Hybrid Diagnosis with Unknown Behavioral Modes

Michael W. Hofbaur[1] and **Brian C. Williams**[2]

Abstract. 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. This paper incorporates this novel capability of model-based diagnosis into a hybrid estimation scheme by calculating partial filters. The filters are based on causal and structural analysis of the specified components and their interconnection within the hybrid automaton model. Incorporating unknown modes provides a robust estimation scheme that can cope, unlike other hybrid estimation and multi-model estimation schemes, with unmodeled situations and partial information.

1 Introduction

Modern technology is increasingly leading to complex artifacts with high demands on performance and availability. As a consequence, fault-tolerant control and an underlying monitoring and diagnosis capability plays an important role in achieving these requirements. Monitoring and diagnosis systems that build upon the discrete model-based reasoning paradigm[8] can cope well with complexity in modern artifacts. As an example, the Livingstone system[22] successfully monitored and diagnosed the DS-1 space probe in flight, a system with approximately 4^{80} modes of operation. However, a widespread application of discrete model-based systems is hindered by their difficulty to reason about the continuous dynamics of an artifact in a comprehensive manner. Continuous behaviors are difficult to capture by the pure qualitative models that are used by the reasoning engines. Nevertheless, additional reasoning in terms of the continuous dynamics is vital for detecting functional failures, as well as low-level incipient (i.e slowly developing) faults and subtle component degradation.

Hybrid systems theory provides a modeling paradigm that integrates both, continuous state evolution and discrete mode changes in a comprehensive manner. Recent work in hybrid estimation[14, 16, 24, 9] attempts to overcome the shortcomings of discrete model-based diagnosis cited above and provides schemes that integrate model-based approaches with techniques from fault detection and isolation (FDI)[23, 4] and multi-model adaptive filtering[13, 11, 10]. The hybrid estimation schemes, as well as their FDI and multi-model filtering ancestors, work well whenever the underlying model(s) are 'close' mathematical descriptions of the physical artifact. They can fail severely whenever unforeseen situations occur. Therefore, it is essential to provide models that capture the entire spectrum of possible behaviors/modes whenever we use the hybrid estimate for closed loop control, for instance. Model-based diagnosis, in contrast, does not impose such a strong modeling assumption. Its concept of the *unknown mode* allows diagnosis of systems where no assumption is made about the behavior of one or several components of the system. In this way, it captures unspecified and unforeseen behaviors of the system under investigation. This paper provides an approach to incorporate the concept of an unknown mode into our hybrid estimation scheme[9]. As a result we obtain an estimation capability that can detect unforeseen situations. Furthermore, it allows us to continue estimation on a degraded basis. We achieve this by causal analysis[17, 20], structural analysis[7] and decomposition of the system.

This paper starts with a brief introduction to our hybrid systems modeling and estimation scheme. Upon this foundation, we extend hybrid estimation to incorporate the unknown mode and demonstrate the underlying structural analysis and decomposition task. Finally, an experimental evaluation with computer simulated data for a Martian live support system demonstrates the advantages of this extended hybrid estimation scheme.

2 Hybrid Systems

The hybrid automaton model used throughout this paper is based on [9] and can be seen as a model that merges hidden Markov models (HMM) with continuous discrete-time dynamical system models (we present the model on the level of detail sufficient for this work and refer the reader to the reference cited above for more detail).

2.1 Concurrent Hybrid Automata

Definition 1 A *discrete-time probabilistic hybrid automaton (PHA)* $\mathcal{A}$ is described as a tuple $\langle \mathbf{x}, \mathbf{w}, F, T, \mathcal{X}_d, T_s \rangle$:

- $\mathbf{x}$ denotes the hybrid *state variables* of the automaton[3], 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* $\mathbf{x}_c$ capture the dynamic evolution of the automaton. $\mathbf{x}$ 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 $\mathbf{u}_d$ (called *command variables*), continuous *input variables* $\mathbf{u}_c$, and continuous *output variables* $\mathbf{y}_c$.
- $F : \mathcal{X}_d \rightarrow F_{DE} \cup F_{AE}$ 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.

[1] Department of Automatic Control, Graz University of Technology, A-8010 Graz, Austria, email: hofbaur@irt.tu-graz.ac.at

[2] MIT Space Systems and AI Laboratories, 77 Massachusetts Ave., Rm. 37-381, Cambridge, MA 02139 USA, email: williams@mit.edu

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

- The finite set, T, of *transitions* specifies the probabilistic discrete evolution of the automaton.

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:

Definition 2 A *concurrent probabilistic hybrid automaton (cPHA)* $\mathcal{CA}$ is described as a tuple $\langle A, \mathbf{u}, \mathbf{y}_c, \mathbf{v}_s, \mathbf{v}_o, N_x, N_y \rangle$:

- $A = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_l\}$ 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 $x_{di}, \mathbf{x}_{ci}, \mathbf{u}_{di}, \mathbf{u}_{ci}, \mathbf{y}_{ci}, F_i, \mathcal{X}_{di}$).
- The *input variables* $\mathbf{u} = \mathbf{u}_d \cup \mathbf{u}_c$ of the automaton consists of the sets of discrete input variables $\mathbf{u}_d = \mathbf{u}_{d1} \cup \ldots \cup \mathbf{u}_{dl}$ (command variables) and continuous input variables $\mathbf{u}_c \subseteq \mathbf{u}_{c1} \cup \ldots \cup \mathbf{u}_{cl}$.
- The *output variables* $\mathbf{y}_c \subseteq \mathbf{y}_{c1} \cup \ldots \cup \mathbf{y}_{cl}$ specify the observed output variables of the cPHA.
- The observation process is subject to additive, zero mean Gaussian *sensor noise*. $N_y : \mathcal{X}_d \rightarrow \mathbb{R}^{m \times m}$ specifies the mode dependent[4] disturbance $\mathbf{v}_o$ in terms of the covariance matrix $\mathbf{R} = \text{diag}(r_i)$.
- N_x specifies additive, zero mean Gaussian *disturbances* that act upon the continuous state variables $\mathbf{x}_c = \mathbf{x}_{c1} \cup \ldots \cup \mathbf{x}_{cl}$. $N_x : \mathcal{X}_d \rightarrow \mathbb{R}^{n \times n}$ specifies the mode dependent disturbance $\mathbf{v}_s$ in terms of the covariance matrix $\mathbf{Q}$.

Definition 3 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 $\mathbf{x}_d = \{x_{d1}, \ldots, x_{dl}\}$ and the continuous state assignment $\mathbf{x}_{c,(k)}$ of the continuous state variables $\mathbf{x}_c = \mathbf{x}_{c1} \cup \ldots \cup \mathbf{x}_{cl}$.

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. 1 illustrates a simple example composed of 3 PHAs.

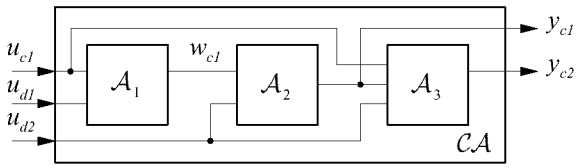


Figure 1. Example cPHA composed of three PHAs

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 additive, zero mean Gaussian disturbances $\mathbf{v}_s$, $\mathbf{v}_o$. The discrete-time evolution of $\mathbf{x}_c$ and $\mathbf{y}_c$ is described by the nonlinear system of difference equations (sampling period T_s)

$$\begin{aligned} \mathbf{x}_{c,(k)} &= \mathbf{f}_{(k)}(\mathbf{x}_{c,(k-1)}, \mathbf{u}_{c,(k-1)}) + \mathbf{v}_{s,(k-1)} \\ \mathbf{y}_{c,(k)} &= \mathbf{g}_{(k)}(\mathbf{x}_{c,(k)}, \mathbf{u}_{c,(k)}) + \mathbf{v}_{o,(k)}. \end{aligned} \tag{1}$$

The functions $\mathbf{f}_{(k)}$ and $\mathbf{g}_{(k)}$ are obtained by symbolically solving[5] the set of equations $F_1(x_{d1,(k)}) \cup \ldots \cup F_l(x_{dl,(k)})$ given the mode $\mathbf{x}_{d,(k)} = [x_{d1,(k)}, \ldots, x_{dl,(k)}]^T$.

[4] E.g. sensors can experience different magnitudes of disturbances for different modes.

[5] Our symbolic solver restricts the algebraic equations and nonlinear functions to ones that can be solved explicitly and utilizes a Gröbner Basis approach[3] to derive a set of equations of form (1).

Consider the illustrative cPHA in Fig. 1 with

$$\begin{aligned} \mathcal{A}_1 &= \langle \{x_{d1}\}, \{u_{d1}, u_{c1}, w_{c1}\}, F_1, T_1, \{m_{11}, m_{12}\} \ldots \rangle \\ \mathcal{A}_2 &= \langle \{x_{d2}, x_{c1}\}, \{u_{d2}, w_{c1}, y_{c1}\}, F_2, T_2, \{m_{21}, m_{22}\} \ldots \rangle \\ \mathcal{A}_3 &= \langle \{x_{d3}, x_{c2}, x_{c3}\}, \{u_{d2}, u_{c1}, y_{c1}, y_{c2}\}, F_3, T_3, \{m_{31}\} \ldots \rangle. \end{aligned}$$

F_1, F_2 and F_3 provide for a cPHA mode $\mathbf{x}_{d,(k)} = [m_{11}, m_{21}, m_{31}]^T$ the equations

$$\begin{aligned} F_1(m_{11}) &= \{u_{c1} = 5.0\, w_{c1}\} \\ F_2(m_{21}) &= \{x_{c1,(k)} = 0.8\, x_{c1,(k-1)} + w_{c1,(k-1)}, \\ &\qquad y_{c1} = x_{c1}\} \\ F_3(m_{31}) &= \{x_{c2,(k)} = x_{c3,(k-1)} + y_{c1,(k-1)}, \\ &\qquad x_{c3,(k)} = 0.4\, x_{c2,(k-1)} + 0.5\, u_{c1,(k-1)}, \\ &\qquad y_{c2} = 2.0\, x_{c2} + x_{c3}\}. \end{aligned} \tag{2}$$

This leads to the discrete-time model:

$$\begin{aligned} x_{c1,(k)} &= 0.8\, x_{c1,(k-1)} + 0.2\, u_{c1,(k-1)} + v_{s1,(k-1)} \\ x_{c2,(k)} &= x_{c1,(k-1)} + x_{c3,(k-1)} + v_{s2,(k-1)} \\ x_{c3,(k)} &= 0.4\, x_{c2,(k-1)} + 0.5\, u_{c1,(k-1)} + v_{s3,(k-1)} \\ y_{c1,(k)} &= x_{c1,(k)} + v_{o1,(k)} \\ y_{c2,(k)} &= 2.0\, x_{c2,(k)} + x_{c3,(k)} + v_{o2,(k)} \end{aligned} \tag{3}$$

2.2 Estimation of Hybrid Systems

To detect the onset of subtle failures, it is essential that a monitoring and diagnosis system is 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 $\{\mathbf{y}_{c,(0)}, \mathbf{y}_{c,(1)}, \ldots, \mathbf{y}_{c,(k)}\}$ and control inputs $\{\mathbf{u}_{(0)}, \mathbf{u}_{(1)}, \ldots, \mathbf{u}_{(k)}\}$, 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* $h_{(k)}[\hat{\mathbf{x}}]$.

We perform hybrid estimation as extended version of HMM-style belief-state update that accounts for the influence of the continuous dynamics upon the system's discrete modes. 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:

$$h_{(\bullet k)}[\hat{\mathbf{x}}_i] = P_{\mathcal{T}}(\mathbf{m}_i | \hat{\mathbf{x}}_{j,(k-1)}, \mathbf{u}_{d,(k-1)}) h_{(k-1)}[\hat{\mathbf{x}}_j] \tag{4}$$

$$h_{(k)}[\hat{\mathbf{x}}_i] = \frac{h_{(\bullet k)}[\hat{\mathbf{x}}_i] P_{\mathcal{O}}(\mathbf{y}_{c,(k)} | \hat{\mathbf{x}}_{i,(k)}, \mathbf{u}_{c,(k)})}{\sum_j h_{(\bullet k)}[\hat{\mathbf{x}}_j] P_{\mathcal{O}}(\mathbf{y}_{c,(k)} | \hat{\mathbf{x}}_{j,(k)}, \mathbf{u}_{c,(k)})} \tag{5}$$

$h_{(\bullet k)}[\hat{\mathbf{x}}_i]$ 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 successor states to be tracked. Consecutive 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 estimate $\hat{\mathbf{x}}_{j,(k)}$ with the highest belief-state $h_{(k)}[\hat{\mathbf{x}}_j] = \max_i(h_{(k)}[\hat{\mathbf{x}}_i])$ is taken as the hybrid estimate at time-step k.

Tracking all possible trajectories of the system is almost always intractable because the number of trajectories becomes too large after only a few time-steps. In [9] we present an approximative anytime anyspace algorithm that copes with the exponential growth, as well as the large number of modes in a typical concurrent hybrid automaton model.

Hybrid estimation and other multi-model estimation schemes have in common that they require models that are 'close' mathematical descriptions of the system. They can fail severely whenever unforeseen, i.e. unmodeled, situations occur. As a consequence, we have to provide models for all operational modes as well as an exhaustive set of models for possible failure modes. Providing all possible failure models can be problematic even under the assumption of an exhaustive failure mode effect analysis (FMEA). For instance, consider an incipient fault in a servo valve that causes the valve to drift off its nominal opening value. The drift (positive, negative, slow, fast...) is subject to the fault. It is surely difficult to provide a mathematical model with the correct parameter values that captures all possible drift situations. Nor is it helpful to introduce a sufficiently large set of modes that captures possible situations of the drift fault as this would introduce additional complexity for hybrid estimation by increasing the number of modes unnecessarily.

This requirement of hybrid mode estimation is in contrast to discrete model-based diagnosis schemes, such as GDE (e.g. [5, 6, 19]). Model-based diagnosis deduces the possible mode of the system based on nominal models, and few specified fault models only. The onset of possible fault scenarios are covered by the so called *unknown mode* which does not impose any constraints on the system's variables.

The next section provides an approach that systematically incorporates the concept of the unknown mode into our hybrid estimation scheme.

3 Estimation with Unknown Modes

The estimation scheme [9] requires a fully specified mode assignment $\mathbf{x}_{di,(k)}$ for each candidate trajectory that is tracked in the course of hybrid estimation. Only a fully specified mode allows us to deduce the mathematical model (1) for the overall system. This model is the basis for the dynamic filter (e.g. extended Kalman filter) that is used in the course of hybrid estimation.

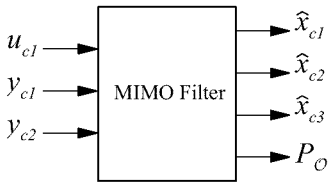


Figure 2. MIMO filter (e.g. extended Kalman filter) for the cPHA example

For our illustrative 3 component example introduced above this would mean that hybrid estimation calculates a multi-input multi-output (MIMO) filter (see Fig. 2) for mode $\mathbf{x}_{di,(k)} = [m_{11}, m_{21}, m_{31}]^T$ based on the mathematical model (3). This filter provides the hybrid state estimate $\hat{\mathbf{x}}_{i,(k)}$ as well as the value for the hybrid probabilistic observation function $P_{\mathcal{O}}(\mathbf{y}_{c,(k)}|\hat{\mathbf{x}}_{i,(k)}, \mathbf{u}_{c,(k)})$ for the hybrid estimator (see Appendix A for the extended Kalman filter estimation details).

Let us assume the mode $\mathbf{x}_{di,(k)} = [?, m_{21}, m_{31}]^T$ which specifies that component 1 ($\mathcal{A}_1$) is in *unknown mode*. A component in unknown mode imposes no constraints (equations) among its variables (u_{c1} and the internal variable w_{c1}, in our case). As a consequence, we cannot deduce an overall mathematical model of the form (1) and fail to provide the basis for the hybrid estimation scheme, the MIMO filter for mode $\mathbf{x}_{di,(k)} = [?, m_{21}, m_{31}]^T$.

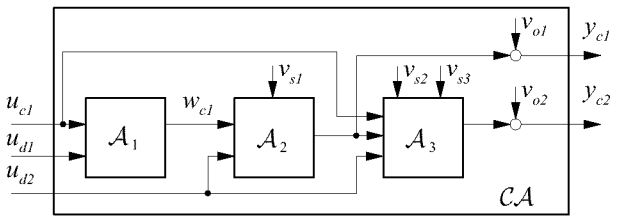


Figure 3. Example cPHA with explicit noise inputs

However, a close look on the PHA interconnection (Fig. 3 - the figure extends Fig. 1 by including the implicit noise inputs, as well as indicating the causality for the internal I/O variables) reveals that we can still estimate component 3 by its observed output y_{c2} and the observation y_{c1} as a substitute for the value of its input. This intuitive approach utilizes a decomposition of the cPHA as shown in Fig. 4.

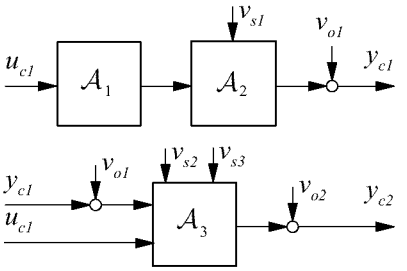


Figure 4. Decomposed cPHA

The decomposition allows us to treat the concurrent parts of the system independently and calculate a *filter cluster* consisting of 2 independent filters. However, when calculating the individual filters for the cluster, we have to take into account that we use the *measurement* of the input to the third component (y_{c1}) in replacement to its true value. This can be interpreted as having additional additive noise at the component's input as indicated in Fig. 4. The following modification of the covariance matrix $\mathbf{Q}_3$ for the state variables of $\mathcal{A}_3$ takes this into account:

$$\tilde{\mathbf{Q}}_3 = \mathbf{b}_3 r_1 \mathbf{b}_3^T + \mathbf{Q}_3, \tag{6}$$

where r_1 denotes the variance of disturbance v_{o1} and $\mathbf{b}_3 = [0,\ 1]^T$

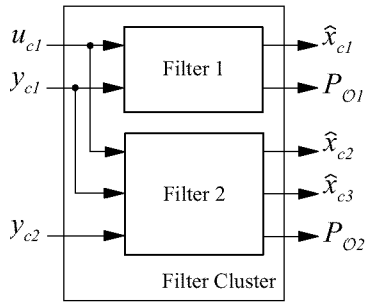


Figure 5. Decomposed filter

denotes the input vector[6] of $\mathcal{A}_3$ with respect to y_{c1}.

A filter cluster consisting of extended Kalman filters and the MIMO extended Kalman filter are interchangeable as they provide the same expected value for the continuous state ($E(\hat{\mathbf{x}}_c)$) whenever the mode of the automaton is fully specified. However, the decomposed filter has the advantage that the probabilistic observation function $P_{\mathcal{O}}$ of the overall system is given by

$$P_{\mathcal{O}} = \prod_j P_{\mathcal{O}j}, \tag{7}$$

where $P_{\mathcal{O}j}$ denotes the probabilistic observation function of the j'th filter in the filter cluster.

This factorization of the probabilistic observation function allows us to calculate an upper bound for $P_{\mathcal{O}}$ whenever one or more components of the system are in unknown mode. We simply take the product over the remaining filters in the cluster. This is equivalent with considering the upper bounds of the inequalities $P_{\mathcal{O}j} \leq 1$ for each unknown filter j. In our example with unknown component $\mathcal{A}_1$ this would mean:

$$P_{\mathcal{O}} \leq P_{\mathcal{O}2},$$

where $P_{\mathcal{O}2}$ denotes the observation function for the filter that estimates the continuous state of component $\mathcal{A}_3$.

The following subsection provides a graph-based approach for filer cluster deduction that grounds the informally introduced decomposition on a more versatile basis.

3.1 System Decomposition and Filter Cluster Calculation

Starting point for the decomposition of the system for a cPHA mode $\mathbf{x}_d$ is the set of equations

$$F_1(x_{d1,(k)}) \cup \ldots \cup F_l(x_{dl,(k)}) =: \mathcal{F}(\mathbf{x}_d), \tag{8}$$

where $F_j(x_{dj,(k)})$ returns the appropriate set of equations for a component $\mathcal{A}_i$ whenever $x_{dj,(k)} \in \mathcal{X}_{dj}$ or the empty set whenever the component is in unknown mode, i.e. $x_{dj,(k)} =?$. Although we still have to solve the set of equations to arrive at the mathematical model of form (1) we can interpret the set of equations (8) as the *raw model* for the system given mode $\mathbf{x}_d$. The following decomposition performs a structural analysis of the raw model-based on causal analysis[17, 20], structural observability analysis[7] and graph decomposition[1].

A cPHA model does not impose a fixed causal structure that specifies directionality of automaton interconnections. Causality is implicitly specified by the set of equations. This increases the expressiveness of the modeling framework but requires us to perform a causal analysis of the raw model (8) as a first step. The deduction of the causal dependencies is done by applying the bipartite-matching based algorithm presented in [17]. The resulting directed graph records the causal dependencies among the variables of the system (Fig. 6 shows the graph for the the illustrative 3 PHA example). Each vertex of the graph represents one equation $e_i \in \mathcal{F}$

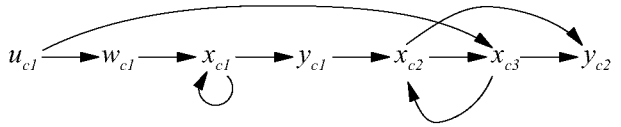


Figure 6. Causal graph for the cPHA example

or an exogenous variable specification (e.g. u_{c1}) and is labeled by its dependent variable which also specifies the outgoing edge (in the following, we will use the variable name to refer to the corresponding vertex in the graph). Vertices without incoming edges specify the *exogenous* variables.

Definition 4 A *causal graph* of a cPHA $\mathcal{CA}$ at a mode $\mathbf{x}_d$ is a directed graph that records the causal dependencies among the variables $v \in \bigcup_i \mathbf{x}_{ci} \cup \mathbf{u}_{ci} \cup \mathbf{y}_{ci}$ of $\mathcal{CA}$. We denote the causal graph by $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_d)$ and sometimes omit arguments where no confusion seems likely.

Goal of our analysis is to obtain a set of independent subsystems that utilize observed variables as virtual inputs. Therefore, we slice the graph at observed variable vertices with outgoing edges, insert a new vertex to represent a virtual input and re-map the sliced outgoing edges to this vertex. Fig. 7 demonstrates this re-mapping for the causal graph of Fig. 6. The observed variables are y_{c1} and y_{c2}. Only the vertex with dependent variable y_{c1} has an outgoing edge, thus we slice the graph at $y_{c1} \rightarrow x_{c2}$ and re-map the edge to the virtual input uy_{c1}.

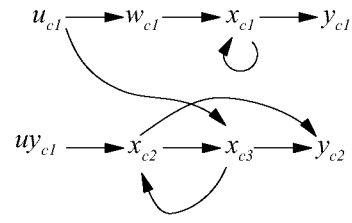


Figure 7. Remapped causal graph for the cPHA example

A dynamic filter (e.g. extended Kalman filter) can only estimate the observable part of the model. Therefore, it is essential to perform

[6] In the general case, we have to calculate $\mathbf{b}_j$ for a cPHA component $\mathcal{A}_j$ and observed inputs $\mathbf{uy}_c$ by linearization, more specifically: $\mathbf{b}_{j,(k)} = \partial \mathbf{f}_j / \partial \mathbf{uy}_c|_{\hat{\mathbf{x}}_{cj,(k-1)}, \mathbf{u}_{cj,(k-1)}}$, where $\mathbf{f}_j$ denotes the right-hand side of the difference equation for component $\mathcal{A}_j$, $\mathbf{uy}_c$ refers to the observed variables that are used as inputs to the component (i.e. $\mathbf{uy}_c \subset \mathbf{y}_c$) and $\hat{\mathbf{x}}_{cj,(k-1)}$ as well as $\mathbf{u}_{cj,(k-1)}$ represent the state estimate and the continuous input for component $\mathcal{A}_j$ at the previous time-step, respectively.

an observability analysis prior calculating the filter so that non observable parts of the model are excluded. We perform this analysis on a structural basis[7].

Definition 5 We call a variable v of a cPHA $\mathcal{CA}$ at mode $\mathbf{x}_d$ *structurally observable (SO)* whenever it is directly observed, i.e. $v \in \mathbf{y}_c$, or there exists at least one path in the causal graph $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_d)$ that connects the variable z to an output variable $y_c \in \mathbf{y}_c$ of $\mathcal{CA}$.

A filter estimates the state variables $\mathbf{x}_c$ of a dynamic system based on observations $\mathbf{y}_c$ and the inputs $\mathbf{u}_c$ that act upon the state variables $\mathbf{x}_c$. The required knowledge about the inputs $\mathbf{u}_c$ indicates that the structural observability criteria is not yet sufficient to determine the submodel for estimation. We have to make sure, that no unknown exogenous input influences a variable. To illustrate this, consider again the 3 PHA example with mode $\mathbf{x}_d = [?, m_{21}, m_{31}]^T$. Component 1 in unknown mode omits the equation that relates the variables u_{c1} and w_{c1}. This leads to a causal graph $\tilde{\mathcal{CG}}$ (Fig. 8), where w_{c1} is labeled as exogenous (no incoming edges). This unknown exogenous input influences the state variable x_{c1} and, as a consequence, prevents us from estimating it!

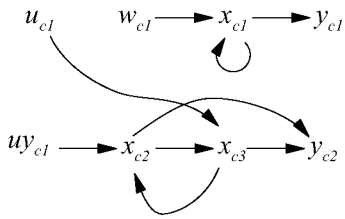


Figure 8. Remapped causal graph for the cPHA example with unknown component $\mathcal{A}_1$

We extend our structural analysis of the causal graph by the following criteria:

Definition 6 We call a variable v of a cPHA $\mathcal{CA}$ at mode $\mathbf{x}_d$ *structurally determined (SD)* whenever it is an input variable of the automaton, i.e. $v \in \mathbf{u}_c$, or there does not exist a path in the causal graph $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_d)$ that connects an exogenous variable $u_e \notin \mathbf{u}_c$ with v.

Furthermore, it is helpful to eliminate loops in the causal graph prior checking variables against both structural criteria. For this purpose, we calculate the *strongly connected components* of the causal graph[1].

Definition 7 A *strongly connected component (SCC)* of the causal graph $\mathcal{CG}$ is a maximal set $\mathcal{SCC}$ of variables in which there is a path from any one variable in the set to another variable in the set.

Fig. 9 shows the remapped causal graph for the 3 PHA example after grouping variables into strongly connected components.

The strong interconnection among variables in an SCC implies that:

1. Structural observability of variables in an SCC follows directly from structural observability of at least one variable in the SCC.

[7] Throughout the paper we assume that loss of observability is caused by a structural defect of the model. Otherwise, it is necessary to perform an additional numerical observability test [18] as structural observability only provides a *necessary* condition for observability.

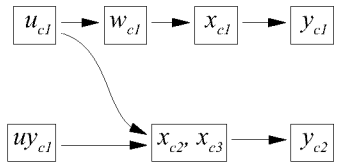


Figure 9. Causal SCC graph for cPHA example

2. A variable in an SCC is structurally determined, if and only if all variables in the SCC are structurally determined.

As a consequence, we can apply our structural analysis to strongly connected components directly and operate on the SCC graph, i.e a causal graph without loops. The analysis of a strongly connected component with respect to structural observability and structural determination (SOD) can be outlined as follows:

---

function determine-SOD-of-SCC($\mathcal{SCC}$, $\mathbf{u}_c$, k)
  when SOD-undetermined?($\mathcal{SCC}$)
    if exogenous?($\mathcal{SCC}$)
      then $v_i \leftarrow$ independent-var($\mathcal{SCC}$)
        if $v_i \in \mathbf{u}_c$ **then** SD($\mathcal{SCC}$) $\leftarrow$ **True**
          else SD($\mathcal{SCC}$) $\leftarrow$ **False**
      else $\mathcal{V} \leftarrow$ uplink-SCCs($\mathcal{SCC}$)
        loop for $\mathcal{SCC}_i$ in $\mathcal{V}$
          do determine-SOD-of-SCC($\mathcal{SCC}_i$, $\mathbf{u_c}$, k)
        SO($\mathcal{SCC}$) $\leftarrow$ **True**
        SD($\mathcal{SCC}$) $\leftarrow$ all-uplink-SCCs-are-SD?($\mathcal{V}$)
        cluster-index($\mathcal{SCC}$) $\leftarrow k \cup$ cluster-indices($\mathcal{V}$)
    SOD-determined($\mathcal{SCC}$) $\leftarrow$ **True**
  return Nil

Our structural analysis algorithm determines structural observability and determination (SOD) of a variable by traversing the SCC graph backwards from the observed variables towards the inputs. In the course of this analysis we label non-exogenous strongly connected components with an index that refers to their cluster membership. This indexing scheme allows us to cluster the variables into non-overlapping clusters with respect to the observed variables. The direct relation between a variable, its determining equation, and the cPHA component that specified this equation leads to the component clusters sought. The structural analysis can be summarized as follows:

---

function component-clustering($\mathcal{CA}$, $\mathbf{x}_d$)
  returns a set of cPHA component clusters
  $\mathbf{y}_c \leftarrow$ observed-vars($\mathcal{CA}$)
  $\tilde{\mathcal{CG}} \leftarrow$ remap-causal-graph($\mathcal{CG}(\mathcal{CA}, \mathbf{x}_d)$, $\mathbf{y}_c$)
  $\mathbf{u}_c \leftarrow$ virtual-inputs($\tilde{\mathcal{CG}}$) $\cup$ input-vars($\mathcal{CA}$)
  $\mathcal{CG}_{SCC} \leftarrow$ strongly-connected-component-graph($\tilde{\mathcal{CG}}$)
  $k \leftarrow 0$
  loop for $\mathcal{SCC}_i$ in output-SCCs($\mathcal{CG}_{SCC}$, $\mathbf{y}_c$)
    do determine-SOD-of-SCC($\mathcal{SCC}_i$, $\mathbf{u}_c$, k)
      $k \leftarrow k + 1$
  graph-clusters $\leftarrow$ get-SOD-SSC-clusters($\mathcal{CG}_{SCC}$)
  return automaton-clusters($\mathcal{CA}$, *graph-clusters*)

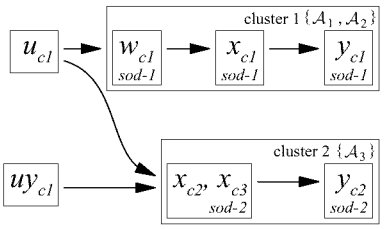


Figure 10. Labeled and partitioned causal SCC graph for the 3 cPHA example

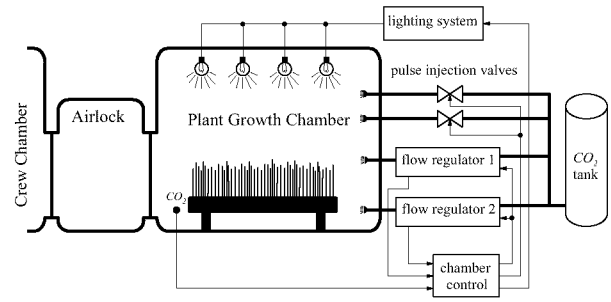


Figure 11. BIO-Plex plant growth chamber

Each component cluster defines the observable and determined raw model for a subsystem of the cPHA. This raw model can be solved symbolically and provides the nonlinear system of difference equations (a model similar to (1), but with the additional virtual inputs) that is the basis for the corresponding filter in the filter cluster. In this way we exclude the unobservable and/or undetermined parts of the overall system from estimation.

Whenever a state variable x_{cj} becomes unobservable and/or undetermined (e.g. due to a mode change) during hybrid estimation, we hold the value for the mean at its last known estimate $\hat{x}_{cj}$ and increase its variance $\sigma_j^2 = p_{jj}$ by a constant factor at each hybrid estimation step. This reflects a continuously decreasing confidence in the estimate $\hat{x}_{cj}$ and allows us to restart estimation whenever the variable becomes observable and determined again[8].

4 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 Martian 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. 11.

The system is composed of several components, such as redundant flow regulators (FR1, FR2) that provide continuous CO_2 supply, redundant pulse injection valves (PIV1, PIV2) that provide a means for increasing the CO_2 concentration rapidly, a lighting system (LS) and the plant growth chamber (PGC), 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).

Hybrid estimation schemes are key to tracking system operational modes, as well as, detecting subtle failures and performing diagnoses. For example, we simulate a failure of the second flow regulator. The regulator becomes off-line and drifts slowly towards its positive limit. This fault situation is difficult to capture by an explicit fault model as we do not know, in advance, whether the regulator drifts towards its postitive or negative limit, nor do we know the magnitude of the drift. A fault of this type, which develops slowly and whose symptom is hidden among the noise in the system is a typical candidate for our unknown-mode detection capability. However, we also provide explicit failure models that describe typical situations. For example, the PGC has 4 plant trays with one illumination bank for each tray. A black out of one illumination bank can be interpreted as a 25% loss in light intensity. This situation can be modeled explicitly by a dynamical model that takes this reduced light intensity into account.

In the following we describe the outcome of a simulated experiment where the flow regulator fault with drifting symptom is injected at time point $k = 700$ and an additional light fault, that harms one of the four illumination banks, is injected at $k = 900$. The faults are 'repaired' at $k = 1100$ and $k = 1300$ for the flow regulator fault and the lighting fault, respectively. This experiment illustrates unknown mode detection and recovery from it, nominal failure mode detection, and the multiple fault detection capability of our approach.

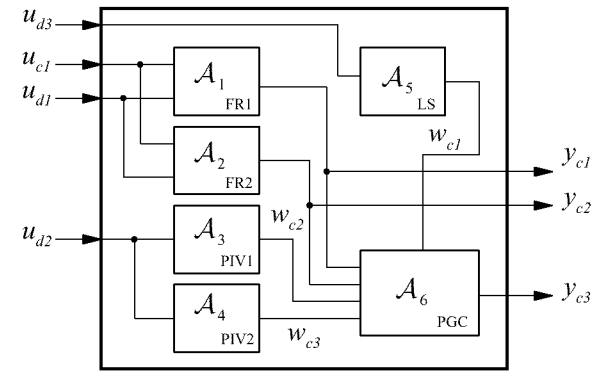


Figure 12. BIO-Plex cPHA model

The simulated data is gathered from the execution of a refined subset of NASA's JSC's CONFIG model for the BIO-Plex system[12]. Hybrid estimation utilizes a cPHA model that consists of 6 components as shown in Fig. 12. To illustrate the complexity of the hybrid estimation problem we should note, that the concurrent automaton has approximately $5^6 \approx 15000$ modes. Each mode describes the dynamic evolution of the chamber system by a third order system of difference equations. For example, the nominal operational condition for plant growth is characterized by the mode

[8] Whenever a state variable x_{cj} is directly observed we also can utilize an alternative approach suggested in [15] that restarts the estimator with the observed value, thus improving the observer convergence time.

$\mathbf{x}_d = [m_{r2}, m_{r2}, m_{v1}, m_{v1}, m_{l2}, m_{p2}]$, where m_{r2} characterizes an partially open flow regulator, m_{v1} a closed pulse injection valve, m_{l2} 100% light on, and m_{p2} plant growth mode at 1200 ppm, respectively. This mode specifies the raw model:

$$\begin{aligned}
F_1(m_{r2}) &= \{x_{c1,(k)} = 0.5\, u_{c1,(k-1)},\ y_{c1} = x_{c1}\} \\
F_2(m_{r2}) &= \{x_{c2,(k)} = 0.5\, u_{c1,(k-1)},\ y_{c2} = x_{c2}\} \\
F_3(m_{v1}) &= \{w_{c2} = 0.0\} \\
F_4(m_{v1}) &= \{w_{c3} = 0.0\} \\
F_5(m_{l2}) &= \{w_{c1} = 1204.0\} \\
F_6(m_{p2}) &= \{x_{c3,(k)} = x_{c3,(k-1)} + 20.163\cdot \\
&\quad [-1.516 \cdot 10^{-4} f_1(w_{c1,(k-1)}) f_2(x_{c3,(k-1)}) + \\
&\quad y_{c1,(k-1)} + y_{c2,(k-1)} + w_{c1,(k-1)} + w_{c2,(k-1)}], \\
&\quad y_{c3} = x_{c3}\},
\end{aligned} \tag{9}$$

where f_1 and f_2 denotes

$$\begin{aligned}
f_1(w_{c1}) &:= -7.615 + 0.111\, w_{c1} - 2.149 \cdot 10^{-5}\, w_{c1}^2 \\
f_2(x_{c3}) &:= 72.0 - 78.89\, e^{-x_{c3}/400.0}.
\end{aligned} \tag{10}$$

$x_{c1,(k)}$ and $x_{c2,(k)}$ denote the gas flow ([g/min]) of flow regulator 1 and 2, respectively and $x_{c3,(k)}$ denotes the CO_2 gas concentration ([ppm]) in the plant growth chamber. $w_{c1,(k)}$ and $w_{c2,(k)}$ denote the gas flow ([g/min]) of the pulse injection valves and $w_{c3,(k)}$ denotes the photosynthetic photon flux ([μ-mol/m^{2}s]) of the lights above the plant trays. The nonlinear expression

$$-1.516 \cdot 10^{-4} f_1(w_{c1,(k-1)}) f_2(x_{c3,(k-1)})$$

approximates the CO_2 gas production [g/min] due to photosynthesis according to the CO_2 gas concentration and chamber illumination[12]. This raw model defines a third order system of discrete-time difference equations with sampling period $Ts = 1$ [min]:

$$\begin{aligned}
x_{c1,(k)} &= 0.5\, u_{c1,(k-1)} + v_{s1,(k-1)} \\
x_{c2,(k)} &= 0.5\, u_{c1,(k-1)} + v_{s2,(k-1)} \\
x_{c3,(k)} &= x_{c3,(k-1)} + 20.163[-1.041 + \\
&\quad 1.141 e^{-x_{c3,(k)}/400.0} + x_{c1,(k-1)} + x_{c2,(k-1)}] + v_{s3,(k-1)} \\
y_{c1,(k)} &= x_{c1,(k)} + v_{o1,(k)} \\
y_{c2,(k)} &= x_{c2,(k)} + v_{o2,(k)} \\
y_{c2,(k)} &= x_{c3,(k)} + v_{o3,(k)},
\end{aligned} \tag{11}$$

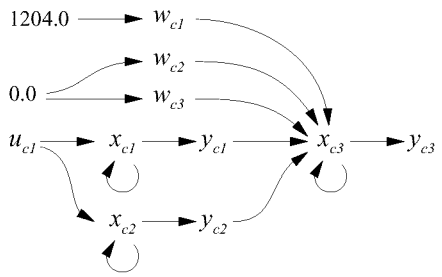


Figure 13. Causal graph of the BIO-Plex cPHA raw model (9)

The causal graph (Fig. 13) of the raw model (9) leads to the decomposition of the system as shown in Fig. 14 (our implementation of the causal analysis and decomposition algorithms treats constant values, such as the value 1204.0 for the photosynthetic photon flux, as known exogenous inputs with constant value). The decomposition of the model leads to a filter cluster with 3 extended Kalman filters - one for each flow regulator and one for the remaining system (pulse injection valves, lighting system and plant growth chamber). This enables us to estimate the mode and continuous state of the flow regulators independent of the remaining system. As a consequence, an unknown mode in a flow regulator does not cause any implications on the estimation of the remaining system.

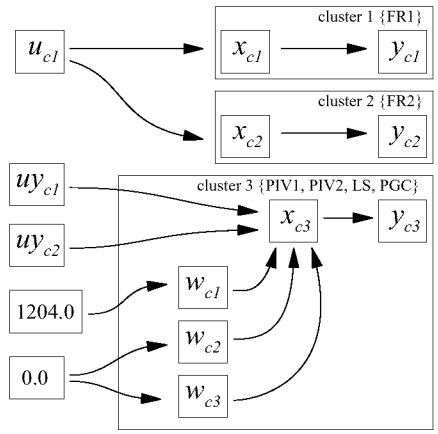


Figure 14. Partitioned causal SCC graph of the BIO-Plex cPHA model

Fig. 15 shows the continuous input (control signal) u_{c1}, observed flow rates for flow regulator 1 and 2 and the CO_2 concentration for the experiment. Both flow regulators provide half of the requested gas injection rate up to $k = 700$. At this time point, the second flow regulator starts to slowly drift towards its positive limit which it will reach at approximately $k = 800$. The camber control system reacts immediately and lowers the control signal in order to keep the CO_2 concentration at the requested 1200 ppm concentration. This transient behavior causes a slight bump in the CO_2 concentration as shown in Fig. 15-b. Our hybrid mode estimation system detects this unmodeled fault at $k = 727$ and declares flow regulator 2 to be in an unknown mode (we indicate the unknown mode by the mode number 0 in Fig. 16). The flow regulator mode *stuck-open* (m_{r5}) be-

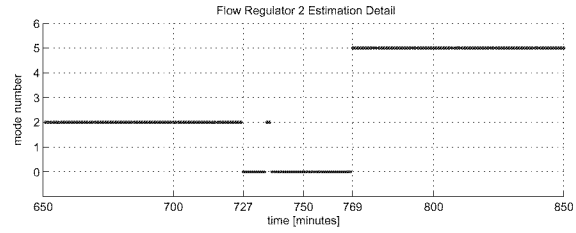
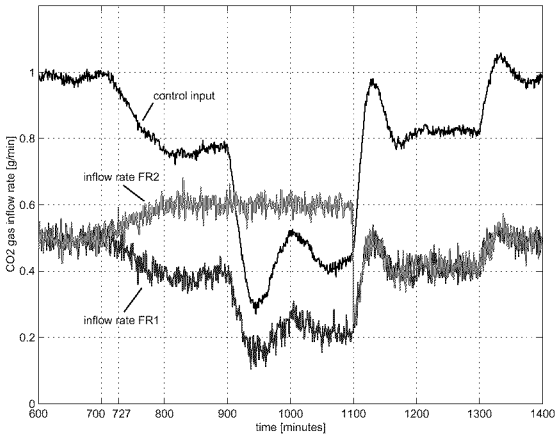
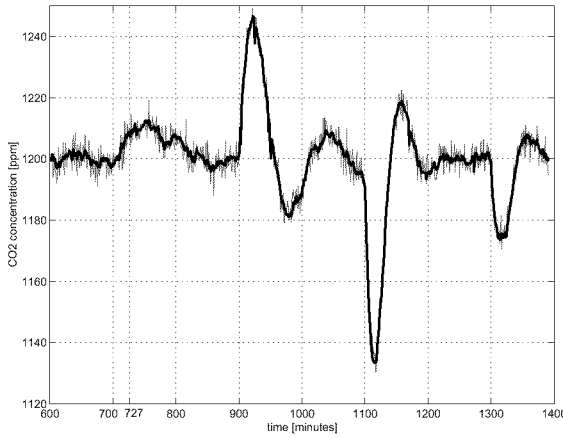


Figure 16. Mode estimate detail for flow regulator 2

comes more and more likely as the regulator drifts towards its open position. Hybrid mode estimation prefers this mode as symptom ex-



(a) Control input u_c and measured CO_2 input flow rates



(b) CO_2 level in PGC (measurement - gray/green, estimate - black)

Figure 15. Observed data and continuous estimation of the CO_2 concentration in plant growth chamber

planation from $k = 769$ onwards, although flow regulator 2 goes into saturation a little bit later at $k = 800$.

The light fault at $k = 900$ is detected almost instantly at $k = 904$ (m_{l4}). This good discrimination among the pre-specified modes (failure and nominal) is further demonstrated at the termination points of the faults. Repairs of the flow regulator 2 and the lighting system are detected immediately at $k = 1101$ and $k = 1301$, respectively. Fig. 17 shows the mode estimation result for the lighting system and flow regulator 2 over the entire experiment horizon.

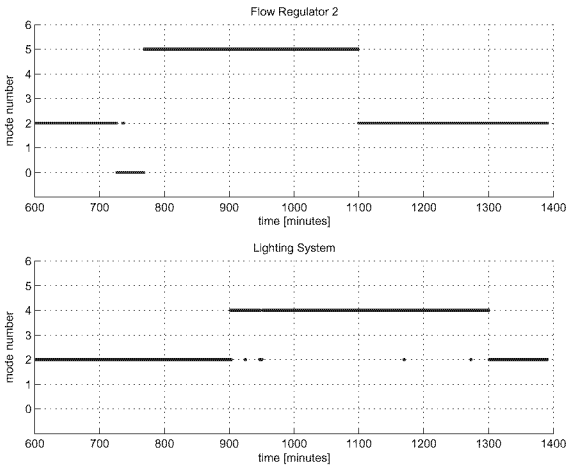


Figure 17. Mode estimates for flow regulator 2 and lighting system

5 Implementation and Discussion

The implementation of our hybrid estimation scheme extends previous work on hybrid estimation [9] and is written in Common LISP. The hybrid estimator uses a cPHA description and performs decomposition and estimation, as outlined above. Decomposition is done on-line according to the mode hypotheses that are tested in the course of hybrid estimation. In general, it can be assumed that the the mode in the system evolves on a lower rate than the hybrid estimation rate, which operates on the sampling period T_s. Therefore, we cache recent decompositions and their corresponding filters for re-use as a compromise between a-priori calculation (space complexity) and pure on-line deduction (time complexity).

Optimized model-based estimation schemes, such as Livingstone[22], utilize *conflicts* to focus the underlying search operation. A conflict is a (partial) mode assignment that makes a hypothesis very unlikely. This requires a more general treatment of unknown modes compared to the filter decomposition task introduced above. The decompositional model-based learning system Moriarty[21] introduced continuous variants of conflicts, so-called *dissents*. We are currently reformulating these dissents for hybrid systems and investigate their incorporation to improve the underlying search scheme. This will lead to an overall framework that unifies our previous work on Livingstone, Moriarty and hybrid estimation.

REFERENCES

[1] A. Aho, J. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983.

[2] B. Anderson and J. Moore, *Optimal Filtering*, Information and System Sciences Series, Prentice Hall, 1979.

[3] *Gröbner Bases and Applications*, eds., B. Buchberger and F. Winkler, Cambridge Univ. Press, 1998.

[4] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer, 1999.

[5] J. de Kleer and B. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, **32**(1), 97–130, (1987).

[6] J. de Kleer and B. Williams, 'Diagnosis with behavioral modes', in *Proceedings of IJCAI-89*, pp. 1324–1330, (1989).

[7] A. Gehin, M. Assas, and M. Staroswiecki, 'Structural analysis of sys-

tem reconfigurability', in *Preprints of the 4th IFAC SAFEPROCESS Symposium*, volume 1, pp. 292–297, (2000).
[8] *Readings in Model-Based Diagnosis*, eds., W. Hamscher, L. Console, and J. de Kleer, Morgan Kaufmann, San Mateo, CA, 1992.
[9] M. Hofbaur and B.C. Williams, 'Mode estimation of probabilistic hybrid systems', in *Hybrid Systems: Computation and Control, HSCC 2002*, eds., C.J. Tomlin and M.R. Greenstreet, volume 2289 of *Lecture Notes in Computer Science*, 253–266, Springer Verlag, (2002).
[10] P. Li and V. Kadirkamanathan, 'Particle filtering based likelyhood ratio approach to fault diagnosis in nonlinear stochastic systems', *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, **31**(3), 337–343, (2001).
[11] X.R. Li and Y. Bar-Shalom, 'Multiple-model estimation with variable structure', *IEEE Transactions on Automatic Control*, **41**, 478–493, (1996).
[12] J. T. Malin, L. Fleming, and T. R. Hatfield, 'Interactive simulation-based testing of product gas transfer integrated monitoring and control software for the lunar mars life support phase III test', in *SAE 28th International Conference on Environmental Systems, Danvers MA*, (July, 1998).
[13] P. Maybeck and R.D. Stevens, 'Reconfigurable flight control via multiple model adaptive control methods', *IEEE Transactions on Aerospace and Electronic Systems*, **27**(3), 470–480, (1991).
[14] S. McIlraith, 'Diagnosing hybrid systems: a bayseian model selection approach', in *Proceedings of the 11th International Workshop on Principles of Diagnosis (DX00)*, pp. 140–146, (June 2000).
[15] P.J. Mosterman and G. Biswas, 'Building hybrid observers for complex dynamic systems using model abstractions', in *Hybrid Systems: Computation and Control (HSCC'99)*, eds., F. Vaandrager and J. Schuppen, volume 1569 of *LNCS*, 178–192, Springer Verlag, (1999).
[16] S. Narasimhan and G. Biswas, 'Efficient diagnosis of hybrid systems using models of the supervisory controller', in *Proceedings of the 12th International Workshop on Principles of Diagnosis (DX01)*, pp. 127–134, (March 2001).
[17] P. Nayak, *Automated Modelling of Physical Systems*, Lecture Notes in Artificial Intelligence, Springer, 1995.
[18] E. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Springer, New York, Berlin, Heidelberg, 2 edn., 1998.
[19] P. Struss and O. Dressler, 'Physical negation: Integrating fault models into the general diagnostic engine', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'89)*, pp. 1318–1323, (1989).
[20] L. Travé-Massuyès and R. Pons, 'Causal ordering for multiple mode systems', in *Proceedings of the 11th International Workshop on Qualitative Reasoning (QR97)*, pp. 203–214, (1997).
[21] B. Williams and B. Millar, 'Decompositional, model-based learning and its analogy to diagnosis', in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, (1998).
[22] B. Williams and P. Nayak, 'A model-based approach to reactive self-configuring systems', in *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, (1996).
[23] A. S. Willsky, 'A survey of design methods for failure detection in dynamic systems', *Automatica*, **12**(6), 601–611, (1974).
[24] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, 'Distributed monitoring of hybrid systems: A model-directed approach', in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'01)*, pp. 557–564, (2001).

Acknowledgments

In part supported by NASA under contract NAG2-1388.

A Extended Kalman Filter

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 an extended Kalman filter[2] for this purpose, which updates its current state, like an HMM observer, in two steps. The first step uses the model to predict mean for 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 $\mathbf{u}_{c,(k-1)}$:

$$\hat{\mathbf{x}}_{c,(\bullet k)} = \mathbf{f}(\hat{\mathbf{x}}_{c,(k-1)}, \mathbf{u}_{c,(k-1)}) \tag{12}$$

$$\mathbf{A}_{(k-1)} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{c,(k-1)}, \mathbf{u}_{c,(k-1)}} \tag{13}$$

$$\mathbf{P}_{(\bullet k)} = \mathbf{A}_{(k-1)} \mathbf{P}_{(k-1)} \mathbf{A}^T_{(k-1)} + \mathbf{Q}. \tag{14}$$

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

$$\mathbf{r}_{(k)} = \mathbf{y}_{c,(k)} - \mathbf{g}(\hat{\mathbf{x}}_{c,(\bullet k)}, \mathbf{u}_{c,(k)}) \tag{15}$$

$$\mathbf{C}_{(k)} = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{c,(\bullet k)}, \mathbf{u}_{c,(k)}} \tag{16}$$

$$\mathbf{S}_{(k)} = \mathbf{C}_{(k)} \mathbf{P}_{(\bullet k)} \mathbf{C}^T_{(k)} + \mathbf{R}. \tag{17}$$

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

$$\mathbf{K}_{(k)} = \mathbf{P}_{(\bullet k)} \mathbf{C}^T_{(k)} \mathbf{S}^{-1}_{(k)} \tag{18}$$

$$\hat{\mathbf{x}}_{c,(k)} = \hat{\mathbf{x}}_{c,(\bullet k)} + \mathbf{K}_{(k)} \mathbf{r}_{(k)} \tag{19}$$

$$\mathbf{P}_{(k)} = \left[\mathbf{I} - \mathbf{K}_{(k)} \mathbf{C}_{(k)}\right] \mathbf{P}_{(\bullet k)}. \tag{20}$$

The output of the extended Kalman filter, as used in our hybrid estimation system, is a sequence of mean/covariance pairs $\langle \hat{\mathbf{x}}_{c,(k)}, \mathbf{P}_{(k)} \rangle$ for $\mathbf{x}_{c,(k)}$ as well as the hybrid probabilistic observation function

$$P_{\mathcal{O}}(\mathbf{y}_{(k)} | \hat{\mathbf{x}}_{(k)}, \mathbf{u}_{c,(k)}) = e^{-\mathbf{r}^T_{(k)} \mathbf{S}^{-1}_{(k)} \mathbf{r}_{(k)}/2}. \tag{21}$$