Paper submitted to IJCAI-01

Mode Estimation of Probabilistic Hybrid Systems

Brian C. Williams, Michael Hofbaur and Thomas Jones
Massachusetts Institute of Technology, Rm. 37-381
77 Massachusetts Ave., Cambridge, MA 02139 USA
E-mail: {williams,hofbaur,jonest}@mit.edu

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 *probabilistic hybrid automata* (PHA), that merge hidden Markov models (HMM) with continuous dynamical system models. Second, we introduce *hybrid estimation* as a method for tracking and diagnosing PHA, 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. This approach pursues the most promising estimates, based on a statistical measure of the probability that an estimate will turn out likely.

1 Introduction

The year 2000 was kicked off with two missions to Mars, following on the heals of the highly successful Mars Pathfinder mission. Just before the first mission, Mars Climate Orbiter, reached Mars orbital insertion, the operations team identified contradictory attitude estimates, one suggesting that the vehicle was coming in at an elevation too low to successfully achieve insertion. Unfortunately time did not allow the team to resolve this inconsistency or to plan a reliable course correction before insertion. The Climate Orbiter proceeded until it burned up in the Martian atmosphere. After extensive investigation it was found that a table used by the navigation system, which describes the small forces impinging upon the space vehicle, was mistranslated into the wrong units. This bug introduced a small, but indiscernible failure that over a lengthy time period produced the loss of the orbiter.

The problem of misinterpreting a system's dynamics was punctuated later in the year when the Orbiter's sibling, 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 50 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.

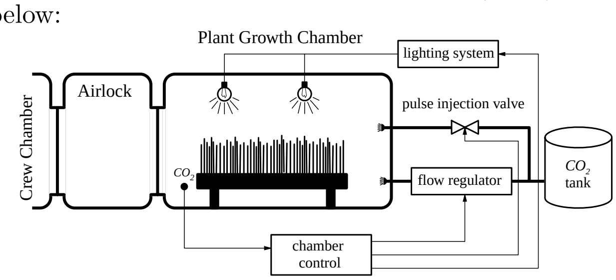
The above case study is a dramatic instance of a common problem – increasingly complex systems are being developed whose failure symptoms are nearly indiscernible up until a catastrophic result occurs. To tackle this problems we must address two issues. First, these failures are manifest through a coupling between a system's continuous dynamics and its evolution through different behavior modes. Hence to address this problem we need hybrid monitoring and diagnosis capabilities that are be able to track a system's behavior along both its continuous state changes 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 statistical methods need to be applied to separate the noise from the true dynamics.

Looking to the literature, deductive mode estimation techniques [Williams and Nayak, 1996] have been successfully demonstrated on space systems that have the ability to track a system as it moves through a series of discrete behavioral modes. Mode estimation achieves this function through a combination of logical deduction and belief update on a hidden Markov model. However, these and related methods fall short in that they presumes that a failure always manifests a symptom that can be easily distinguished from system noise.

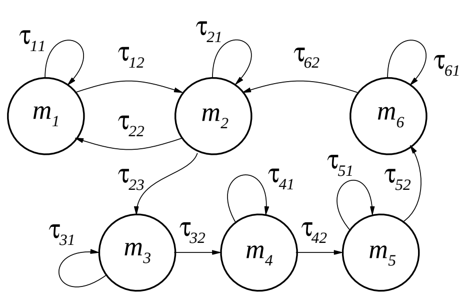
We address this challenge by extending deductive mode estimation capabilities so that they reason about continuous dynamics using classical methods for state estimation. This paper focuses on one piece of this problem: the interplay between discrete probabilistic mode changes and continuous dynamics. After a motivating example, we discuss traditional methods for separately estimating discrete and continuous behaviors. We then introduce a modeling formalism, called *probabilistic hybrid automata* (PHA), that merges hidden Markov models with continuous dynamical system models. Third, we introduce a method called *hybrid mode estimation* that tracks and diagnoses PHA, by creating a hybrid HMM observer that uses the results of continuous state estimates to estimate a system's mode changes, and that coordinates the actions of a set of continuous state observers. Finally, we introduce a novel, any-time, any-space algorithm for computing approximate hybrid estimates, that pursues the most "promising" estimates, based on a statistical measure of the probability that an estimate will turn out likely.

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 provides all the air, water, and most of the food (up to 90%) for a crew of four on a continuous basis. Plants are grown in plant growth chambers, where they provide food for the crew and convert the exhaled CO_2 into the required O_2. In order to maintain a closed-loop system, it is necessary to control the resource exchange between the plant growth chambers and the crew chambers. 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 below:



The PGC nominal operational modes are shown below:



The operational modes execute according to the following typical sequence: The control system maintains an artificial 20/4 hour day / night (m_1) schedule for the plants to obtain optimal growth and gas conversion. Several operational modes are possible during the day phase. Most of the time is spent in a plant growth mode (m_2) where the CO_2 concentration is kept at the level of 1200 ppm, which is optimal for plant growth and gas conversion. A flow controller maintains a continuous flow of CO_2 from a storage tank to compensate for the CO_2 used for photo-synthesis at m_2. The high CO_2 concentration in plant growth mode is not suitable for humans, so the gas concentration must be lowered to 500 ppm in case crew members request to enter the chamber, for example, for harvesting, re-planting or other service activities. The transition to the lower set-point is achieved by turning the CO_2 injection off (m_3). The system then maintains a set-point at 500 ppm (m_4) and releases the door. At this stage, crew members are allowed to enter the PGC (τ_{42}). Safety precaution requires the system to inhibit gas injection via the pulse-injection path while crew members are in the PGC (m_5). The gas concentration is raised to the growth optimal set-point using full continuous and pulse injection (m_6) as soon as the crew exits the chamber and closes the door (τ_{52}).

The safety precaution at mode m_5 highlights the hybrid mode estimation task. Mode m_5 differs from mode m_4 only by a slight change in CO_2 balance, due to the CO_2 exhaled by the crew in the PGC. This small quantitative difference is subject to detection by a hybrid mode estimation scheme. Besides tracking operational modes of the system it is the task of a hybrid mode estimation scheme to detect faults such as stuck-open or clogged injection valves, or a suddenly opened door. To handle these failures we extend classical hybrid modeling schemes[Henzinger, 1996; Branicky, 1995] by introducing probabilistic mode transitions.

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 from 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 $\langle \mathcal{M}, \mathcal{Y}_d, \mathcal{U}_d, P_\Theta, P_\mathcal{T}, P_\mathcal{O} \rangle$. $\mathcal{M}$, $\mathcal{Y}_d$ 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*,

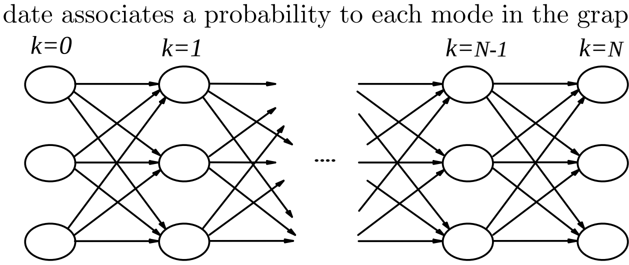
$P_{\mathcal{T}}(m_i|u_d, m_j)$, describes the probability of transitioning from mode $m_{j,(k-1)}$ to $m_{i,(k)}$ at time-step k, given a discrete control action $u_{d,(k-1)}$. The *observation function* $P_{\mathcal{O}}(y_d|m_i)$ describes the probability that a discrete value $y_{d,(k)}$ is observed at k, given $m_{i,(k)}$.

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 $y_{d,(k)}$, 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 $b_{(k)}[m_i]$:

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

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

The space of possible trajectories of an HMM can be visualized using a *Trellis diagram*, which enumerates all possible modes at each time-step and all transitions between modes at adjacent time-steps. Belief update associates a probability to each mode in the graph.



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 Kalman filter[Gelb, 1974] that captures the continuous dynamics based on a discrete-time model of the dynamic system.

Definition 2 A *discrete-time model (DTM)* can be described by a tuple $\langle \mathbf{x}_c, \mathbf{y}_c, \mathbf{u}_c, \mathbf{v}_c, \mathbf{f}_c, \mathbf{g}_c \rangle$. $\mathbf{x}_c$, $\mathbf{y}_c$, $\mathbf{u}_c$, $\mathbf{v}_c$ denote the finite sets of *independent state-variables* x_{ci}, *observed variables* y_{ci}, *control variables* u_{ci} and *exogenous input variables* v_{ci}, respectively. The *state transition function* $\mathbf{f}_c$ specifies the evolution of the state-variables $\mathbf{x}_{c,(k+1)} = \mathbf{f}_c(\mathbf{x}_{c,(k)}, \mathbf{u}_{c,(k)}, \mathbf{v}_{c,(k)})$ and the *output function* $\mathbf{g}_c$ determines the observed variables $\mathbf{y}_{c,(k)} = \mathbf{g}_c(\mathbf{x}_{c,(k)}, \mathbf{v}_{c,(k)})$.

We assume that $\mathbf{f}_c$ and $\mathbf{g}_c$ can be adequately approximated by a set of linear time-invariant difference equations:

$$\mathbf{x}_{c,(k+1)} = \mathbf{A}\mathbf{x}_{c,(k)} + \mathbf{B}\mathbf{u}_{c,(k)} + \mathbf{v}_{c1,(k)} \quad (3)$$

$$\mathbf{y}_{c,(k)} = \mathbf{C}\mathbf{x}_{c,(k)} + \mathbf{v}_{c2,(k)}, \quad (4)$$

where the exogenous inputs are partitioned into $\mathbf{v}_{c1,(k)}$ and $\mathbf{v}_{c2,(k)}$ to denote input disturbance and measurement noise, respectively. 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 $E[\mathbf{v}_{c1,(k)} \mathbf{v}_{c1,(k)}{}^T] =: \mathbf{Q}$ and $E[\mathbf{v}_{c2,(k)} \mathbf{v}_{c2,(k)}{}^T] =: \mathbf{R}$.

The disturbances and imprecise knowledge about the initial state $\mathbf{x}_{c,(0)}$ make it necessary to estimate the state by its mean $\hat{\mathbf{x}}_{c,(k)}$ and covariance matrix $\mathbf{P}_{(k)}$. We use a 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 $\hat{\mathbf{x}}_{c,(\bullet k)}$ and its covariance $\mathbf{P}_{(\bullet k)}$, based on the previous estimate $\langle \hat{\mathbf{x}}_{c,(k-1)}, \mathbf{P}_{(k-1)} \rangle$, and the control input $\mathbf{u}_{c,(k-1)}$:

$$\hat{\mathbf{x}}_{c,(\bullet k)} = \mathbf{A}\hat{\mathbf{x}}_{c,(k-1)} + \mathbf{B}\mathbf{u}_{c,(k-1)} \quad (5)$$

$$\mathbf{P}_{(\bullet k)} = \mathbf{A}\mathbf{P}_{(k-1)}\mathbf{A}^T + \mathbf{Q} \quad (6)$$

The second step calculates the Kalman filter gain $\mathbf{K}_{(k)}$ and refines the prediction based on the current measurement $\mathbf{y}_{c,(k)}$, and produces the estimate:

$$\mathbf{K}_{(k)} = \mathbf{P}_{(\bullet k)}\mathbf{C}^T[\mathbf{C}\mathbf{P}_{(\bullet k)}\mathbf{C}^T + \mathbf{R}]^{-1} \quad (7)$$

$$\hat{\mathbf{x}}_{c,(k)} = \hat{\mathbf{x}}_{c,(\bullet k)} + \mathbf{K}_{(k)}[\mathbf{y}_{(k)} - \mathbf{C}\hat{\mathbf{x}}_{c,(\bullet k)}] \quad (8)$$

$$\mathbf{P}_{(k)} = [\mathbf{P}_{(\bullet k)}{}^{-1} + \mathbf{C}^T\mathbf{R}\mathbf{C}]^{-1}. \quad (9)$$

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

4 Probabilistic Hybrid Automata

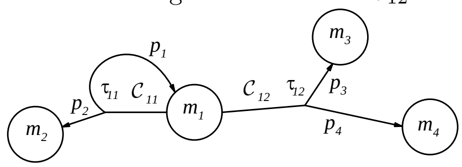
Given this background, our task is to develop a state observer for *probabilistic hybrid automata (PHA)*. A PHA is a hidden Markov model, encoded as a set of modes that exhibit a continuous dynamical behavior, expressed by difference equations. More precisely:

Definition 3 A *probabilistic hybrid automaton* $\mathcal{PHA}$ can be described as a tuple $\langle \mathcal{M}, \mathbf{x}_c, \mathbf{y}_c, \mathcal{U}, \mathcal{F}_c, \mathcal{G}_c, \mathcal{T} \rangle$.

- The finite set $\mathcal{M}$ denotes the modes $m_i \in \mathcal{M}$ of the automaton.
- $\mathbf{x}_c$ and $\mathbf{y}_c$ denote the set of independent continuous *state-variables* and *output variables*, respectively. The set of input variables $\mathcal{U} = \mathbf{u}_c \cup \mathbf{u}_d \cup \mathbf{v}_c$ is partitioned into *continuous control variables* $\mathbf{u}_c$, *continuous exogenous variables* $\mathbf{v}_c$, and *discrete control variables* $\mathbf{u}_d$. Components of continuous variables range over $\Re$, whereas components of discrete variables range over finite domains $\mathcal{D}$.
- The sets $\mathcal{F}_c$ and $\mathcal{G}_c$ associate with each mode $m_i \in \mathcal{M}$ functions $\mathbf{f}_{ci}$ and $\mathbf{g}_{ci}$ that govern the continuous dynamics exhibited at mode m_i by $\mathbf{x}_{c,(k+1)} = \mathbf{f}_{ci}(\mathbf{x}_{c,(k)}, \mathbf{u}_{c,(k)}, \mathbf{v}_{c,(k)})$ and $\mathbf{y}_{c,(k)} = \mathbf{g}_{ci}(\mathbf{x}_{c,(k)}, \mathbf{v}_{c,(k)})$.
- $\mathcal{T}$ specifies for each mode $m_{i,(k)}$ a set of transition functions $\mathcal{T}_i = \{\tau_{i1}, \ldots, \tau_{in}\}$. Each transition function τ_{ij} has an associated guard condition

$\mathcal{C}_{ij}(\mathbf{x}_{c,(k)}, \mathbf{u}_{d,(k)})$ and specifies the probability distribution over target modes $m_{l,(k+1)}$ together with an assignment for $\mathbf{x}_{c,(k+1)}$.

The figure below shows a transition function for a mode m_1 with $\mathcal{T}_1 = \{\tau_{11}, \tau_{12}\}$. The transition function τ_{12} specifies a transition to mode m_3 with probability p_3 or to mode m_4 with probability p_4, whenever its guard condition $\mathcal{C}_{12}$ is satisfied.



The probabilistic transition function adds expressiveness and provides an elegant way to model failure transitions. Furthermore, allowing the mode transitions to be triggered by the dynamic behavior, i.e. by a state-variable reaching the domain-boundary for a mode, provides a significant advantage to other hybrid systems diagnosis schemes[McIlraith *et al.*, 1999] which were restricted to non-autonomous mode transitions.

The **hybrid state** $\mathbf{X}_{(k)}$ of a probabilistic hybrid automaton at time-step k is specified by the tuple $\langle m_{(k)}, \mathbf{x}_{c,(k)} \rangle$, where $m_{(k)} \in \mathcal{M}$ specifies the mode of the automaton and $\mathbf{x}_{c,(k)}$ specifies the values of the state-variables. We use the shorter notation $m_{i,(k)}$ to denote $m_{(k)} = m_i$.

A probabilistic hybrid automaton is a model for a plant with inputs $\mathbf{u}_c$, $\mathbf{u}_d$ and $\mathbf{v}_c$, output $\mathbf{y}_c$ and internal hybrid state $\langle m, \mathbf{x}_c \rangle$. The behavior of the $\mathcal{PHA}$, called the **trajectory**, is represented by the sequence of hybrid states $\mathbf{t} = \{\mathbf{X}_{(0)}, \mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(k)}\}$

5 Hybrid Estimation

To detect the onset of subtle failures, such as Mars Climate Orbiter's conversion error in its small forces table, 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 probabilistic hybrid automaton $\mathcal{PHA}$ for a system, a sequence of observations $(\mathbf{y}_{(0)}, \ldots, \mathbf{y}_{(k)})$, the history of control inputs $(\mathbf{u}_{(0)}, \ldots, \mathbf{u}_{(k-1)})$, and statistical measures of exogenous inputs $\mathbf{v}$, generate the most likely hybrid state 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 $\langle m_i, \hat{\mathbf{x}}_{c,(k)}, \mathbf{P}_{(k)} \rangle$ where $\hat{\mathbf{x}}_{c,(k)}$ specifies the mean and $\mathbf{P}_{(k)}$ the covariance for the state-variables $\mathbf{x}_{c,(k)}$. The probability of being in a hybrid state $\mathbf{X}$ is specified by the hybrid belief-state $h_{(k)}[\mathbf{X}]$. This allows us to denote the likelihood $h_{(k)}[\hat{\mathbf{X}}_i]$ of estimate $\hat{\mathbf{X}}_{i,(k)} = \langle m_i, \hat{\mathbf{x}}_{ci,(k)}, \mathbf{P}_{i(k)} \rangle$.

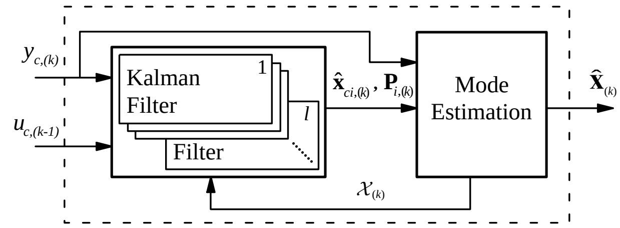


Figure 1: Information flow within the hybrid observer

The hybrid observer is composed of two components (see Figure 1). The first generalizes from the Kalman filter, and is responsible for maintaining continuous state variable estimates. The second generalizes from the Markov observer, and is responsible for maintaining hybrid mode estimates. In the next two sections we specify how the Kalman filter uses information from the Markov observer to guide continuous state estimation, and how the Markov observer uses results from the Kalman filter to guide mode estimation.

5.1 Continuous State Estimation

Continuous state estimation is performed by a bank of Kalman filters which track the set of trajectories under consideration. The filter bank provides a state-variable estimate $\langle \hat{\mathbf{x}}_{i,(k)}, \hat{\mathbf{x}}_{i,(\bullet k)}, \mathbf{P}_{i,(k)}, \mathbf{P}_{i,(\bullet k)} \rangle$ for each trajectory as input to mode estimation. Mode estimation controls the state estimation by specifying $\mathcal{X}_{(k-1)}$, the fringe of the trajectories that the Kalman filters should track. This allows us to dynamically invoke a Kalman filter whenever a transition occurs and generalizes the concept of *multi-model estimation*[Stengel, 1994]. [Rinner and Kuipers, 1999] also dynamically adapt an observer bank, but are used to support mixed qualitative/quantitative estimation.

5.2 Mode Estimation

To extend HMM-style belief update to hybrid mode 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 state variables, as well as discrete events injected via $\mathbf{u}_d$. More generally, modes of continuous devices are often defined to apply over a region of the device's state space. A mode transition occurs whenever the state variables reach this region's boundary. To account for this influence we modify $P_{\mathcal{T}}$ to depend on continuous state-variables $\mathbf{x}_c$. Second, the observed variables $\mathbf{y}$ include real-valued variables $\mathbf{y}_c$. These continuous observations offer important evidence that can significantly shape the hybrid state probabilities. For example, suppose a valve is presumed closed, but a slow trickle is observed. If it is measured for only a brief moment, it can be accounted for by noise, but as the time increases it becomes more likely that the valve is open. To account for this influence we modify $P_{\mathcal{O}}$ to depend on $\mathbf{x}_c$.

A major difference between hybrid mode estimation and an HMM-style belief-state update is, however, that hybrid mode estimation tracks a set of trajectories, whereas standard belief-state update aggregates trajectories. This difference is reflected in the first of the following two recursive functions which define hybrid mode estimation:

$$h_{(\bullet k)}[\hat{\mathbf{X}}_i] = P_{\mathcal{T}}(\mathcal{C}_{lj}|\hat{\mathbf{x}}_{cl,(k-1)}, \mathbf{u}_{d,(k-1)})\tau_{lj}[m_i]h_{(k)}[\hat{\mathbf{X}}_l]$$

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

Once again, $h_{(\bullet k)}[\hat{\mathbf{X}}_i]$ denotes an intermediate hybrid belief-state, predicted from the previous belief-state using the transition probabilities. This operation determines for each $\hat{\mathbf{X}}_{j,(k-1)}$ the possible transitions, thus specifying the set $\mathcal{X}_{(k-1)}$ of candidate trajectories to be tracked by the continuous state estimation. $h_{(k)}[\hat{\mathbf{X}}_i]$ denotes the final hybrid belief-state, adjusted to account for the current observations at time-step k.

The next two subsections complete the story by outlining techniques for calculating the hybrid probabilistic transition function $P_{\mathcal{T}}(\mathcal{C}_{lj}|\hat{\mathbf{x}}_{cl,(k-1)}, \mathbf{u}_{d,(k-1)})$ and the hybrid probabilistic observation function $P_{\mathcal{O}}(\mathbf{y}_{(k)}|\hat{\mathbf{X}}_{j,(k)})$.

Hybrid Transition Function

A transition τ includes a condition $\mathcal{C}$, called a *guard*, that must be satisfied, in order for the transition to be taken. Given that the automaton is in mode m_l, the probability that it will take a transition to m_i is the probability that its guard is satisfied, that is, $P_{\mathcal{T}}(\mathcal{C}_{lj}|\hat{\mathbf{x}}_{cl,(k-1)}, \mathbf{u}_{d,(k-1)})$, times the probability of transition $\tau_{ij}[m_i]$, given that the guard $\mathcal{C}_{lj}$ is satisfied.

For a probabilistic hybrid automata, the guard $\mathcal{C}_{lj}$ is a constraint over continuous variables $\mathbf{x}_c$ and the discrete control inputs $\mathbf{u}_d$. $\mathcal{C}_{lj}$ is of the form $[\mathbf{b}^- \leq \mathbf{q}_{clj}(\mathbf{x}_c) \leq \mathbf{b}^+] \wedge \mathbf{q}_{dlj}(\mathbf{u}_c)$, where $\mathbf{q}_{clj}$ is a vector function, $\mathbf{b}^-$ and $\mathbf{b}^+$ denote two vectors, and $\mathbf{q}_{dlj}$ is a propositional logic formulae. Assuming independence of $\mathbf{x}_c$ and $\mathbf{u}_d$ allows us to determine both constraints separately.

The probability $P(\mathcal{C}_{clj})$ that the continuous constraint $\mathbf{b}^- \leq \mathbf{q}_{clj}(\mathbf{x}_c) \leq \mathbf{b}^+$ is satisfied, can be expressed by the volume integral over the multi-variate Gaussian distribution of the state-variable estimate $\{\hat{\mathbf{x}}_{lc}, \mathbf{P}_l\}$:

$$P(\mathcal{C}_{clj}) = \frac{|\mathbf{P}_l|^{-1/2}}{(2\pi)^{n/2}} \int_{\mathcal{Q}} \int e^{-\frac{(\mathbf{x}-\hat{\mathbf{x}}_{lc})^T \mathbf{P}_l^{-1}(\mathbf{x}-\hat{\mathbf{x}}_{lc})}{2}} dx_1 \ldots dx_n$$

where $\mathcal{Q} \subset \Re^n$ denotes the domain specified by the continuous constraint $\mathbf{b}^- \leq \mathbf{q}_{clj}(\mathbf{x}_c) \leq \mathbf{b}^+$. Routines for computing this cumulative distribution are provided by a variety of statistical and numerical packages, such as Splus or Matlab. An open research issue is to compute the volume integral efficiently, through a combination of restricting and approximating the form of $\mathbf{q}$.

The discrete constraint $\mathbf{q}_{dlj}(\mathbf{u}_c)$ has probability $P(\mathcal{C}_{dlj}) = 1$ or $P(\mathcal{C}_{dlj}) = 0$, according to its truth value. Independence of the continuous state and the discrete input leads to

$$P_{\mathcal{T}}(\mathcal{C}_{lj}|\hat{\mathbf{x}}_{cl,(k-1)}, \mathbf{u}_{d,(k-1)}) = P(\mathcal{C}_{clj})P(\mathcal{C}_{dlj}) \qquad (10)$$

Hybrid Observation Function

The Kalman filters provide us with an estimate of the continuous state-variables $\hat{\mathbf{x}}_{ci,(k)}$. From this estimate it is straightforward to calculate $P_{\mathcal{O}}(\mathbf{y}_{(k)}|\hat{\mathbf{X}}_{i,(k)})$ using the standard relation for a multi-variable Gaussian probability density function:

$$P_{\mathcal{O}}(\sim) = \frac{1}{(2\pi)^{n/2}|\mathbf{S}_{i,(k)}|^{1/2}} e^{-(\mathbf{r}_{i,(k)}^T \mathbf{S}_{i,(k)}^{-1} \mathbf{r}_{i,(k)})/2}, \qquad (11)$$

where $\mathbf{r}_{i,(k)} = \mathbf{y}_{(k)} - \mathbf{C}_i\hat{\mathbf{x}}_{ci,(\bullet k)}$ denotes the measurement residual. The associated covariance matrix $\mathbf{S}_{i,(k)} = \mathbf{C}_i\mathbf{P}_{i,(\bullet k)}\mathbf{C}_i^T + \mathbf{R}$ accounts for the estimation error in $\hat{\mathbf{x}}_{i,(\bullet k)}$.

6 Tracking "Promising" 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. Hybrid belief update adds an additional computational burden to trajectory tracking, by introducing a Kalman filter for each trajectory that calculates equations (5-9) at each step. In order to ensure accurate numerical estimates, these filters must operate in the real-time loop, at the controller's sampling rate. This rate is in contrast to discrete mode estimation methods, such as Livingstone [Williams and Nayak, 1996], that can adequately operate at time scales of seconds.

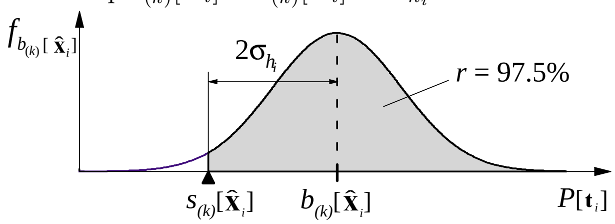
We address this problem with an any-time, any-space solution that dynamically adjusts the number of trajectories tracked in order to fit within the processor's computational and memory limits. [de Kleer and Williams, 1989] successfully introduced a focusing approach for model-based diagnosis that enumerates the modes of a system down to a threshold based on a mode's a priori probability. This approach is successful 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 focussed subset of the possible trajectories, based on what will fit within the available processor resources.

The key difference in our approach, however, is that we select those trajectories for which we know with *high confidence* that their probability will be high. This is different from selecting the trajectories with high a priori probability. The distinction is due to the fact that estimates are inherently uncertain, including probability estimates, and that this uncertainty greatly impacts our confidence in a probability estimate. However, an a priori probability specifies only an expected probability, it does not include the influence of this uncertainty.

7 Computing High Confidence Probabilities

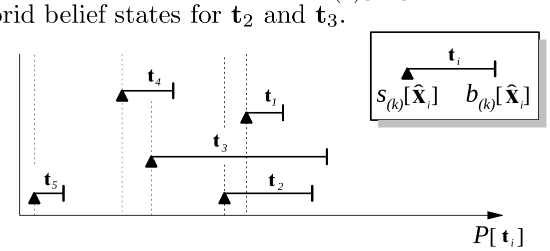
Rather than ranking based on the expected probability $P(\mathbf{t}_i) = h_{(k)}[\hat{\mathbf{X}}_i]$ of the tracked trajectory $\mathbf{t}_i = \langle \ldots, \hat{\mathbf{X}}_{i,(k)} \rangle$, our solution is to rank based on the greatest probability, $s_{(k)}[\hat{\mathbf{X}}_i]$, for which we have high confidence. First we introduce probability r, which quantifies our term "high confidence"; for example, we might select $r = 97.5\%$. We then define $s_{(k)}[\hat{\mathbf{X}}_i]$ as the highest probability for which $P(t_i) > s_{(k)}[\hat{\mathbf{X}}_i]$ with confidence r. In other words, we know from this relation that the probability of trajectory t_i is at least $s_{(k)}[\hat{\mathbf{X}}_i]$, with confidence r.

To compute $s_{(k)}[\hat{\mathbf{X}}_i]$ we need the probability density function (PDF) that describe the uncertainty in $P(\mathbf{t}_i)$. We follow the standard practice of approximating the PDF by a Gaussian distribution with mean $h_{(k)}[\hat{\mathbf{X}}_i]$ and approximated variance $\sigma^2_{h_i}$ [ISO, 1995]. Assuming the reasonable confidence $r = 97.5\%$ leads then to the simple relationship $s_{(k)}[\hat{\mathbf{X}}_i] = h_{(k)}[\hat{\mathbf{X}}_i] - 2\sigma_{h_i}$:



Ranking and Tracing Trajectories

To decide which trajectories to track, we rank trajectories $\mathbf{t}_i$ based on the high confidence probability, $s_{(k)}[\hat{\mathbf{X}}_i]$, rather than expected probability. The following figure illustrates such a ranking for trajectories terminating on 5 modes. The ranking $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5\}$ reflects the high confidence in $\mathbf{t}_1$, which is ranked highest, despite the fact that its hybrid belief state $h_{(k)}[\hat{\mathbf{X}}_i]$ is lower than the hybrid belief states for $\mathbf{t}_2$ and $\mathbf{t}_3$.



Our focused hybrid state estimation system then selects the first l trajectories to be tracked and updates the corresponding belief-states only. The value of l depends on the available computing resources. The sum $\varsigma = \sum_{i=1}^{l} s_{(k)}[\hat{\mathbf{X}}_i]$ indicates the portion of the probability space that is covered by our focused hybrid monitoring technique with 97.5% confidence. As the confidence in the estimates increases, for example due to a long observation horizon, it is possible that a small set of trajectories account for the majority of the probability space.

Computing Distributions of Mode Probability

To complete the development of our ranking strategy we need to compute the uncertainty in the belief state, described by its variance $\sigma^2_{h_i}$. Space limits preclude a detailed explanation, but we summarize the results for completeness. Variance is computed based on the functional relationship $h_{(k)}[\hat{\mathbf{X}}_i] = f_i(\cdots, \mathbf{y}_c, \cdots)$ between the uncertain measurements and the belief-state, as specified in the hybrid belief state update equations. In particular:

$$\begin{aligned} \sigma^2_{h_i} &= \sum_{j=1}^{m} \left(\frac{\partial f_i}{\partial y_{cj}} \right)^2 \sigma^2_{y_{cj}} \\ &\quad + 2 \sum_{j=1}^{m} \sum_{l=i+1}^{m} \frac{\partial f_i}{\partial y_{cj}} \frac{\partial f_i}{\partial y_{cl}} \sigma_{y_{cj}, y_{cl}}, \end{aligned} \tag{12}$$

$\{y_{c1}, \ldots, y_{cm}\}$ denotes the components of the measurement vector $\mathbf{y}_c$. The associated variances and covariances are $\sigma^2_{y_{cj}}$ and $\sigma^2_{y_{cj}, y_{cl}}$, respectively, and are drawn from the covariance matrix $\mathbf{S}_{i,(k)}$.

The sensitivities $\partial f_i / \partial y_{cj}$ are computed according to:

$$\frac{\partial f_i}{\partial y_{cj}} \propto \frac{\frac{\partial P_{\mathcal{O}i}}{\partial y_{cj}} \sum_j h_j P_{\mathcal{O}j} - P_{\mathcal{O}i} \sum_j h_j \frac{\partial P_{\mathcal{O}j}}{\partial y_{cj}}}{\left[\sum_j h_j P_{\mathcal{O}j} \right]^2}, \tag{13}$$

where $P_{\mathcal{O}i}$ and h_i are abbreviations of $P_{\mathcal{O}}(\mathbf{y}_{(k)} | \hat{\mathbf{X}}_{i,(k)})$ and $h_{(k)}[\hat{\mathbf{X}}_i]$ respectively. While this expression seems complex, we can calculate it efficiently by exploiting the intermediate results from the calculation of $P_{\mathcal{O}}$, according to:

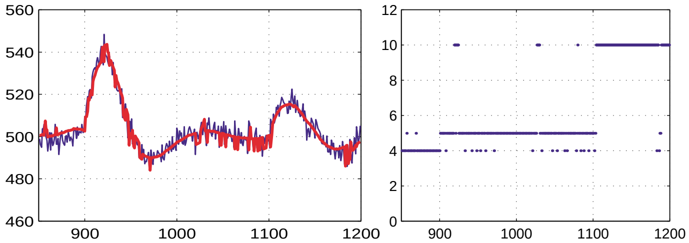
$$\left[\frac{\partial P_{\mathcal{O}i}}{\partial y_{c1}}, \ldots, \frac{\partial P_{\mathcal{O}i}}{\partial y_{cm}} \right] = P_{\mathcal{O}i} \mathbf{S}^{-1}_{i,(k)} \mathbf{r}_{i,(k)}. \tag{14}$$

8 Example continued

We demonstrate hybrid mode estimation on two operational conditions: (1) detection of crew entry into the PGC ($m_4 \to m_5$) and (2) a lighting failure (m_{10}) that reduces the light intensity in the PGC by 20%. The dynamic behavior of the CO_2 concentration (in ppm) at the modes m_4 and m_5 is governed by (1 time-step represents 1 minute):

$$\begin{aligned} m_4: &\quad x_{c,(k+1)} = x_{c,(k)} + 11.8373[f(x_{c,(k)}) + u_{c,(k)}] \\ m_5: &\quad x_{c,(k+1)} = x_{c,(k)} + 11.8373[f(x_{c,(k)}) + u_{c,(k)} + h_c] \\ &\quad f(x_{c,(k)}) = -1.4461 \cdot 10^{-2} \left[72.0 - 78.89 e^{-\frac{x_{c,(k)}}{400}} \right], \end{aligned}$$

where the term h_c accounts for the CO_2 exhaled by the crew members. The noisy measurement of the transient behavior of the controlled CO_2 concentration is given in the left graph of the figure below. 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 at $k = 903$. The light fault is then injected at time-step 1100 and diagnosed 9 time-steps later at at $k = 1109$. The right graph shows this discrimination among the modes m_4, m_5 and m_{10} extracted from the measurement (numbers on the vertical axis denote the mode numbers).



9 Implementation and Discussion

The implementation of our hybrid mode estimation scheme is written in Common LISP. The hybrid mode estimator uses a PHA description and performs hybrid mode 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.

In this paper we used Kalman filters, due to their simplicity and computational efficiency. However, our framework easily extends to other observer types that provide mean and variance for the state-variables. Other appropriate observers include extended Kalman filters and nonlinear filtering methods.

Our larger objective is to extend PHAs to handle concurrent automata, as well as discrete constraints, thus subsuming discrete, concurrent mode-estimation methods [Williams and Nayak, 1996]. This requires an extension of best first enumeration methods to high confidence probabilities.

Acknowledgments

This research is supported by NASA under contract NAG2-1388.

References

[Branicky, 1995] M. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control.* PhD thesis, MIT, 1995.

[de Kleer and Williams, 1989] J. de Kleer and B. Williams. Diagnosis with behavioral modes. In *Proc. of IJCAI-89.*

[Gelb, 1974] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[Henzinger, 1996] T. Henzinger. The theory of hybrid automata. In *Proc. of LICS-96*, pages 278–292, 1996.

[ISO, 1995] Guide to the expression of uncertainty in measurement. Internat. Organization for Standardization, 1995.

[Kurien and Nayak, 2000] J. Kurien and P. Nayak. Back to the future for consistency-based trajectory tracking. In *Proc. of AAAI-00*, 2000.

[McIlraith *et al.*, 1999] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta. Towards diagnosing hybrid systems. In *Proc. of DX-99*, 1999.

[Rinner and Kuipers, 1999] B. Rinner and B. Kuipers. Monitoring piecewise continuous behaviors by refining semi-quantitative trackers. In *Proc. of IJCAI-99*, 1999.

[Stengel, 1994] R. Stengel. *Optimal Control and Estimation.* Dover, 1994.

[Williams and Nayak, 1996] B. Williams and P. Nayak. A model-based approach to reactive self-configuring systems. In *Proc. of AAAI-96*, 1996.