Hybrid Estimation of Complex Systems

Michael W. Hofbaur, *Member, IEEE,* and Brian C. Williams

***Abstract*—Modern automated systems evolve both continuously and discretely, and hence require estimation techniques that go well beyond the capability of a typical Kalman Filter. Multiple model (MM) estimation schemes track these system evolutions by applying a bank of filters, one for each discrete system mode. Modern systems, however, are often composed of many interconnected components that exhibit rich behaviors, due to complex, system-wide interactions. Modeling these systems leads to complex stochastic hybrid models that capture the large number of operational and failure modes. This large number of modes makes a typical MM estimation approach infeasible for online estimation.**

This paper analyzes the shortcomings of MM estimation, and then introduces an alternative hybrid estimation scheme that can efficiently estimate complex systems with large number of modes. It utilizes search techniques from the toolkit of model-based reasoning in order to focus the estimation on the set of most likely modes, without missing symptoms that might be hidden amongst the system noise. In addition, we present a novel approach to hybrid estimation in the presence of unknown behavioral modes. This leads to an overall hybrid estimation scheme for complex systems that robustly copes with unforeseen situations in a degraded, but fail-safe manner.

***Index Terms*—Artificial intelligence, diagnosis, fault detection and isolation (FDI), hybrid systems, multiple model estimation.**

I. Introduction

TODAY'S COMPLEX artifacts, such as space probes or process automation systems, can be characterized by their high complexity and the demand for high performance, availability, and safety. Advanced control and automation methods play an important role in achieving these requirements. While advanced control methods improve performance and robustness along many dimensions, their added complexity also makes these systems vulnerable to unanticipated failures. For instance, a malfunction in a sensor or an actuator can produce undesired effects. To avoid these negative effects, it is essential to be able to track a system's operation accurately and to identify faults, in order to react appropriately.

Traditional model-based fault detection and isolation (FDI) techniques (e.g., [1], [2]) are frequently used to identify faults in artifacts. These techniques compare the system's operation with predictions of a model and identify faults whenever the predictions deviate from measurements taken. They then isolate faults using some sort of diagnostic method. Our aim is to be able to track both a variety of operational modes of the system and arbitrary combinations of fault conditions. This goes well beyond the scope of the state of the art in FDI methods, due to the exponential number of ways in which fault conditions may be combined.

Hybrid estimation schemes solve a similar problem, but provide an integrated approach to estimate the system's current operational or failure mode, as well as the unmeasured (continuous) state variables in the system. An optimal hybrid estimator tracks every possible state trajectory of the system, that is, all possible mode sequences and their associated continuous state evolution. This *full hypotheses estimation* involves a number of trajectory hypotheses that is exponential in the number of time steps considered. Typically, the number of possible trajectories becomes too large to track after only a few time steps. Approximate hybrid estimation schemes address this exponential growth by merging trajectories, or pruning unlikely trajectories. For example, *multiple model (MM)* estimation algorithms work with a set of models that captures the possible operational and fault modes for a system. Estimation is performed by executing filters concurrently (one for each mode) and by combining the *mode-conditioned* estimates according to an algorithm-specific (probabilistic) weighting scheme. Examples of MM-estimation schemes include the generalized pseudo-Bayesian algorithm (GPB) [3], the detection-estimation method [4], the residual correlation Kalman filter bank [5], and the interacting multiple model algorithm (IMM) [6]. These methods track multiple hybrid estimates over time, but require at least as many filters as there are modes in the model of the system. This number can be very large for models of real-world artifacts. For example, consider a system with ten components. Each component has three nominal and two fault modes. The resulting hybrid model has $5^{10} \approx 10,000,000$ modes. It is currently infeasible for an online algorithm to execute 10 million filters.

Adaptive MM-estimation has been proposed as a possible solution to this dilemma. It adapts the mode set to a subset of modes that are most likely at a given time point. Adaptive MM-estimation methods (e.g., [7]–[9]) differ in the way in which they obtain an appropriate mode set. One example is to use the mode transition graph of the model in order to deduce the set of modes that are immediately reachable, given the previous estimate. This set, however, can still be very large. Reconsider the 10-component example above, while assuming that each mode of a component has on average two successor modes (one nominal and one fault mode). An adaptive MM-estimation method considers the current mode and its two successor modes for each component. This leads to $3^{10} \approx 60,000$ modes–still unrealistically large for online estimation.

In the domain of discrete systems, monitoring and diagnosis methods based on model-based reasoning [10] have proven to cope well with the complexity of modern artifacts. For example,

Manuscript received June 23, 2002; revised August 31, 2003. This work was supported in part by NASA under Contracts NAG2-1388 and NCC2-1235. This paper was recommended by Guest Editor G. Biswas.

M. W. Hofbaur is with the Institute of Automation and Control, Graz University of Technology, A-8010 Graz, Austria (e-mail: hofbaur@irt.tu-graz.ac.at)

B. C. Williams is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, (e-mail: williams@mit.edu)

Digital Object Identifier 10.1109/TSMCB.2004.835009

the Livingstone system [11] successfully monitored and diagnosed the DS-1 space probe in flight, a system with approximately 4^{80} modes of operation. Livingstone's mode estimation capability utilizes the concept of concurrent belief-state update and reformulates estimation as a k-best search problem. This reformulation enables the application of conflict directed A* search techniques ([12], [13]) in order to focus estimation on the most likely estimates only. 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 re-frames hybrid estimation as a k-best search problem. This allows us to utilize algorithms similar to Livingstone's discrete mode estimation, thus coping with the complexity of hybrid estimation for real-world systems.

Hybrid estimation and MM estimation schemes, in particular, work well as long as the underlying models are "close" mathematical descriptions of the physical artifact. They can fail severely whenever unforeseen situations occur. This means that the models must capture the exhaustive set of possible behavioral modes. This exhaustive model requirement of hybrid estimation is in contrast to assumptions taken in discrete model-based diagnosis [10]. Its underlying concept of constraint suspension [14] allows diagnosis of systems where no assumption is made about the behavior of one or several components of the system. In this way, model-based diagnosis schemes, such as the General Diagnostic Engine (GDE) [15] or the Sherlock system [16], capture unspecified and unforeseen behaviors of the system by considering an *unknown mode* that does not impose any constraint on the system's variables. This paper provides an approach for incorporating the concept of unknown mode into our hybrid estimation scheme. Our approach is based on an online model decomposition method that works around each unknown mode, by constructing a set of concurrent estimators for subsets of the system in which component modes have specified behaviors.

Several other approaches have been recently presented for hybrid system estimation and diagnosis. Many of them provide alternative bridges between the fields of model-based diagnosis, MM estimation and hybrid systems. Most of these approaches (e.g., [17] and [18]) utilize a heterogeneous set of techniques taken from the two distinct research communities. [17], for example, utilizes Bayesian mode estimation together with a timed Petri net model for diagnosis. [18] combines hybrid behavior tracking, mode estimation, and qualitative-quantitative reasoning techniques for fault diagnosis of continuous systems with supervisory control. The hybrid framework for simulation, diagnosis, and system tracking that is described in [19] builds upon the same foundation as our work, the Livingstone system [11]. It utilizes interval descriptions to express uncertainties at the continuous level, probabilities for discrete mode estimation and a continuous/discrete interface in order to synchronize both types of estimation. In contrast, our work provides a homogeneous framework that makes extensive use of probabilities for both continuous state and discrete mode estimation. Other estimation methods that should be noted here build upon alternative AI methods for state estimation, such as Bayesian networks [20] or particle filters [21]–[26].

This paper begins with an introduction to hybrid estimation in Section II. We then motivate approximate hybrid estimation schemes in Section III, review traditional approaches in Section III-A, and show their limitations for complex system estimation. Section III-B presents our hybrid estimation framework (HME), introduced in [27], and reviews its model-based reasoning roots. In Section IV, we extend our basic hybrid estimation framework to handle the unknown mode, through a model decomposition method based on causal ordering. This leads to a robust and efficient hybrid estimation method that combines classical state estimation with model-based reasoning. Section V uses examples to demonstrate HME, evaluates its performance, and compares it to IMM.

II. Hybrid Estimation

We model a physical artifact using a stochastic hybrid discrete-time model (sampling-period T_s) with continuously valued inputs $\mathbf{u}_c = [u_{c1}, \ldots, u_{cn_u}]^T$, observations $\mathbf{y}_c = [y_{c1}, \ldots, y_{cn_y}]^T$ and discretely valued (command) inputs $\mathbf{u}_d = [u_{d1}, \ldots, u_{dn_c}]^T$. The *mode* of the hybrid model is captured by the discrete state variables[1] $\mathbf{x}_d = [x_{d1}, \ldots, x_{d\zeta}]^T$ and has domain $\mathcal{X}_d = \{\mathbf{m}_1, \ldots, \mathbf{m}_l\}$. The *continuous state variables* $\mathbf{x}_c = [x_{c1}, \ldots, x_{cn_x}]^T$ capture the continuously valued dynamic evolution of the hybrid model, which is specified in terms of stochastic difference equations

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

with observations

$$\mathbf{y}_{c,k} = \mathbf{g}(\mathbf{x}_{c,k}, \mathbf{u}_{c,k}, \mathbf{x}_{d,k}) + \mathbf{v}_{o,k}. \tag{2}$$

With $\mathbf{x}_{c,k}$ and $\mathbf{y}_{c,k}$, we denote the valuation of the continuous state and output at the time step k ($t = T_s k$). $\mathbf{x}_{d,k}$ denotes the mode that is in effect during the sampling-period k ($T_s(k-1) < t \leq T_s k$). $\mathbf{v}_{s,k}$ and $\mathbf{v}_{o,k}$ denote zero-mean Gaussian state and measurement noise that are characterized in terms of their covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. The mode $\mathbf{x}_{d,k}$, together with the continuous state $\mathbf{x}_{c,k}$, specify the *hybrid state* $\mathbf{x}_k = \langle \mathbf{x}_{d,k}, \mathbf{x}_{c,k} \rangle$ at time step k. The discretely valued evolution of the hybrid model is characterized in terms of the *probabilistic transition function*

$$P_{\mathcal{T}}(\mathbf{m}_i, \mathbf{x}_{k-1}, \mathbf{u}_{d,k-1}) := P(\mathbf{m}_i | \mathbf{x}_{k-1}, \mathbf{u}_{d,k-1}) \tag{3}$$

which specifies the conditional probability of mode transitions $\mathbf{x}_{d,k-1} = \mathbf{m}_j \rightarrow \mathbf{x}_{d,k} = \mathbf{m}_i$, given the hybrid state $\mathbf{x}_{k-1} = \langle \mathbf{m}_j, \mathbf{x}_{c,k-1} \rangle$ and command input $\mathbf{u}_{d,k-1}$.

The mode $\mathbf{x}_{d,k}$ and the continuous state $\mathbf{x}_{c,k}$ are only observed indirectly, in terms of the noisy observation $\mathbf{y}_{c,k}$. As a consequence, one has to use a *hybrid estimator* to infer the state. This hybrid estimation problem can be stated as follows.

A. Hybrid Estimation Problem

Given a hybrid discrete-time model [(1)–(3)], initial state information $\mathcal{X}_0$ and the sequences of observa-

[1] We assume a system with ζ components and specify its mode in terms of one discrete state variable $x_{di} \in \mathbf{x}_d$ per component.

tions $Y_{c,k} = \{\mathbf{y}_{c,1}, \dots, \mathbf{y}_{c,k}\}$ and control inputs $U_{c,k} = \{\mathbf{u}_{c,0}, \mathbf{u}_{c,1}, \dots \mathbf{u}_{c,k}\}$ and $U_{d,k} = \{\mathbf{u}_{d,0},\ \mathbf{u}_{d,1},\ \dots,\ \mathbf{u}_{d,k}\}$, estimate the hybrid state $\mathbf{x}_k$ at time step k.

Our model allows switching among the modes. As a consequence, an optimal hybrid estimator has to track *possible trajectories* involving mode changes

$$X_k = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k-1}, \mathbf{x}_k\} \tag{4}$$

and obtain their estimates, which we denote by $\hat{X}_k = \{\hat{\mathbf{x}}_0, \dots, \hat{\mathbf{x}}_k\}$. A hybrid state estimate $\hat{\mathbf{x}}_k$ defines the tuple

$$\hat{\mathbf{x}}_k := \langle \hat{\mathbf{x}}_{d,k}, p_{c,k} \rangle \tag{5}$$

with mode $\hat{\mathbf{x}}_{d,k}$ and the multivariate probability density function (pdf) $p_{c,k}$ as continuous state estimate with mean $\hat{\mathbf{x}}_{c,k}$ and covariance matrix $\mathbf{P}_k$.

Since we are dealing with a *set of trajectory hypotheses*[2], we will use a superscript index in parentheses, for example, $\hat{X}_k^{(j)}$, to refer to the j'th *trajectory hypothesis* $\hat{X}_k^{(j)} = \{\dots, \hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{x}}_k^{(j)}\}$ that estimates a possible trajectory up to the time step k. This trajectory hypothesis includes the hybrid estimate $\hat{\mathbf{x}}_k^{(j)}$, at its fringe. This *fringe estimate* specifies the mode estimate $\hat{\mathbf{x}}_{d,k}^{(j)}$ and the continuous estimate in terms of a pdf $p_{c,k}^{(j)}$ with mean $\hat{\mathbf{x}}_{c,k}^{(j)}$. The trajectory hypotheses $\hat{X}_k^{(j)}$, $j = 1, \dots, \lambda_k$ are ranked according to their conditional probability $P(\hat{X}_k^{(j)} | Y_{c,k}, U_k)$, where U_k denotes the combined sequence of continuous inputs $U_{c,k}$ and discrete (command) inputs $U_{d,k}$ up to time step k. These probabilities define a probability distribution b_k

$$b_k\left(\hat{X}_k^{(j)}\right) := P\left(\hat{X}_k^{(j)} | Y_{c,k}, U_k\right), \quad j = 1, \dots, \lambda_k \tag{6}$$

over the set of trajectory hypotheses

$$\{\hat{X}_k^{(1)}, \dots, \hat{X}_k^{(\lambda_k)}\}.$$

Thus, b_k and the fringe estimates $\{\hat{\mathbf{x}}_k^{(1)}, \dots, \hat{\mathbf{x}}_k^{(\lambda_k)}\}$ of the trajectory hypotheses specify the hybrid state distribution at time step k. In the following we will use the short notation $b_k^{(j)}$ for $b_k(\hat{X}_k^{(j)})$.

Hybrid estimation deduces trajectory estimates incrementally, starting with a set of initial estimates $\{\hat{X}_0^{(1)}, \dots, \hat{X}_0^{(\lambda_0)}\}$, specified in terms of the initial information $\mathcal{X}_0$ of the hybrid estimation problem. Hybrid estimation determines for all hypotheses $\hat{X}_{k-1}^{(i)} = \{\dots, \hat{\mathbf{x}}_{k-1}^{(i)}\}$ all possible transitions $\hat{\mathbf{x}}_{d,k-1}^{(i)} \rightarrow \hat{\mathbf{x}}_{d,k}^{(j)}$ and deduces the associated hybrid estimate $\hat{\mathbf{x}}_k^{(j)}$. Thus it extends a hypothesis $\hat{X}_{k-1}^{(i)}$ with possible successor states $\hat{\mathbf{x}}_k^{(j)}$ and obtains new trajectory hypotheses $\hat{X}_k^{(j)} = \{\dots, \hat{\mathbf{x}}_{k-1}^{(i)}, \hat{\mathbf{x}}_k^{(j)}\}$. This operation can be interpreted as building a *full-hypothesis tree* that encodes the estimates for the possible trajectories that the system can take on (see Fig. 1).

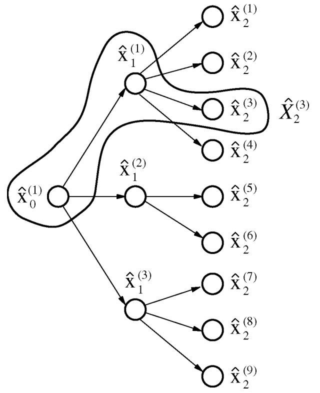

Fig. 1. Full-hypothesis tree and trajectory path for behavior $\hat{X}_2^{(3)}$.

The fringe estimates $\hat{\mathbf{x}}_k^{(j)}$ and their associated conditional probabilities $b_k^{(j)}$ are calculated according to the following two-step process (e.g., see [28] or [29]). The first step deduces possible transitions $\hat{\mathbf{x}}_{d,k-1}^{(i)} \rightarrow \hat{\mathbf{x}}_{d,k}^{(j)}$ and forms trajectory hypotheses $\hat{X}_k^{(j)}$. It then calculates the *prior probability* $b_{k|k-1}^{(j)}$ according to the previous posterior probability $b_{k-1}^{(i)}$ and the transition probability as

$$b_{k|k-1}^{(j)} = P_{\mathcal{T}}\left(\hat{\mathbf{x}}_{d,k}^{(j)}, \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{d,k-1}\right) b_{k-1}^{(i)}. \tag{7}$$

The second step takes the current measurement $\mathbf{y}_{c,k}$ into account. It performs (extended Kalman) filtering

$$p_{c,k-1}^{(i)} \rightarrow p_{c,k|k-1}^{(j)} \rightarrow p_{c,k}^{(j)}$$

according to the model for mode $\hat{\mathbf{x}}_{d,k}^{(j)}$ and calculates the *posterior probability* $b_k^{(j)}$ as

$$b_k^{(j)} = \frac{P_{\mathcal{O}}\left(\mathbf{y}_{c,k}, \hat{\mathbf{x}}_{k|k-1}^{(j)}, \mathbf{u}_{c,k}\right) b_{k|k-1}^{(j)}}{\sum_{i=1}^{\lambda_k} P_{\mathcal{O}}\left(\mathbf{y}_{c,k}, \hat{\mathbf{x}}_{k|k-1}^{(i)}, \mathbf{u}_{c,k}\right) b_{k|k-1}^{(i)}}. \tag{8}$$

The *hybrid observation function*

$$P_{\mathcal{O}}\left(\mathbf{y}_{c,k}, \hat{\mathbf{x}}_{k|k-1}^{(j)}, \mathbf{u}_{c,k}\right) := p\left(\mathbf{y}_{c,k} | \hat{\mathbf{x}}_{k|k-1}^{(j)}, \mathbf{u}_{c,k}\right) \tag{9}$$

quantifies the level of agreement between the prior estimate $\hat{\mathbf{x}}_{k|k-1}^{(j)} = \langle \hat{\mathbf{x}}_{d,k}^{(j)}, p_{c,k|k-1}^{(j)} \rangle$ and the observation $\mathbf{y}_{c,k}$. Its value is determined by

$$P_{\mathcal{O}}\left(\mathbf{y}_{c,k}, \hat{\mathbf{x}}_{k|k-1}^{(j)}, \mathbf{u}_{c,k}\right) = \frac{1}{|2\pi \mathbf{S}_k|^{1/2}}\, e^{-0.5\mathbf{r}_k^T \mathbf{S}_k^{-1} \mathbf{r}_k} \tag{10}$$

where $\mathbf{r}_k = \mathbf{y}_{c,k} - \mathbf{g}(\hat{\mathbf{x}}_{c,k|k-1}^{(j)}, \mathbf{u}_{c,k}, \hat{\mathbf{x}}_{d,k}^{(j)})$ denotes the filter innovation with associated covariance matrix $\mathbf{S}_k$.

III. Approximate Hybrid Estimation

As discussed earlier, tracking all possible trajectories of a system is almost always intractable, because the number of trajectory hypotheses becomes too large after only a few time steps (Fig. 2 illustrates this fact for the ten-component example that was introduced above). Therefore, it is inevitable to use approximate hybrid estimation schemes.

[2] To be precise, one tracks possible *mode sequences* with their associated continuous trajectories and the estimates thereof [28], [29]. However, for the sake of a concise presentation, we omit this extra detail and describe hybrid estimation in terms of *trajectory hypotheses* only.

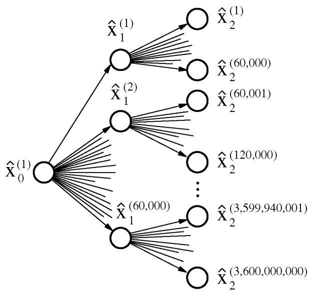

Fig. 2. Two-step full hypothesis tree for a (moderately) complex system.

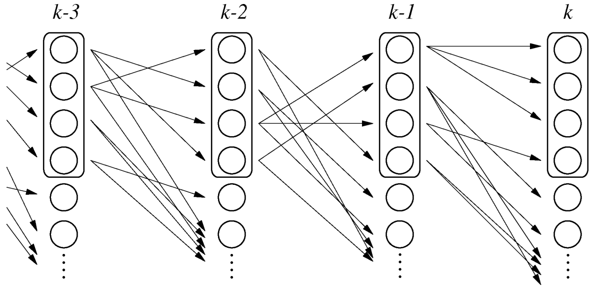

Fig. 3. K-best filtering for hybrid estimation.

A. MM Estimation

Various MM estimation schemes address the intractability problem of hybrid estimation. A good tradeoff between computational cost and estimation quality is achieved by the *interacting multiple model (IMM)* algorithm [6]. IMM copes with the exponential explosion over time by merging trajectory hypotheses which share the same mode in their fringe estimate, prior filtering. As a consequence, IMM requires at most l concurrent filters, where l is the number of modes of the hybrid model. The shift from *trajectory conditioned* estimates to *mode conditioned* ones is reflected in the first estimation step. IMM replaces (7) with

$$b_{k|k-1}^{(j)} = \sum_{i=1}^{\lambda_{k-1}} P_{\mathcal{T}}\left(\hat{\mathbf{x}}_{d,k}^{(j)}, \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{d,k-1}\right) b_{k-1}^{(i)} \tag{11}$$

and merges the previous mode conditioned estimates $p_{c,k-1}^{(i)}$ accordingly

$$\bar{p}_{c,k-1}^{(j)} = \frac{1}{b_{k|k-1}^{(j)}} \sum_{i=1}^{\lambda_{k-1}} P_{\mathcal{T}}\left(\hat{\mathbf{x}}_{d,k}^{(j)}, \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{d,k-1}\right) b_{k-1}^{(i)}\, p_{c,k-1}^{(i)}. \tag{12}$$

The merged mode conditioned estimates $\bar{p}_{c,k-1}^{(j)}$ serve as initial values for the consecutive filtering step of the IMM algorithm. This step is equivalent to (8) but utilizes a limited number of $\lambda_k \leq l$ *mode* hypotheses.

Standard IMM uses as many hypotheses as there are modes in the system, that is, $\lambda_k = l$. This number of modes, however, can be very large. Therefore, *variable-structure*, or *adaptive* MM-estimation [7]–[9] was proposed to reduce the number of hypotheses. Adaptive MM-estimation works in two phases. The first *mode-set adaption phase* uses a statistical test to choose suitable subsets of modes $\mathcal{M}_k \subseteq \mathcal{X}_d$ that are most likely at a given time point k. The second phase applies a standard MM-estimation algorithm with the reduced mode-set $\mathcal{M}_k$. This clear separation of mode-set adaption and estimation has the advantage that various mode-set adaption schemes can be used together with a standard MM-estimation algorithm, such as IMM. A major drawback, however, is that the algorithms do not make use of *posterior* information that is gathered in the course of estimation. Furthermore, the mode-sets $\mathcal{M}_k$ can still be very large. Recall our hypothetical ten-component model that has $l = 5^{10} \approx 10,000,000$ modes. Mode-set adaption would consider for each state update the current mode and, on average, two successor modes (one nominal and one fault mode). This leads to $3^{10} \approx 60,000$ modes, which is still beyond the scope of a typical MM-estimation algorithm.

B. Focused Hybrid Estimation

Some discrete model-based monitoring and diagnosis systems, for example the Livingstone and Titan systems [11], [30], build upon the concept of the hidden Markov model (HMM) style *belief-state update*. They reformulate HMM belief-state update as a discrete, constraint-based, multiattribute utility problem, and employ conflict-directed search techniques ([12], [13]) in order to focus estimation onto the most likely estimates. For hybrid estimation, we utilize similarities with this previous work in order to re-frame the hybrid estimation task as search. This bridges prior work in model-based reasoning with MM-estimation and allows us to cope with the complexity of hybrid estimation for real-world systems.

In order to achieve successful operation within a real-time computing environment, we proposed an any-time/any-space solution [27] that dynamically adjusts the number of *trajectory hypotheses* tracked in order to fit within the processor's computational and memory limits. This leads to an approximate hybrid estimation algorithm that *focuses* onto the leading set $\mathcal{X}_k = \{\hat{X}_k^{(1)}, \ldots, \hat{X}_k^{(\eta_k)}\}$ of $\eta_k \ll \lambda_k$ hypotheses and *truncates* less likely ones (for simplicity, we assume in the following that the superscript index j of a hypothesis $\hat{X}_k^{(j)}$ also denotes its *rank*). One can interpret this operation as carefully exploring the full hypothesis tree with *beam search*, also known as k-best filtering. Fig. 3 illustrates this operation for a fixed fringe size of $\eta_k = 4$.

Limiting the number of hypotheses at each time step to η_k deals with the exponential growth of hypotheses over time. However, we still face the dilemma of generating the next η_k best estimates, at each time step, without incurring an prohibitive amount of work. To accomplish this we employ a *best-first successor state enumeration scheme*, generalizing from [11].

Let us recall the two operational steps of hybrid estimation. Given the fringe estimate $\hat{\mathbf{x}}_{k-1}^{(i)}$, hybrid estimation selects a possible transition $\hat{\mathbf{x}}_{d,k-1}^{(i)} \rightarrow \hat{\mathbf{x}}_{d,k}^{(j)}$ with associated probabilistic transition function $P_{\mathcal{T}}$ in the first step, and filters the state in the second step. The conditional probability for the resulting state is

$$b_k^{(j)} = \frac{1}{c} \cdot P_{\mathcal{O}} \cdot P_{\mathcal{T}} \cdot b_{k-1}^{(i)} \tag{13}$$

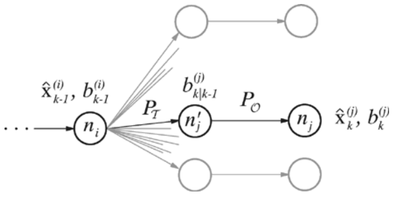
$\hat{x}^{(i)}_{k-1}$, $b^{(i)}_{k-1}$
n_i
$P_\mathcal{T}$
$b^{(j)}_{k|k-1}$
n'_j
$P_\mathcal{O}$
n_j
$\hat{x}^{(j)}_k$, $b^{(j)}_k$

Fig. 4. Successor deduction $\hat{\mathbf{x}}_{k-1}^{(i)} \rightarrow \hat{\mathbf{x}}_k^{(j)}$.

where $P_\mathcal{O}$ denotes the probabilistic observation function for $\hat{\mathbf{x}}_{c,k}^{(j)}$ and c denotes the normalization term. We can interpret these operations as building the search tree as shown in Fig. 4. The transition function ($P_\mathcal{T}$) and the observation function ($P_\mathcal{O}$) along a path determine the rank of the corresponding estimate $\hat{\mathbf{x}}_k^{(j)}$ among all successors of $\hat{\mathbf{x}}_{k-1}^{(i)}$. If we take the *negative logarithm* of $b_k^{(j)}$, then we can view this problem as a *shortest path* problem, where the negative logarithm of the conditional probabilities along the arcs of the tree correspond to *path costs*, and one seeks the path with the *lowest combined cost*. In a shortest path problem, it makes sense to ensure that the path costs are nonnegative. The negative logarithm of the transition function $P_\mathcal{T}$ obeys this property. However, $P_\mathcal{O}$ denotes the value of a multivariate Gaussian pdf [(10)] so that its negative logarithm is not necessarily nonnegative. Therefore, we use a slightly modified observation function that fixes this problem[3]:

$$\bar{P}_\mathcal{O}\left(\mathbf{y}_{c,k}, \hat{\mathbf{x}}_{k|k-1}^{(j)}, \mathbf{u}_{c,k}\right) := e^{-0.5\mathbf{r}_k^T\mathbf{S}_k^{-1}\mathbf{r}_k}. \qquad (14)$$

In order to solve this search problem efficiently, we exploit the fact that the overall hybrid model is composed of ζ individual components and that the transitions of the components are conditionally independent of each other. Thus, we can consider possible mode transitions component-wise. Component-based approaches for identifying mode transitions were introduced within the Livingstone and Titan mode estimation components, based on the conflict-directed A* search algorithm [13], generalized from the GDE and Sherlock systems for multiple fault diagnosis. Similar to this mode estimation, we can now formulate successor enumeration as a shortest path problem with an unfolded search tree that consists of ζ transition steps, one for each component, and one estimation step as shown in Fig. 5.

Search based successor enumeration starts with one or several estimates $\{\hat{\mathbf{x}}_{k-1}^{(1)}, \ldots, \hat{\mathbf{x}}_{k-1}^{(\eta_k-1)}\}$ and incrementally builds a search tree. A node n at a tree-depth of $d(n) = \zeta + 1$ represents a *goal node* that encodes a successor estimate $\hat{\mathbf{x}}_k^{(j)}$. The underlying operations are: 1) to decide possible transitions for individual components (expansion of nodes with a tree depth $0, \ldots, \zeta - 1$) and 2) to perform (extended Kalman) filtering in the last expansion step.

Best-first search with A* represents a search procedure for shortest path problems that focuses onto the paths that lead to goal nodes that encode the leading set of successor states $\hat{\mathbf{x}}_k^{(1)}, \ldots, \hat{\mathbf{x}}_k^{(\eta_k)}$. For this purpose, A* search incrementally expands search tree nodes that seem most promising. The "promise" of a node n_ν is measured by *utility* $f(n_\nu)$, which combines the cost $g(n_\nu)$ from the root node to node n_ν and a (conservative) estimate $h(n_\nu)$ of the expected cost to go. More specifically, we define $g(n_\nu)$ and $h(n_\nu)$ for a node n_ν at tree depth $d(n_\nu) = \iota$ of a search tree with a root node that encodes $\hat{\mathbf{x}}_{k-1}^{(i)}$ as follows: In terms of the cost from the root to node n_ν, we add the negative logarithms of the transition functions $P_{\mathcal{T}\,\xi j}$ so far and the modified observation function $\bar{P}_{\mathcal{O}\,j}$ if $\iota = \zeta + 1$

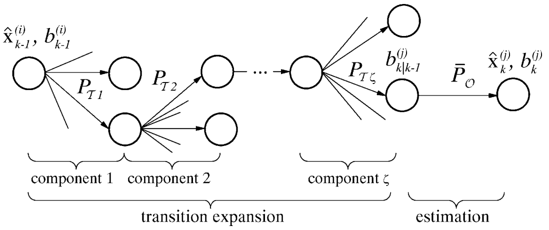

Fig. 5. Best-first search tree for successor enumeration.

$$g(n_\nu) := -\ln(b_{k-1}^{(i)}) - \sum_{\xi=1}^{\iota} \ln(P_\xi)$$
$$\text{where } P_\xi := \begin{cases} P_{\mathcal{T}\,\xi j} & \xi = 1, \ldots, \zeta \\ \bar{P}_{\mathcal{O}\,j} & \xi = \zeta + 1 \end{cases}. \qquad (15)$$

The offset $-\ln(b_{k-1}^{(i)})$ for the root node allows us to handle several root nodes, one per fringe estimate $\hat{\mathbf{x}}_{k-1}^{(i)} \in \{\hat{\mathbf{x}}_{k-1}^{(1)}, \ldots, \hat{\mathbf{x}}_{k-1}^{(\eta_k-1)}\}$, within one concurrent search problem. In terms of the expected cost to go, we consider the best transitions for the remaining components[4] and obtain

$$h(n_\nu) := \sum_{\xi=\iota+1}^{\zeta} -\ln(\max_\gamma P_{\mathcal{T}\,\xi\gamma}). \qquad (16)$$

The cost $g(n_\nu)$ and the heuristic $h(n_\nu)$ define the utility of a node n_ν as

$$f(n_\nu) := g(n_\nu) + h(n_\nu). \qquad (17)$$

A* is a complete and optimal search algorithm whenever the heuristic for the cost to go $h(n_\nu)$ is an *admissible heuristic*, that is, it *never overestimates* the cost [12]. Our definition in (16) ensures this property, since the most likely transitions provide the lowest cost. As a consequence, our search-based successor enumeration scheme provides the leading estimate $\hat{\mathbf{x}}_k^{(1)}$ upon its first call to A* search and any consecutive call provides the next best successor in the order of decreasing $b_k^{(j)}$. Generating successors in the order of decreasing likelihood consecutively enables us to terminate successor enumeration whenever we run short of computational resources. Completeness of the underlying A* search procedure guarantees that the set of successors $\{\hat{\mathbf{x}}_k^{(1)}, \ldots, \hat{\mathbf{x}}_k^{(\eta)}\}$, that was found so far, represents the leading set. Any consecutive call would only add a less likely successor.

[3] This solution is legitimate since the second step of hybrid estimation (8) ensures normalization of b_k. Independent of our motivation, Maybeck and Stevens [31] suggest this change for standard MM-estimation. They justify this change because of their observation that the normalization term represents an artificial bias that can lead to incorrect mode estimates.

[4] We omit an estimate for $\bar{P}_\mathcal{O}$ since the best possible value $\bar{P}_\mathcal{O} = 1$ induces $-\ln(1) = 0$.

IV. Hybrid Estimation With Unknown Behavioral Modes

Hybrid estimation schemes require models that are "close" mathematical descriptions of the system. They can fail severely whenever unforeseen, that is, unmodeled, situations occur. However, unanticipated failures do occur in real world systems, hence it is important that hybrid estimation copes with an *unknown mode* of operation. The concept of the *unknown mode* is central to model-based reasoning [10]. Model-based diagnosis systems, such as constraint suspension [14] or GDE [15], [32], capture unspecified and unforeseen behaviors of the system in terms of an *unknown mode* that does not impose any constraint on the system's variables.

This section incorporates this key concept into our hybrid estimation scheme. Our approach builds upon an online system decomposition capability that operates on our *concurrent probabilistic hybrid automata (cPHA)* model. We begin with a brief summary of our hybrid model (initially presented in [27]).

A. CPHA

Typical applications for our hybrid estimation scheme are comprised of individual components, each of which has several modes of operation and failure. We model individual components as a *probabilistic hybrid automata (PHA)*. The model of the overall system is then composed of a set of concurrently operating automata that interact via shared variables. This is a hybrid analogue of the concurrent constraint automata that were introduced in [11].

A *PHA* can be seen as an extension of a hidden Markov model that incorporates discrete-time difference and algebraic equations for each mode, in order to capture the continuously valued dynamic evolution of a system. More specifically, we describe the PHA as a tuple

$$\mathcal{A} := \langle \mathbf{x}, \mathbf{w}, F, P_{\mathcal{T}}, \mathcal{X}_d, \mathcal{X}_0, T_s \rangle \tag{18}$$

where $\mathbf{x} = \{x_d\} \cup \mathbf{x}_c$ denotes the hybrid *state variables* of the automaton, composed of the mode x_d and the continuous state variables $\mathbf{x}_c$[5]. $\mathcal{X}_d = \{m_1, \ldots, m_l\}$ specifies the set of possible modes. The set of I/O variables $\mathbf{w} = \mathbf{u}_d \cup \mathbf{w}_c$ aggregates the discrete (command) inputs $\mathbf{u}_d$ and continuous variables $\mathbf{w}_c$ that will establish the interconnection with other components and the outside world[6]. The continuous evolution of the automaton is specified by $F : \mathcal{X}_d \to F_{\mathrm{DE}} \cup F_{\mathrm{AE}}$ in terms of discrete-time difference equations F_{DE} (with sampling-period T_s) and algebraic equations F_{AE}. Finally, the state and command input dependent probabilistic transition function $P_{\mathcal{T}}$ specifies the discrete (mode) evolution of the automaton, as introduced in (3) above, and $\mathcal{X}_0$ denotes the initial state information.

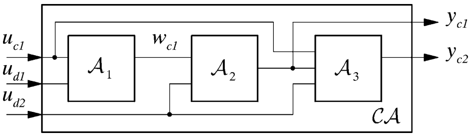

Fig. 6. Example cPHA composed of three PHAs.

A cPHA specifies the composition of a set of component automata [7]$A := \{\mathcal{A}_1, \ldots, \mathcal{A}_\zeta\}$ as a tuple

$$\mathcal{CA} := \langle A, \mathbf{u}_d, \mathbf{u}_c, \mathbf{y}_c, \mathbf{v}_s, \mathbf{v}_o, N_x, N_y \rangle. \tag{19}$$

The continuous input $\mathbf{u}_c$ and output variables $\mathbf{y}_c$ define disjoint subsets of the I/O variables $\mathbf{w}_c = \mathbf{w}_{c1} \cup \cdots \cup \mathbf{w}_{c\zeta}$ and establish the interconnection to the outside world. Shared continuous I/O variables $w_c \in \mathbf{w}_c$ establish the interconnection among the cPHA components $\mathcal{A}_i$. $\mathbf{u}_d = \mathbf{u}_{d1} \cup \cdots \cup \mathbf{u}_{d\zeta}$ denotes the set of discrete (command) inputs. A cPHA characterizes disturbances with respect to the outside world in terms of the noise variables $\mathbf{v}_s$ and $\mathbf{v}_o$, which act upon the state variables $\mathbf{x}_c = \mathbf{x}_{c1} \cup \cdots \cup \mathbf{x}_{c\zeta}$ and the output variables (observations) $\mathbf{y}_c$, respectively. We model the disturbances as zero-mean Gaussian noise and characterize them with $N_x : \mathcal{X}_d \to \mathbb{R}^{n_x \times n_x}$ and $N_y : \mathcal{X}_d \to \mathbb{R}^{n_y \times n_y}$. These mode dependent functions provide the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ for the state disturbance $\mathbf{v}_s$ and measurement noise $\mathbf{v}_o$, respectively. Fig. 6 illustrates the cPHA composition in terms of a simple example with three PHAs.

A cPHA specifies a mode dependent discrete-time model for a system with command inputs $\mathbf{u}_d$, continuous inputs $\mathbf{u}_c$, continuous outputs $\mathbf{y}_c$, mode $\mathbf{x}_d = \{x_{d1}, \ldots, x_{d\zeta}\}$, continuous state variables $\mathbf{x}_c = \mathbf{x}_{c1} \cup \cdots \cup \mathbf{x}_{c\zeta}$ and disturbances $\mathbf{v}_s$ and $\mathbf{v}_o$. The discrete-time evolution of $\mathbf{x}_c$ and $\mathbf{y}_c$ is described, similar to (1) and (2), by the nonlinear system of difference equations

$$\begin{aligned} \mathbf{x}_{c,k} &= \mathbf{f}_k(\mathbf{x}_{c,k-1}, \mathbf{u}_{c,k-1}) + \mathbf{v}_{s,k-1} \\ \mathbf{y}_{c,k} &= \mathbf{g}_k(\mathbf{x}_{c,k}, \mathbf{u}_{c,k}) + \mathbf{v}_{o,k}. \end{aligned} \tag{20}$$

The functions $\mathbf{f}_k$ and $\mathbf{g}_k$ are obtained automatically by symbolically solving[8] the set of equations

$$F(\mathbf{x}_{d,k}) = F_1(x_{d1,k}) \cup \cdots \cup F_\zeta(x_{d\zeta,k}) \tag{21}$$

for the mode of operation $\mathbf{x}_{d,k} = [x_{d1,k}, \ldots, x_{d\zeta,k}]^T$.

For example, consider the cPHA of Fig. 6 with

$$\begin{aligned} \mathcal{A}_1 &= \langle \{x_{d1}\}, \{u_{d1}, u_{c1}, w_{c1}\}, F_1, T_1, \{m_{11}, m_{12}\}, \ldots \rangle \\ \mathcal{A}_2 &= \langle \{x_{d2}, x_{c1}\}, \{u_{d2}, w_{c1}, y_{c1}\}, F_2, T_2 \\ &\qquad \{m_{21}, m_{22}, m_{23}\}, \ldots \rangle \\ \mathcal{A}_3 &= \langle \{x_{d3}, x_{c2}, x_{c3}\}, \{u_{d2}, u_{c1}, y_{c1}, y_{c2}\}, F_3, T_3 \\ &\qquad \{m_{31}, m_{32}, m_{33}\}, \ldots \rangle \\ \mathcal{CA} &= \langle \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}, \{u_{d1}, u_{d2}\}, \{u_{c1}\}, \{y_{c1}, y_{c2}\}, \ldots \rangle. \end{aligned} \tag{22}$$

[5] In the context of PHAs/cPHAs, we interpret a vector $\mathbf{v} = [v_1, \ldots, v_l]^T$ with components v_i also as a *set* of variables $\{v_1, \ldots, v_l\}$. This enables us, for example, to apply set operations to aggregate variables, identify shared variables among components.

[6] A PHA does not explicitly specify whether a variable $w_{ci} \in \mathbf{w}_c$ denotes an input or output. The interconnection topology of a concurrent automata and the set of equations that are valid for a particular mode of operation implicitly determine whether a variable w_{ci} acts as an input or output of the PHA component.

[7] We denote the components of a PHA $\mathcal{A}_i$ by $x_{di}, \mathbf{x}_{ci}, \mathbf{u}_{di}, \mathbf{w}_{ci}, F_i$ *etc.*.

[8] Our symbolic solver utilizes the Weyl computer algebra system [33] and currently restricts the algebraic equations and nonlinear functions to ones that can be explicitly solved.

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

$$\begin{aligned} F_1(m_{11}) &= \{u_{c1} = 2.0 w_{c1}\} \\ F_2(m_{21}) &= \{x_{c1,k+1} = 0.95 x_{c1,k} + w_{c1,k},\ y_{c1} = 2.0 x_{c1}\} \\ F_3(m_{31}) &= \{x_{c2,k+1} = x_{c3,k} + 0.2 y_{c1,k} \\ &\qquad x_{c3,k+1} = -0.63 x_{c2,k} + 1.6 x_{c3,k} + 0.1 u_{c1,k} \\ &\qquad y_{c2} = 0.5 x_{c2} + 0.1 x_{c3}\}. \end{aligned} \tag{23}$$

This leads to the discrete-time model

$$\begin{aligned} x_{c1,k+1} &= 0.95 x_{c1,k} + 0.5 u_{c1,k} + v_{s1,k} \\ x_{c2,k+1} &= 0.4 x_{c1,k} + x_{c3,k} + v_{s2,k} \\ x_{c3,k+1} &= -0.63 x_{c2,k} + 1.6 x_{c3,k} + 0.1 u_{c1,k} + v_{s3,k} \\ y_{c1,k} &= 2.0 x_{c1,k} + v_{o1,k} \\ y_{c2,k} &= 0.5 x_{c2,k} + 0.1 x_{c3,k} + v_{o2,k}. \end{aligned} \tag{24}$$

1) Unknown Mode of a cPHA: As a preparation for dealing with unknown behaviors, our cPHA compiler includes an unknown mode $m_?$ for each PHA $\mathcal{A}_i$ of the cPHA. In addition, the probabilistic transition function $P_{\mathcal{T}\,i}$ of $\mathcal{A}_i$ includes *unlikely* transitions with low probability (e.g., 0.001) from each mode $m_i \in \mathcal{X}_d$ to $m_?$ and uniformly distributed reverse transitions. An unknown mode does not impose any constraint (equation) among its variables [14]. Hence, the function F_i returns an *empty set of equations* upon a call with argument $m_?$.

These extensions complete our integration of the unknown mode into our hybrid model. The next section outlines extensions to our hybrid estimation scheme that enables HME to deal with unanticipated behavior in terms of the unknown mode.

B. Filter Decomposition and Hybrid Estimation With Unknown Modes

Hybrid estimation deduces trajectory hypotheses by repeatedly generating possible modes $\hat{\mathbf{x}}_{d,k}^{(j)}$ through consecutive filtering. The addition of the unknown mode to the cPHA can lead to mode hypotheses in which one or more components are at $m_?$. Such a mode hypothesis, however, leads to an incomplete set of equations $F(\hat{\mathbf{x}}_{d,k}^{(j)})$, since components with mode $m_?$ do not impose any constraint upon their continuous state and I/O variables. As a consequence, we cannot deduce a mathematical model in the form of (20) for the overall system, which is required to derive the appropriate extended Kalman filter in order to perform the estimation step.

Let us assume that our 3-PHA model is in the mode $\hat{\mathbf{x}}_{d,k}^{(j)} = [m_?, m_{21}, m_{31}]^T$, that is, component 1 ($\mathcal{A}_1$) is in the unknown mode. The set of equations $F(\hat{\mathbf{x}}_{d,k}^{(j)})$ omits any relation between the input u_{c1} and the internal variable w_{c1}. As a consequence, we fail to provide the extended Kalman filter. However, a close look at the PHA interconnection (Fig. 7)[9] reveals that we can estimate component 3 by its observed output y_{c2} and the observation y_{c1} as a substitute for its input value. This intuitive approach utilizes a decomposition of the cPHA into two systems, as shown in Fig. 8.

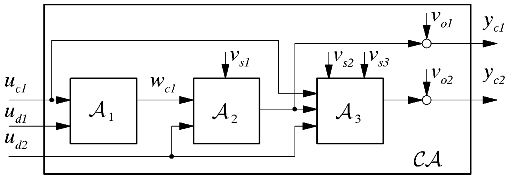

Fig. 7. Example cPHA extended with noise inputs and causality.

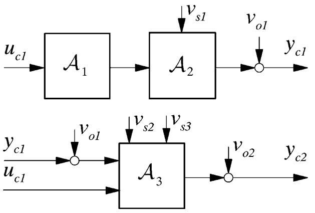

Fig. 8. Decomposed cPHA.

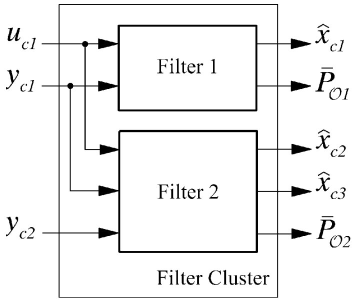

Fig. 9. Decomposed filter.

The decomposition allows us to treat the parts of the system independently. We deduce a *filter cluster* (Fig. 9) that consists of two concurrent filters. The incomplete set of equations prevents us from calculating the first filter, for the components $\mathcal{A}_1$ and $\mathcal{A}_2$, but it provides enough information to derive the second filter, which estimates the state of $\mathcal{A}_3$. However, filter cluster deduction has to account for the fact that we use the *measurement* y_{c1} of the input to $\mathcal{A}_3$ in replacement for its true value. This can be interpreted as having virtual additive noise at the component's input, as indicated in Fig. 8. A straightforward modification of the covariance matrix $\mathbf{Q}_{f2}$ for the state variables of the second filter takes this into account

$$\mathbf{Q}_{f2} = \mathbf{b}_3 r_1 \mathbf{b}_3^T + \mathbf{Q}_3 \tag{25}$$

where r_1 denotes the variance of v_{o1}, and $\mathbf{b}_3 = [0,\ 1]^T$ denotes the input vector of $\mathcal{A}_3$ with respect to y_{c1}.

The system decomposition is essential for incorporating the unknown mode. The decomposition leads to a factorization of the probabilistic observation function

$$\bar{P}_{\mathcal{O}} = \prod_j \bar{P}_{\mathcal{O}j} \tag{26}$$

where $\bar{P}_{\mathcal{O}j}$ denotes the probabilistic observation function of the jth filter in the filter cluster. This factorization allows us to calculate an upper bound for $\bar{P}_{\mathcal{O}}$ whenever one or more components of the system are in the unknown mode. We simply take

[9] The figure extends Fig. 6 with the implicit noise inputs, as well as it indicates the causality (directionality) for the internal I/O variables.

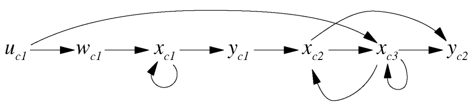

Fig. 10. Causal graph for the 3-PHA example.

the product over the remaining filters in the cluster.[10] In our example, with unknown component $\mathcal{A}_1$ this would mean

$$\bar{P}_{\mathcal{O}} \leq \bar{P}_{\mathcal{O}2}$$

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

The following subsection provides an algorithmic solution for the system analysis, decomposition and filter cluster deduction that we introduced informally above.[11]

C. System Decomposition and Filter Cluster Deduction

The starting point for the decomposition of the system at a mode $\mathbf{x}_{d,k}$ is the set of equations

$$F(\mathbf{x}_{d,k}) = F_1(x_{d1,k}) \cup \cdots \cup F_\zeta(x_{d\zeta,k}). \tag{27}$$

Although we still solve the equations symbolically in order to obtain a mathematical model of form (20), we interpret (27) as the *raw model* for the mode $\mathbf{x}_{d,k}$. The following decomposition performs a structural analysis of the raw model based on causal analysis [35], [36], structural observability analysis [37]–[39] and graph decomposition [40].

A cPHA model does not *a priori* impose a fixed causal structure that explicitly specifies directionality of automaton interconnections. The specification of the (exogenous) input variables $\mathbf{u}_c \in \mathbf{w}_c$ of the cPHA and the set of equations specify causality implicitly. We obtain the causal dependencies for a particular mode $\mathbf{x}_{d,k}$ using the bipartite-matching based algorithm presented in [35]. The resulting *causal graph* records the causal dependencies among the variables of the cPHA $\mathcal{CA}$. We denote the causal graph by $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_{d,k})$ and omit the arguments when clear from context. Fig. 10 shows the graph for the the 3-PHA example at mode $\mathbf{x}_{d,k} = [m_{11}, m_{21}, m_{31}]^T$. Each vertex of the causal graph represents one equation $e_i \in F(\mathbf{x}_{d,k})$ 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 causal graph. Vertices without incoming edges specify the *exogenous* variables.

The goal of our analysis is to obtain a set of independent subsystems that utilize observed variables as *virtual inputs*. To accomplish this we: 1) slice the graph at observed variable vertices with outgoing edges; 2) insert new vertices to represent virtual inputs; and 3) re-map the sliced outgoing edges to the appropriate virtual input vertex. Fig. 11 demonstrates this re-mapping for the causal graph of Fig. 10. The cPHA specification [(22)] asserts y_{c1} and y_{c2} as observed. The vertex with dependent variable y_{c1} has an outgoing edge, thus we slice the graph at $y_{c1} \rightarrow x_{c2}$ and remap the edge to the virtual input uy_{c1}.

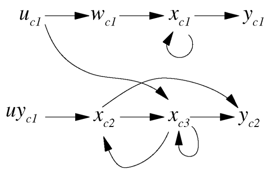

Fig. 11. Remapped causal graph for the 3-PHA example.

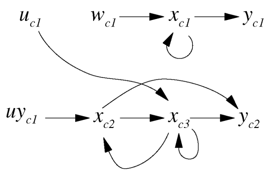

Fig. 12. Remapped causal graph for the 3-PHA example with unknown component $\mathcal{A}_1$.

A dynamic filter, such as an extended Kalman filter, can only estimate the observable part of the model. Therefore, it is essential to perform an observability analysis prior to calculating the filter, so that nonobservable parts of the model are excluded. Throughout this paper, we assume that loss of observability is caused by a structural defect of the model (e.g., a stuck-at fault of a sensor that disconnects, in the causal graph, the measurement from the rest of the model).[12] Therefore, we perform this analysis on a structural basis and evaluate for every variable $z \in \mathbf{x}_c \cup \mathbf{w}_c$ whether there exists at least one causally dependent output variable $y_c \in \mathbf{y}_c$ that can be used to estimate the value of z. More specifically:

Definition 1: We call a variable $z \in \mathbf{x}_c \cup \mathbf{w}_c$ of a cPHA $\mathcal{CA}$ at mode $\mathbf{x}_{d,k}$ structurally observable *(SO) whenever it is directly observed, that is, $z \in \mathbf{y}_c$, or there exists at least one path in the causal graph $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_{d,k})$ that connects the variable z to an output variable $y_c \in \mathbf{y}_c$* of $\mathcal{CA}$.

A filter estimates the state variables $\mathbf{x}_c$ based on observations $\mathbf{y}_c$ and the inputs $\mathbf{u}_c$ that act upon the state variables $\mathbf{x}_c$. Thus, estimation involves knowledge about the actuated inputs $\mathbf{u}_{c,k}$. This indicates that structural observability is not yet sufficient to determine the sub-model for estimation. We have to ensure that no variable with unknown value influences the state variables. To illustrate this fact, let us again consider the 3–PHA example with mode $\mathbf{x}_{d,k} = [m_?, m_{21}, m_{31}]^T$. Component 1 in the unknown mode omits the equation that relates u_{c1} with w_{c1}. This leads to a causal graph $\tilde{\mathcal{CG}}$ that labels w_{c1} as exogenous (Fig. 12). Thus, w_{c1} acts as an *unknown exogenous input*, influences the state variable x_{c1} and, as a consequence, prevents us from estimating it. Again, we perform a structural analysis of the causal graph that identifies variables that causally depend upon unknown exogenous variables, more specifically:

Definition 2: We call a variable $z \in \mathbf{x}_c \cup \mathbf{w}_c$ of a cPHA $\mathcal{CA}$ at mode $\mathbf{x}_{d,k}$ structurally determined *(SD) whenever it is an input variable, that is, $z \in \mathbf{u}_c$, or there does not exist a path in the causal graph $\mathcal{CG}(\mathcal{CA}, \mathbf{x}_{d,k})$ that connects an unknown exogenous variable $u_e \notin \mathbf{u}_c$ with z.*

From an algorithmic point of view, it is helpful to eliminate loops in the causal graph prior to checking variables against both

[10] This is equivalent to considering the upper bounds of the inequalities $\bar{P}_{\mathcal{O}j} \leq 1$ for each unknown filter j.

[11] An earlier version of this decomposition scheme was presented at the Principles of Diagnosis Workshop – DX-02 [34].

[12] Otherwise, it is necessary to perform an additional numerical observability test [41] as structural observability only provides a *necessary* condition for observability.

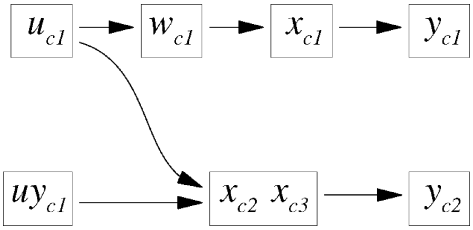

Fig. 13. Causal SCC graph for the 3-PHA example.

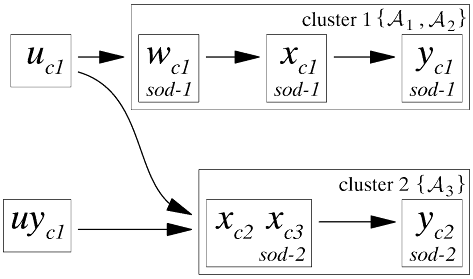

Fig. 14. Labeled and partitioned causal SCC graph for the 3-PHA example.

structural criteria. For this purpose, we calculate the *strongly connected components* [40] of the causal graph.

Definition 3: 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. 13 shows the remapped causal graph for the 3-PHA example after grouping variables into strongly connected components. The strong interconnection among variables in a strongly connected component implies that: 1) structural observability of variables in a strongly connected component follows directly from structural observability of at least one variable in the strongly connected component and 2) a variable in a strongly connected component is structurally determined, if and only if all variables in the strongly connected component are structurally determined. As a consequence, we can apply our structural analysis directly to strongly connected components and operate on the acyclic strongly connected component graph.

Our structural analysis algorithm [34] determines structural observability and determination (SOD) of a variable by traversing the strongly connected component graph backward from observed variables toward the inputs. In the course of this analysis, we label nonexogenous strongly connected components with a tag that records their influence upon observed variables. This indexing scheme allows us to cluster the variables into nonoverlapping subsets. The direct relation between a variable, its determining equation, and the cPHA component that specified this equation leads to the component clusters sought for (see Fig. 14 for the partitioned SCC graph of the 3-PHA example).

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 a nonlinear system of difference equations, of the form (20) with additional virtual inputs. This nonlinear system of difference equations determines the individual filter for the filter cluster.

1) Filter Cluster Deduction Within Our Hybrid Estimation Framework: A filter cluster with several concurrent extended Kalman filters and the overall extended Kalman filter are interchangeable, as they provide the same expected value for the continuous state ($E(\hat{\mathbf{x}}_{c,k})$) for nominal modes of the cPHA. However, the execution of several "small" filters outperforms a single "large" filter for the whole system (the computational requirements for a Kalman Filter with n_x state variables are approximately proportional to n_x^3 [42]). Therefore, we do not limit decomposition to hypotheses with unknown modes, but also use filter clusters for nominal modes $\mathbf{m}_j \in \mathcal{X}_d$ of the cPHA.

The number of modes of hybrid models for a complex physical artifact can be very large. This prevents us from compiling all filters a priori. However, deducing the same filter clusters over and over again would impose an unnecessarily large computational burden. Therefore, we *cache* a limited number of filters for re-use. This strategy represents a good compromise between the run-time cost of online deduction and the memory requirement of a priori filter compilation. Decomposition helps here as well. It significantly reduces the number of filter deductions since we can re-use the cached filters as "building blocks" for filter clusters. For example, the 3-PHA system of (22) has $2 \cdot 3 \cdot 3 = 18$ possible modes. The deduction of the 18 possible filter clusters (Fig. 9), however, requires only 9 filter deductions, $2 \cdot 3 = 6$ variations for filter 1 (components $\mathcal{A}_1$ and $\mathcal{A}_2$), and three variations for filter 2 (component $\mathcal{A}_3$).

Our overall hybrid estimation framework utilizes system decomposition and filter deduction as follows: Hybrid estimation deduces a suitable mode candidate $\hat{\mathbf{x}}_{d,k}^{(j)}$ after the first ζ steps of the underlying A* search procedure. A consecutive expansion of the associated search node performs the continuous estimation step. This involves: 1) retrieval of the raw model $F(\hat{\mathbf{x}}_{d,k}^{(j)})$; 2) the decomposition thereof; 3) cache retrieval or application of an algebraic solver to deduce subsets of difference and algebraic equations for each component cluster with consecutive filter deduction; and 4) execution of the filter cluster that leads to the new estimate $\hat{\mathbf{x}}_k^{(j)}$.

V. Examples

In this section, we present numerical examples and simulation results. This demonstrates our HME, and compares it to IMM and an optimal estimator. In addition, we demonstrate HME with a process automation example that is beyond the scope of MM-estimation.

A. A Simple Concurrent Automaton

We consider the three-component cPHA introduced above as a first example, in order to compare our hybrid estimation scheme to an IMM and variable-structure IMM based hybrid estimator. As a baseline, we use an artificial "optimal" estimator that utilizes the correct mode information.

The raw model for the modes m_{11}, m_{21} and m_{31} of the three components is given in (23). What remains is to specify the variations in the equation-sets for the other modes

$$\begin{aligned}
&F_1(m_{12}) = \{u_{c1} = -2.0w_{c1}\}\\
&F_2(m_{22}) = \{x_{c1,k+1} = 0.6x_{c1,k} + w_{c1,k}, \ldots\}\\
&F_2(m_{23}) = \{x_{c1,k+1} = 1.01x_{c1,k} + w_{c1,k}, \ldots\}\\
&F_3(m_{32}) = \{x_{c3,k+1} = -0.8x_{c2,k} + 1.6x_{c3,k} + 0.1u_{c1,k}, \ldots\}\\
&F_3(m_{33}) = \{x_{c3,k+1} = -0.3x_{c2,k} + 1.1x_{c3,k} + 0.1u_{c1,k}, \ldots\}.
\end{aligned} \tag{28}$$

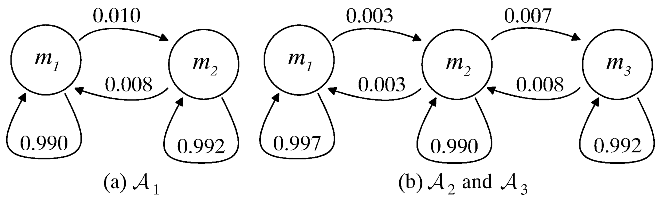

Fig. 15. Transition graphs for the three PHA example.

The transition graphs that specify T_1, T_2, and T_3 are given in Fig. 15. The transition probabilities are given as labels on the arcs of the digraphs and are independent of the continuous state variables and the command inputs.

The noise is assumed to be independent of the mode. Thus, N_x and N_y of the cPHA specify

$$\mathbf{Q} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.3 \end{bmatrix}. \tag{29}$$

The simulation results are based on a randomly generated simulation, with $k_{\max} = 5,000$ time samples. Figs. 16 and 17 show the first 500 samples of the mode sequence and measurements, respectively. We use the average relative estimation error of the algorithms to compare continuous filtering quality of the various algorithms. Mode estimation is evaluated by the number of wrongly classified modes, in particular, single, double and triple mode estimation errors, relative to the overall number of estimations ($k_{\max}$) as a percentage. Table I summarizes the estimation results for the IMM, variable-structure IMM (vs-IMM) and our hybrid estimation algorithm with a variety of fixed fringe sizes η. We analyze both the basic nonclustering algorithm (HMEη) and the extended algorithm with component clustering, as described above (c-HMEη). HME tests $\kappa > \eta$ hypotheses in order to obtain the leading set of η estimates (thus, it deduces and executes κ extended Kalman filters). The columns average and maximal number of tested hypotheses provide this information. Finally, we include a runtime comparison based on relative runtime for each algorithm under investigation (relative to the standard IMM algorithm)[13].

All algorithms perform well in terms of continuous estimation quality. Variable-structure IMM provides the same continuous estimation quality for half of the runtime cost of the standard IMM. However, the mode estimation of variable-structure IMM is significantly worse than IMM. Hybrid estimation (HME and c-HME) show slightly degraded continuous estimation quality when compared to IMM. However, HME and c-HME are significantly better than IMM in terms of runtime cost. This advantage will be even more dramatic in systems with a larger number of modes. It is evident that the estimation quality of HME and c-HME increases with the number of trajectory hypotheses that are considered during estimation (fringe size). Hybrid estimation with clustering, when compared to its nonclustering variant, performs better in terms of continuous estimation and mode estimation quality, as well as in terms of runtime cost. The reasonably good estimation quality for a small fringe size (e.g., c-HME2) suggests that a hybrid estimator that utilizes our estimation scheme can work reasonably well even under stringent resource constraints.

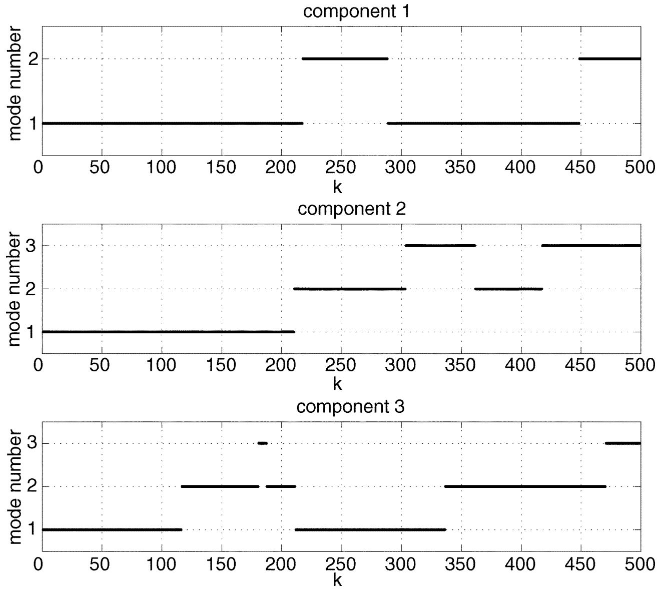

Fig. 16. Mode sequence $\mathbf{x}_{d,k} = \{x_{d1,k}, x_{d2,k}, x_{d3,k}\}$ of the 3-PHA experiment.

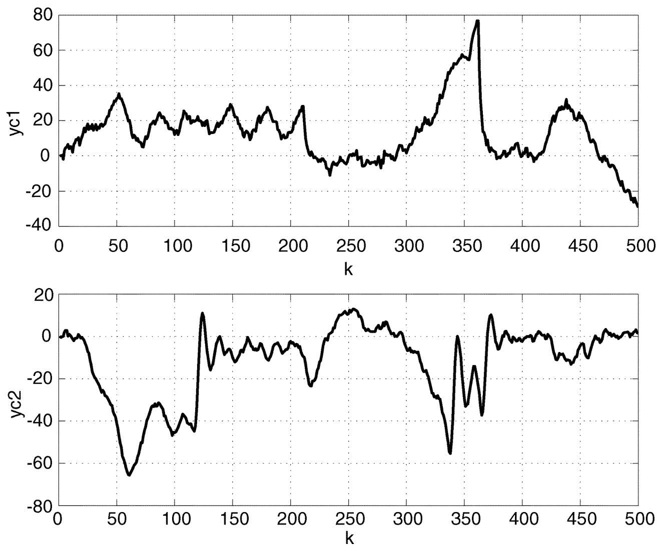

Fig. 17. Measurements $y_{c1,k}$, $y_{c2,k}$ for the 3-PHA experiment.

TABLE I
ALGORITHM COMPARISON

Algorithm	rel. error	mode errors [%]			fringe	tested hypotheses		rel.
	e	single	double	triple	size	average	max	runtime
optimal	0.1100	-	-	-	1	1	1	-
IMM	0.1130	12.6	1.5	0.3	18	18	18	1.00
vs-IMM	0.1130	39.4	24.7	0.8	-	10.4	18	0.47
HME2	0.1367	37.2	14.7	1.7	2	3.9	36	0.10
HME5	0.1173	20.5	7.0	0.4	5	10.3	70	0.24
HME10	0.1170	17.5	5.6	0.6	10	20.8	140	0.47
HME20	0.1172	18.3	5.6	0.6	20	42.0	280	0.98
c-HME2	0.1190	24.0	9.6	0.7	2	4.0	36	0.08
c-HME5	0.1169	17.1	4.4	0.2	5	10.4	70	0.19
c-HME10	0.1167	16.7	4.5	0.2	10	20.7	140	0.37
c-HME20	0.1167	13.9	3.2	0.5	20	42.0	280	0.77

[13] The IMM and vs-IMM algorithms that are used in this comparison are implemented as a subset of the overall cPHA estimation tool that is written in Common LISP. All estimation algorithms (IMM, vs-IMM, HME, and c-HME) utilize the same concurrent filter bank data structures and calling mechanisms, thus we can directly compare the runtime performances.

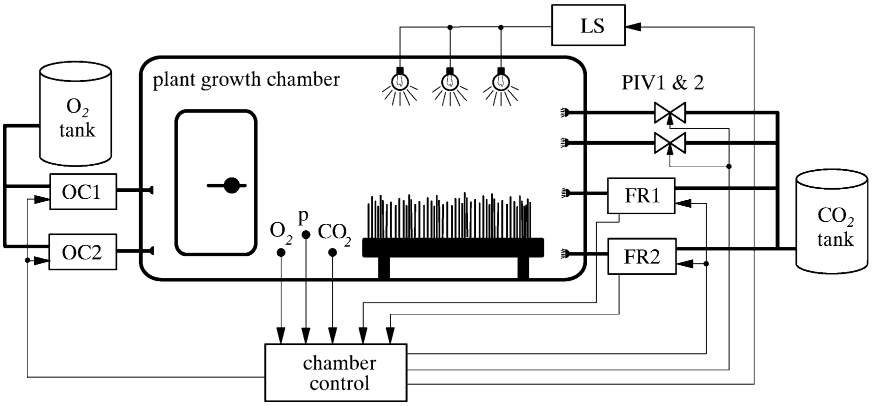

Fig. 18. BIO-Plex plant growth chamber.

B. Process Automation Example—BIO-Plex

Our major application is a simulation of 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 CO_2 and O_2 control in one plant growth chamber (PGC), as shown in Fig. 18.

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), the plant growth chamber (PGC), and redundant O_2 concentrators (OC1, OC2). The control system maintains a plant growth optimal CO_2 concentration of 1200 ppm and an O_2 concentration of 21% during the day phase of the system.

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 toward its closed position. This fault situation is difficult to capture by an explicit fault model, as we do not know, in advance, whether the regulator drifts toward its positive 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 system noise, is a typical candidate for our unknown-mode detection capability. In addition, we also provide explicit failure models that describe common situations. For example, the PGC has a plant growth area that is arranged in ten shelves, stacked in three columns – a large center stack and two small side stacks. We model five possible lighting failures in terms of a reduced illumination at the impaired shelves.

In the following we describe the outcome of a simulated experiment with two failures. A flow regulator fault with drifting symptom is injected at time point $k = 700$, and a light fault, which harms one side shelf, is injected at $k = 900$. The faults are "repaired" at $k = 1,100$ and $k = 1,300$ for the flow regulator fault and the lighting fault, respectively.

The simulated data is gathered from the execution of a refined and extended subset of NASA's JSC's CONFIG model for the BIO-Plex system [43], [44]. Hybrid estimation utilizes a cPHA model that consists of eight components, as shown in Fig. 19. In terms of the complexity of the hybrid estimation problem, we note that the concurrent automaton has approximately 450 000 modes, a number that is far beyond the scope of standard MM-estimation schemes.

Each mode describes the dynamic evolution of the chamber system by a fifth order system of difference equations. For example, the nominal operation of the plant growth can be characterized by the mode $\mathbf{x}_{d,k} = [m_{r2}, m_{r2}, m_{v1}, m_{v1}, m_{l2}, m_{p2}, m_{o1}, m_{o1}]$, where m_{r2} characterizes a partially open flow regulator, m_{v1} a closed pulse injection valve, m_{l2} all lights on, m_{p2} a plant growth mode at 1200 ppm, and m_{o1} a set of inactive O_2 concentrators, respectively. This mode specifies the raw model

$$\begin{aligned}
&F_1(m_{r2}) = \{x_{c1,k+1} = 0.5\, u_{c1,k},\ y_{c1} = x_{c1}\}\\
&F_2(m_{r2}) = \{x_{c2,k+1} = 0.5\, u_{c1,k},\ y_{c2} = x_{c2}\}\\
&F_3(m_{v1}) = \{w_{c2} = 0.0\},\ F_4(m_{v1}) = \{w_{c3} = 0.0\}\\
&F_5(m_{l2}) = \{w_{c1} = 9,028.5\}\\
&F_6(m_{p2}) = \Big\{ x_{c3,k+1} = x_{c3,k} + w_{c6,k} + w_{c7,k}\\
&\qquad x_{c4,k+1} = x_{c4,k} + \frac{10^6}{x_{c3,k}}\,[w_{c6,k} - f_1(x_{c4,k}, w_{c1,k})]\\
&\qquad x_{c5,k+1} = x_{c5,k} + \frac{100}{x_{c3,k}}\,[w_{c7,k} + f_1(x_{c4,k}, w_{c1,k})]\\
&\qquad w_{c6} = \frac{1}{44}(w_{c2} + w_{c3} + y_{c1} + y_{c2})\\
&\qquad w_{c7} = \frac{1}{32}(w_{c4} + w_{c5})\\
&\qquad y_{c3} = x_{c4},\ y_{c4} = x_{c5},\ y_{c5} = 18.178\, x_{c3} \Big\}\\
&F_7(m_{o1}) = \{w_{c4} = 0.0\},\ F_8(m_{o1}) = \{w_{c5} = 0.0\} \qquad (30)
\end{aligned}$$

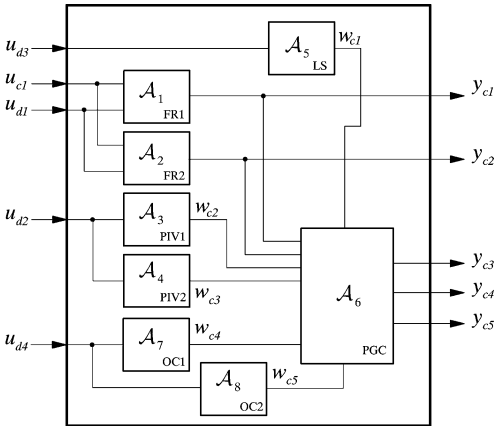

Fig. 19. BIO-Plex cPHA model.

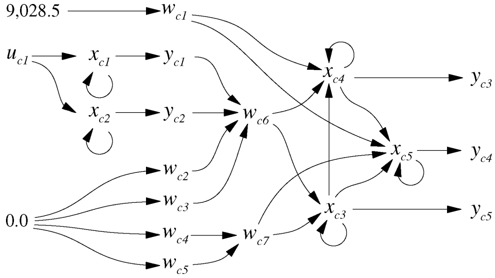

Fig. 20. Causal graph of the BIO-Plex cPHA raw model (30).

where f_1 denotes

$$f_1(x_{c4}, w_{c1}) := 2.323 \cdot 10^{-7}\, w_{c1} \left[72.0 - 78.89\, e^{-x_{c4}/400.0}\right]. \tag{31}$$

$x_{c1,k}$ and $x_{c2,k}$ denote the gas flow ([g/min]) of flow regulator 1 and 2, respectively, $x_{c3,k}$ captures the total number of gram-moles of gas in the chamber, $x_{c4,k}$ denotes the CO_2 concentration ([ppm]) and $x_{c5,k}$ denotes the O_2 concentration ([vol.%]) in the plant growth chamber. $w_{c1,k}$ denotes a multiplicative constant that captures the dependency between plant growth and the photo-synthetic photon flux of the lights above the plant trays. $w_{c2,k}$ and $w_{c3,k}$ denote the CO_2 gas flow ([g/min]) of the pulse injection valves and $w_{c4,k}$ and $w_{c5,k}$ denote the O_2 gas flow ([g/min]) to the oxygen concentrators. Equation (31) approximates the gas production/consumption rate due to photo-synthesis in the plants [43].

The causal graph (Fig. 20) of the raw model (30) leads to the decomposition that groups the components into a cluster with the first flow regulator ($\{\mathcal{A}_1\}$), one with the second flow regulator ($\{\mathcal{A}_2\}$) and one cluster for the remaining components, that is, the pulse injection valves, the lighting system, the chamber and the oxygen concentrators ($\{\mathcal{A}_3, \dots, \mathcal{A}_8\}$). 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 impose any constrain on the estimate for the remaining system.

Fig. 21 shows the continuous input (control signal) u_{c1} and the 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 toward its closed position, which it reaches at $k = 854$. The chamber control system reacts immediately and increases the control signal in order to keep the CO_2 concentration at 1,200 ppm. This transient behavior causes a slight bump in the CO_2 concentration, as shown in Fig. 22. The outcome of mode estimation is shown in Fig. 23 for the flow regulator and the lighting system. Our hybrid mode estimation system detects this unmodeled fault at $k = 706$ and declares flow regulator 2 to be in the unknown mode (we indicate the unknown mode by the mode number 0 in Fig. 23). The flow regulator mode *stuck-closed* (m_{r4}) becomes more and more likely as the regulator drifts toward its closed position. Hybrid mode estimation prefers this mode as symptom explanation from $k = 841$ onward, although the flow regulator goes into saturation a little bit later at $k = 854$.

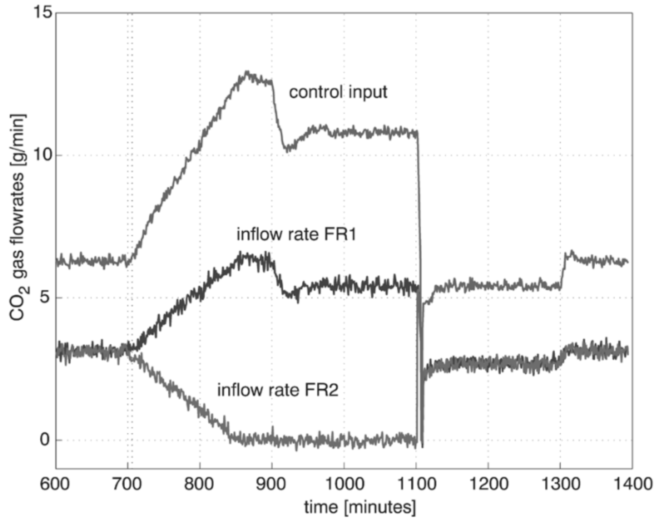

Fig. 21. Control input u_c and measured CO_2 input flow rates.

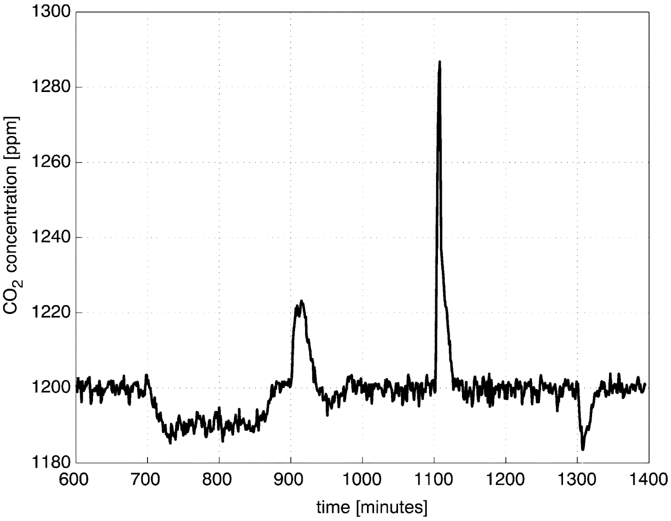

Fig. 22. CO_2 level in plant growth chamber.

The light fault at $k = 900$ is detected almost instantly at $k = 903$ (m_{l4}). However, lighting failure at one side stack

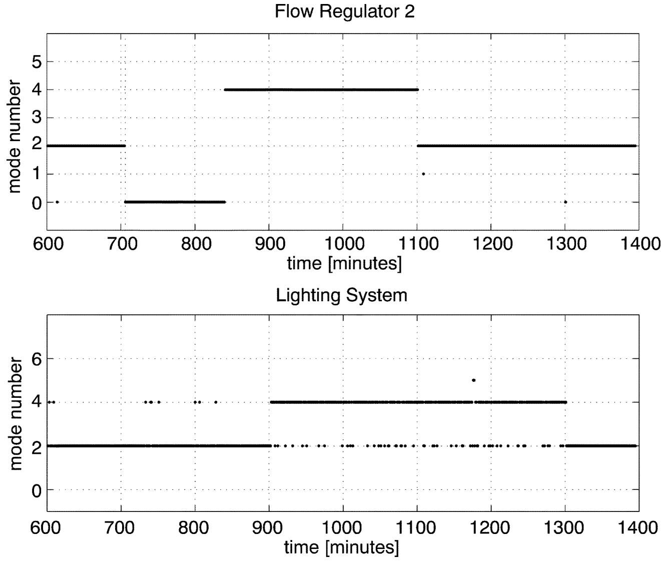

Fig. 23. Mode estimates for flow regulator 2 and lighting system.

leads to a slightly modified dynamic behavior that is hard to discriminate from the plant growth mode at full light. As a consequence, we obtain some mis-classifications. The good discrimination among the pre-specified modes (failure and nominal) is further demonstrated at the termination points of the faults. Repair of the flow regulator 2 and the lighting system are detected immediately at $k = 1,102$ and $k = 1,302$, respectively. Taking all components into account, we obtain the correct mode estimation at 91% of all time samples of this experiment.

This simplified real-world process automation problem also effectively demonstrates the benefit of our focused hybrid estimation scheme. We performed an estimation that maintained a fringe of the ten best estimates at each of the 795 time steps. This involved 496 809 filter executions. The majority of these filter operations used cached filters. Only 196 online deductions of extended Kalman filters were necessary for the experiment. Our focused hybrid estimation scheme tested, on average, 218 hypotheses to obtain the leading set of ten estimates. The worst-case performance experienced in the course of our experiment involved testing 2970 hypotheses. Compared to the 451 584 modes of the cPHA, this is equivalent to testing less than 0.05% or 0.7% of the possible hypotheses for the average and worst case, respectively.

VI. Conclusion

Classical hybrid estimation schemes, such as the family of MM estimation algorithms, do not scale up to the demanding estimation and diagnosis problems that arise in modern complex artifacts. This paper analyzed the shortcomings of MM estimation and presented an alternative method (HME) that can remedy this situation. HME builds upon prior work in model-based reasoning, in particular model-based monitoring, and reformulates the estimation task as a search problem that makes extensive use of probabilities and best-first enumeration. This allows HME to focus on highly probable hypotheses, without checking a prohibitively large number of unlikely hypotheses. The basic estimation algorithm was then extended with a novel clustering scheme for online filter deduction. Clustering improves the filtering performance and enables HME to cope with unforeseen situations in a degraded, but fail-safe manner. This property is particularly useful whenever the estimate is used for closed loop control, resulting in a versatile fault-tolerant control system.

Experimental evaluation with a simulated, multicomponent system demonstrated that HME provides similarly good estimation results, compared to MM estimation. However, it has the significant advantage in that it scales up to systems of high complexity, which are beyond the scope of classical multiple mode estimation algorithms. This advantage as well as the unknown mode and the multiple fault detection capability were demonstrated with a simulated process automation system.

One direction for future research is to utilize the continuous variant of *conflicts*, in order to further 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 [45] introduced continuous variants of conflicts, called *dissents*. We are currently reformulating these dissents for hybrid systems and investigating their incorporation into the underlying search operation. This will lead to an overall framework for hybrid estimation that unifies our previous work on Livingstone, Titan, Moriarty, and HME.

References

[1] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1974.

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

[3] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Automat. Contr.*, vol. AC-15, pp. 10–17, 1970.

[4] J. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, pp. 607–615, 1982.

[5] P. Hanlon and P. Maybeck, "Multiple-model adaptive estimation using a residual correlation Kalman filter bank," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, pp. 393–406, Apr. 2000.

[6] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 780–783, Aug. 1988.

[7] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 478–493, 1996.

[8] X. R. Li, X. Zhi, and Y. Zhang, "Multiple-model estimation with variable structure—Part III: Model-group switching algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, pp. 225–241, 1999.

[9] X. R. Li, "Multiple-model estimation with variable structure—Part II: Model-set adaption," *IEEE Trans. Automatic Control*, vol. 45, pp. 2047–2060, Nov. 2000.

[10] W. Hamscher, L. Console, and J. de Kleer, Eds., *Readings in Model-Based Diagnosis*. San Mateo, CA: Morgan Kaufmann, 1992.

[11] B. C. Williams and P. Nayak, "A model-based approach to reactive self-configuring systems," in *Proc. 13th Nat. Conf. Artificial Intelligence (AAAI-96)*, 1996.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst., Man, Cybern.*, vol. 4, pp. 100–107, 1968.

[13] B. C. Williams and R. J. Ragno, "Conflict-directed A* and its role in model-based embedded systems," J. Discrete Appl. Math., submitted for publication.

[14] R. Davis, "Diagnostic reasoning based on structure and behavior," *Artific. Intell.*, vol. 24, pp. 347–410, 1984.

[15] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artific. Intell.* , vol. 32, no. 1, pp. 97–130, 1987.

[16] ——, "Diagnosis with behavioral modes," in *Proc. IJCAI-89*, 1989, pp. 1324–1330.

[17] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Distributed monitoring of hybrid systems: A model-directed approach," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI'01)*, 2001, pp. 557–564.

[18] S. Narasimhan and G. Biswas, "An approach to model-based diagnosis of hybrid systems," in *Hybrid Systems: Computation and Control, HSCC 2002*, C. J. Tomlin and M. R. Greenstreet, Eds. New York: Springer-Verlag, 2002, vol. 2289, Lecture Notes in Computer Science, pp. 308–322.

[19] E. Benazera, L. Travé-Massuyès, and P. Dague, "State tracking of uncertain hybrid concurrent systems," in *Proc. 13th Int. Workshop Principles Diagnosis (DX02)*, May 2002, pp. 106–114.

[20] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proc. 7th Nat. Conf. Artificial Intelligence (AAAI'00)*, Austin, TX, 2000.

[21] S. McIlraith, "Diagnosing hybrid systems: A bayseian model selection approach," in *Proc. 11th Int. Workshop on Principles Diagnosis (DX00)*, June 2000, pp. 140–146.

[22] P. Li and V. Kadirkamanathan, "Particle filtering based likelyhood ratio approach to fault diagnosis in nonlinear stochastic systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 31, pp. 337–343, Aug. 2001.

[23] R. Dearden and F. Hutter, "The Gaussian particle filter for efficient diagnosis of nonlinear systems," in *Proc. 14th Int. Workshop Principles Diagnosis (DX03)*, Washington, DC, 2003.

[24] X. Koutsoukos, J. Kurien, and F. Zhao, "Estimation of distributed hybrid systems using particle filtering methods," in *Hybrid Systems: Computation and Control, HSCC 2003*, O. Maler and A. Pnueli, Eds. New York: Springer-Verlag, 2003, vol. 2623, Lecture Notes in Computer Science, pp. 298–313.

[25] S. Funiak and B. C. Williams, "Multi-modal particle filtering for hybrid systems with autonomous mode transitions," in *Proc. 14th Int. Workshop Principles Diagnosis (DX03)*, 2003.

[26] V. Verma, S. Thrun, and R. Simmons, "Variable resolution particle filter," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI'03)*, 2003, pp. 976–981.

[27] M. W. Hofbaur and B. C. Williams, "Mode estimation of probabilistic hybrid systems," in *Hybrid Systems: Computation and Control, HSCC 2002*, C. J. Tomlin and M. R. Greenstreet, Eds. New York: Springer-Verlag, 2002, vol. 2289, Lecture Notes in Computer Science, pp. 253–266.

[28] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 2001.

[29] M. W. Hofbaur, *Hybrid Estimation and its Role in Automation*. Graz, Austria: Faculty of Elect. Eng., Graz Univ. Technol., 2003, Habilitationsschrift.

[30] B. C. Williams, M. Ingham, S. Chung, P. Elliott, M. W. Hofbaur, and G. Sullivan, "Model-based programming of fault-aware systems," *AI Mag.*, vol. 24, no. 4, pp. 61–75, 2003.

[31] P. Maybeck and R. D. Stevens, "Reconfigurable flight control via multiple model adaptive control methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, pp. 470–480, May 1991.

[32] P. Struss and O. Dressler, "Physical negation: Integrating fault models into the general diagnostic engine," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI'89)*, 1989, pp. 1318–1323.

[33] R. E. Zippel, "The weyl computer algebra substrate," in *Design and Implementation of Symbolic Computation Systems*. New York: Springer-Verlag, 1993, vol. 722, Lecture Notes in Computer Science, pp. 303–318.

[34] M. W. Hofbaur and B. C. Williams, "Hybrid diagnosis with unknown behavioral modes," in *Proc. 13th Int. Workshop Principles Diagnosis (DX02)*, May 2002, pp. 97–105.

[35] P. Nayak, *Automated Modeling of Physical Systems*: Springer, 1995, Lecture Notes in Artificial Intelligence.

[36] L. Travé-Massuyès and R. Pons, "Causal ordering for multiple mode systems," in *Proc. 11th Int. Workshop Qualitative Reasoning (QR97)*, 1997, pp. 203–214.

[37] K. Reinschke, *Multivariable Control—A Graph-Theoretic Approach*. New York: Springer-Verlag, 1988, vol. 108, Lecture Notes in Control and Information Sciences.

[38] A. Gehin, M. Assas, and M. Staroswiecki, "Structural analysis of system reconfigurability," in *Preprints 4th IFAC SAFEPROCESS Symp.*, vol. 1, 2000, pp. 292–297.

[39] W. Kleissl, "Structural Analysis of Hybrid Systems," M.Sc. thesis, Inst. Automat. Control, Graz Univ. of Technol.,, Graz, Austria, 2002.

[40] A. Aho, J. Hopcroft, and J. Ullman, *Data Structures and Algorithms*. Reading, MA: Addison-Wesley, 1983.

[41] E. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, 2 ed. New York: Springer, 1998.

[42] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice*, 2 ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[43] 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 *Proc. SAE 28th Int. Conf. Environmental Systems*, Danvers, MA, July 1998.

[44] A. Hanford, "Advanced Life Support Baseline Values and Assumptions Document," NASA, Johnson Space Center, Houston, TX, Tech. Rep. CTSD-ADV-484, 2002.

[45] B. C. Williams and B. Millar, "Decompositional, model-based learning and its analogy to diagnosis," in *Proc. 15th Nat. Conf. Artificial Intelligence (AAAI-98)*, 1998.

Michael W. Hofbaur (M'99) received the Dipl.-Ing. and the Dr.techn. degrees in electrical engineering from Graz University of Technology, Graz, Austria, in 1995 and 1999, respectively.

Since 1999, he has been with the Institute of Automation and Control, Graz University of Technology. From 2000 to 2001, he was with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. His research interests include nonlinear and robust control, hybrid systems, and artificial intelligence.

Brian C. Williams received the B.S., M.S., and Ph.D. degrees in computer science and artificial intelligence from Massachusetts Institute of Technology (MIT), Cambridge.

He pioneered multiple fault, model-based diagnosis in the 1980s at the Xerox Palo Alto Research Center, Palo Alto, CA, and model-based autonomy in the 1990s through the Livingstone model-based health management and the Burton model-based execution systems. At the NASA Ames Research Center, from 1994 to 1999, he formed the Autonomous Systems Area and co-invented the Remote Agent model-based autonomous control system, for which he received a NASA Space Act Award in 1999. He was a member of the NASA Deep Space One probe flight team, which used remote agents to create the first fully autonomous self-repairing explorer, demonstrated in flight in 1999. He was a member of the Tom Young Blue Ribbon Team in 2000, assessing future Mars missions in light of the Mars Climate Orbiter and Polar Lander incidents. Upon joining the MIT faculty in 1999, he formed the Model-Based Embedded and Robotic Systems Group (MERS), focusing on using model-based autonomy and model-based programming technology to create cooperative robots that can perform agile and rapidly changing missions. He is currently also a member of the Advisory Council of the NASA Jet Propulsion Laboratory at California Institute of Technology, Pasadena.