar{h}_{i,(k)} = h_{(\bullet,k)}[\hat{\mathbf{x}}_i]P_{\mathcal{O}}(\mathbf{y}_{c,(k)}|\hat{\mathbf{x}}_{i,(k)}, \mathbf{u}_{c,(k)})$  in consecutive order.

## 6 Example Continued

We demonstrate hybrid mode estimation for two operational conditions of the BIO-Plex system: (1) detection of crew entry into the PGC ( $m_{p4} \rightarrow m_{p5}$ ) and (2) a failure of the lighting system that reduces the light intensity in the PGC by 20% ( $m_{l2} \rightarrow m_{l4}$ ). The dynamic behavior of the  $CO_2$  concentration (in ppm) at the modes  $m_{p4}$  and  $m_{p5}$  with operational lighting system ( $m_{l2}$ ), for instance, is governed by ( $T_s = 1$  [min]):

$$\begin{aligned} x_{c1,(k)} &= u_{c,(k-1)} \\ x_{c2,(k)} &= x_{c2,(k-1)} + 11.8373[f(x_{c,(k-1)}) + x_{c1,(k-1)} + h_{c,(k-1)}] \\ y_{c1,(k)} &= x_{c1,(k)} \\ y_{c2,(k)} &= x_{c2,(k)} \\ f(x_{c,(k)}) &= -1.4461 \cdot 10^{-2} \left[ 72.0 - 78.89e^{-\frac{x_{c,(k)}}{400.0}} \right], \end{aligned}$$

<sup>6</sup> The maximum value for the probabilistic observation function ( $P_{\mathcal{O}} = 1$ ) can be omitted as  $\ln(1) = 0$ .

where  $h_c$  accounts for the exhaled  $CO_2$  ( $m_{p4} : h_c = 0, m_{p5} : h_c = 0.3$ ). The noisy measurement of the controlled  $CO_2$  concentration (black) and its estimation (red/gray) are given in the left graph of Fig. 6. The crew enters the PGC at time step 900 and cause an adaption of the gas injection. Hybrid mode estimation filters this noisy measurement and detects the mode change immediately at  $k = 901$ . The light fault is then injected at time-step 1100 and diagnosed 17 time-steps later. The graphs to the right in Fig. 6 show the mode of the leading trajectory estimate for the plant growth chamber and the lighting system.

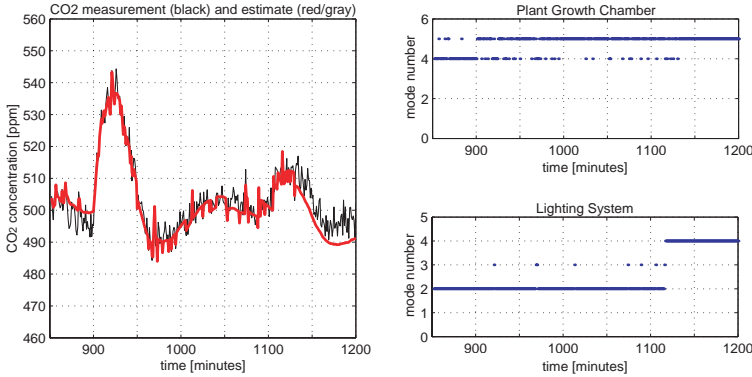


Fig. 6. Estimation results for BIO-Plex scenarios

## 7 Implementation and Discussion

The implementation of our hybrid estimation scheme is written in Common LISP. The hybrid estimator uses a cPHA description and performs estimation, as outlined above. Although designed to operate online, we used the estimator to determine the hybrid state of the PGC based on input data gathered from simulating a subset of NASA JSC's CONFIG model for the BIO-Plex system.

Optimized model-based estimation schemes, such as Livingstone[1], utilize *conflicts* to focus the search operation. A conflict is a (partial) mode assignment that makes a hypothesis very unlikely. The decompositional model-based learning system, Moriarty[17], introduced an algebraic version of conflicts, called *dissents*. We are currently reformulating dissents for hybrid systems and investigate their incorporation in the search scheme. This will lead to an overall framework for hybrid estimation that unifies our previous work on Livingstone and Moriarty.

A novel capability of discrete model-based diagnosis methods is the ability to handle *unknown modes* where no assumption is made about the behavior of one or several components of the system. We are in the process of incorporating this novel capability of model-based diagnosis into our estimation scheme by

calculating partial filters. The filters are based on causal analysis of the specified components and their interconnection within the cPHA model. Incorporating unknown modes provides a robust estimation scheme that can cope with unmodeled situations and partial information.

## References

1. Williams, B., Nayak, P.: A model-based approach to reactive self-configuring systems. In: Proc. of the 13th Nat. Conf. on Artificial Intelligence (AAAI-96). (1996)
2. Anderson, B., Moore, J.: Optimal Filtering. Prentice Hall (1979)
3. Branicky, M.: Studies in Hybrid Systems: Modeling, Analysis, and Control. PhD thesis, Department of Electrical Engineering and Computer Science, MIT (1995)
4. Henzinger, T.: The theory of hybrid automata. In: Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96) (1996) 278–292
5. Hu, J., Lygeros, J., Sastry, S.: Towards a theory of stochastic hybrid systems. In Lynch, N., Krogh, B., eds.: Hybrid Systems: Computation and Control. Lecture Notes in Computer Science, 1790. Springer (2000) 160–173
6. Nancy Lynch, Roberto Segala, F.V.: Hybrid I/O automata revisited. In M.D. Di Benedetto, A.S.V., ed.: Hybrid Systems: Computation and Control, HSCC 2001. Lecture Notes in Computer Science, 2034. Springer Verlag (2001) 403–417
7. Maybeck, P., Stevens, R.: Reconfigurable flight control via multiple model adaptive control methods. IEEE Transactions on Aerospace and Electronic Systems **27** (1991) 470–480
8. Bar-Shalom, Y., Li, X.: Estimation and Tracking. Artech House (1993)
9. Li, X., Bar-Shalom, Y.: Multiple-model estimation with variable structure. IEEE Transactions on Automatic Control **41** (1996) 478–493
10. McIlraith, S., Biswas, G., Clancy, D., Gupta, V.: Towards diagnosing hybrid systems. In: Proc. of the 10th Internat. Workshop on Principles of Diagnosis. (1999) 194–203
11. Narasimhan, S., Biswas, G.: Efficient diagnosis of hybrid systems using models of the supervisory controller. In: Proc. of the 12th Internat. Workshop on Principles of Diagnosis. (2001) 127–134
12. Zhao, F., Koutsoukos, X., Haussecker, H., Reich, J., Cheung, P.: Distributed monitoring of hybrid systems: A model-directed approach. In: Proc. of the Internat. Joint Conf. on Artificial Intelligence (IJCAI'01). (2001) 557–564
13. Lerner, U., Parr, R., Koller, D., Biswas, G.: Bayesian fault detection and diagnosis in dynamic systems. In: Proc. of the 17th Nat. Conf. on Artificial Intelligence (AAAI'00). (2000)
14. McIlraith, S.: Diagnosing hybrid systems: a Bayesian model selection approach. In: Proc. of the 11th Internat. Workshop on Principles of Diagnosis. (2000) 140–146
15. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer-Verlag (1999)
16. Hanlon, P., Maybeck, P.: Multiple-model adaptive estimation using a residual correlation Kalman filter bank. IEEE Transactions on Aerospace and Electronic Systems **36** (2000) 393–406
17. Williams, B., Millar, B.: Decompositional, model-based learning and its analogy to diagnosis. In: Proc. of the 15th Nat. Conf. on Artificial Intelligence (AAAI-98). (1998)