

Distributed Diagnosis of Networked Hybrid Systems

James Kurien

Xenofon Koutsoukos

Feng Zhao

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 93404
jkurien,koutsouk,zhao@parc.com

Abstract

Networked embedded systems are composed of a large number of distributed nodes that interact with the physical world via a set of sensors and actuators, have their own computational capabilities, and communicate with each other via a wired or wireless network. Diagnostic systems for such applications must address new challenges caused by the distribution of resources, the networking environment, and the tight coupling between the computational and the physical worlds. Our approach is to move from centralized, discrete or continuous techniques toward a distributed, hybrid diagnosis architecture. This paper demonstrates distributed, discrete diagnosis algorithms that leverage the topology of the physical plant to limit inter-diagnoser communication and compute diagnoses in an anytime and any information manner, making them robust to communication and processor failures. It also presents a particle filtering based estimation algorithm that addresses the challenge of the interaction between continuous and discrete dynamics in hybrid systems. The distributed qualitative diagnosis and hybrid estimation techniques are demonstrated using a rocket propulsion system.

Introduction

Our diagnostic research is motivated by existing and emerging applications of networked, embedded systems. In such systems the physical plant is composed of a large number of distributed nodes, each of which performs a moderate amount of computation, collaborates with other nodes via a wired or wireless network, and is embedded in the physical world via a set of sensors and actuators. Examples include distributed sensor networks (Chu, Haussecker, & Zhao 2001), complex electromechanical systems with embedded controllers (Zhao *et al.* 2001), data networks, and smart matter systems (Jackson *et al.* 2001). Such systems present a number of interesting new challenges for diagnostic systems. A large amount of computation is potentially available, but it may be partitioned into relatively small, embedded chunks. Communication between nodes is available, but may involve unreliable delivery, power-constrained wireless networks, or large, complex topologies requiring multiple hops to connect two arbitrary nodes. Since the nodes are embedded in the physical world, its dynamics may be hybrid

and small deviations from expected behavior may accumulate over time (Williams, Hofbaur, & Jones 2002) or may be summed over multiple components (Zhao *et al.* 2001) before being detected and diagnosed.

Many existing diagnostic techniques are not well-matched to these challenges. In most model-based diagnostic techniques, prior knowledge about the physical plant consists of a description of the behavior of each component of the plant, including normal and faulty behaviors, and the interconnections between components (Hamscher, Console, & de Kleer 1992). These component descriptions are combined into a single, global store, observations are obtained from the physical plant, and a centralized algorithm is applied to find a system-wide diagnosis. These centralized approaches require that all sensor readings and component models from across a large, distributed system be forwarded to and interpreted by a central system which then returns diagnoses. As detailed in subsequent sections of this paper, this defeats the scalability and robustness advantages of distributing the plant. Many model-based diagnostic techniques, in addition to being centralized, are based upon a logical framework for diagnosis (de Kleer & Williams 1987) and are thus discrete. As such, they cannot resolve between and often cannot even detect failures that result in small continuous variations in the plant's behavior, nor can they provide sufficient resolution to enable compensatory control of continuous degradations in the plant. These limitations render such discrete techniques ill-suited for diagnosis and control of many embedded systems, as demonstrated in practical applications (Goodrich & Kurien 2001). Current FDI techniques (Frank, Ding, & Köppen-Seliger 2000) model continuous behavior, but cannot address the hybrid behavior exhibited by many physical systems, for example continuous processes coupled with digital controllers. They are also typically unable to cope with sensor faults, and are computationally expensive in that they rely on computing statistics over raw sensor signals in order to form a diagnosis. They are therefore practical for a relatively small number of fault hypotheses.

Our approach is to move from centralized, discrete or continuous techniques toward a distributed, hybrid diagnosis architecture. In an effort to accommodate the moderate local computation resources and hybrid behavior of embedded systems, the architecture leverages both fast, distributed,

discrete diagnosis and hybrid estimation. We first divide the discrete diagnostic model into a set of local diagnosers that mimic the topology of the physical plant. In decentralized diagnosis, e.g. (Debouk, Lafortune, & Teneketzis 2000), local diagnosers assemble a single global diagnosis via a coordination process, which is still subject to communication robustness and scalability issues. In distributed diagnosis, e.g. (Baroni *et al.* 1999), diagnosers collaborate without a centralized coordinator. This paper demonstrates distributed, discrete diagnosis algorithms that leverage the topology of the physical plant to limit inter-diagnoser communication and compute diagnoses in an anytime and any information manner, making them robust to communication and processor failures. These algorithms provide a fast, qualitative-level diagnosis of the physical plant which we then combine with more detailed hybrid estimation.

Hybrid estimation for an embedded system is particularly challenging because keeping track of multiple models and the autonomous transitions between them is computationally very expensive. The extension of conventional estimation techniques, like the Kalman filter, leads to algorithms that require tracking of all possible trajectories and therefore, are exponential in the number of time steps. Approximation by Gaussians is often used to collapse the distributions for each trajectory resulting in poor performance. A related approach to our work based on banks of extended Kalman filters has been presented in (Williams, Hofbaur, & Jones 2002) where only trajectories with high confidence probability are traced. Sequential Monte Carlo (or particle filtering) methods can support process densities that contain both continuous and discrete dynamics and have been explored for hybrid diagnosis in (McIlraith *et al.* 2000). However, autonomous transitions between modes triggered by the continuous dynamics have not been considered. Particle filtering has been applied also for a class of hybrid systems modeled by dynamic Bayesian networks in (Koller & Lerner 2001) where the autonomous transitions between discrete states are only defined using the so-called softmax conditional probability distributions. In this paper, we present a particle filter based estimation algorithm that addresses the challenge of the double-sided interaction between continuous and discrete dynamics in hybrid systems. We show how we can estimate autonomous transitions based on complex guard conditions and we describe how we can improve the performance and robustness of the algorithm by using guard conditions that cover the state space of the system. We demonstrate the algorithm for the estimation and fault detection of a rocket propulsion example.

The remainder of the paper is organized as follows. In the second section we describe a motivating hybrid diagnosis example, followed by a section outlining our diagnostic architecture that leverages both qualitative and hybrid diagnosis. The fourth and fifth sections describe the techniques we are exploring for distributed, qualitative diagnosis and hybrid diagnosis, respectively, including experimental results. The final section discusses variations for realizing the diagnostic architecture, along with challenges and advantages of each, and provides directions for future work.

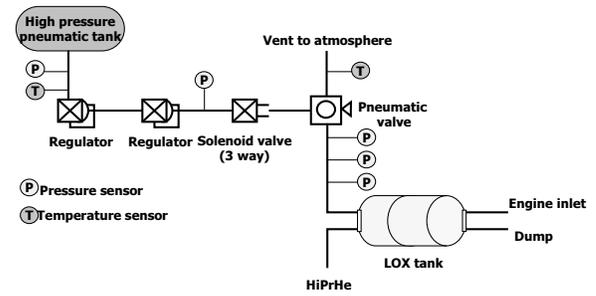


Figure 1: LOX Tank System

The Propulsion System Domain

In order to motivate our work in hybrid diagnosis, we chose to examine an application related to rocket propulsion. All space launch vehicles that reach Earth orbit do so by carrying large quantities of oxygen which is combined with a fuel and burned to produce thrust. The oxygen is stored in the form of liquid oxygen (LOX) at a temperature several hundred degrees below that of the launch environment.

Figure 1 illustrates the LOX venting system for the X-34, an experimental, rocket-powered vehicle designed for NASA. When the pneumatic valve is open, the LOX tank can vent to the atmosphere. The vehicle's control system does not directly actuate the pneumatic valve. Instead, the pneumatic valve opens when it is pressurized by the pneumatic system to its left. The pneumatic tank and regulators provide high pressure gas to the solenoid valve. When the control system opens the solenoid valve, the pneumatic valve is pressurized and opens. There are a wide variety of failures possible within this system. The pneumatic valve might fail to open because either of the valves is stuck closed, either of the regulators are too low, or the pneumatic tank is leaking. The pneumatic valve might stay open because it is stuck or the solenoid valve is stuck open, and similarly might open or close more slowly than originally anticipated. The LOX tank may also lose mass because the pneumatic valve is leaking, the LOX tank is leaking, or components downstream of the Engine Inlet, Dump or HiPrHe lines (not shown) are leaking. A slowly actuating pneumatic valve might be compensated for by the control system, whereas a leaking LOX tank is not recoverable and a potential safety hazard. Detection and diagnosis rely heavily upon estimating the mass of LOX and gaseous oxygen (GOX) in the tank, a task complicated by the fact that the LOX is modeled by a 10^{th} order hybrid system with non-linear dynamics and both commanded (venting, not-venting) and autonomous (boiling, not boiling) mode changes.

The Diagnostic Architecture

The challenge of diagnosing networked embedded systems is that they have both complex, hybrid dynamics and a relatively large number of components that can interact in a system-wide manner. The LOX tank pressure has a complex, hybrid relationship with the behavior of the pneumatic valve. The behavior of the pneumatic valve is influenced by

the state of all components in the pneumatic system. Qualitative techniques perform diagnostic inference involving multiple components in a computationally efficient manner, but they are limited by the low-resolution introduced by discretization of the continuous variables. For example a qualitative model cannot distinguish whether the pneumatic valve has a small leak, the valve is stuck completely open, or the temperature sensor that indicates the presence of LOX in the vent is faulty. Hybrid estimation techniques produce high-resolution state estimates that can distinguish between these failures, but they are computationally expensive and can be only used to detect faults that can be described by detailed analytical models. Thus diagnosis of networked embedded systems suggests collaboration between qualitative diagnosis and hybrid estimation. Figure 2 illustrates our conceptual hybrid diagnosis architecture for integrating these two techniques. Given a physical plant such as a spacecraft, we

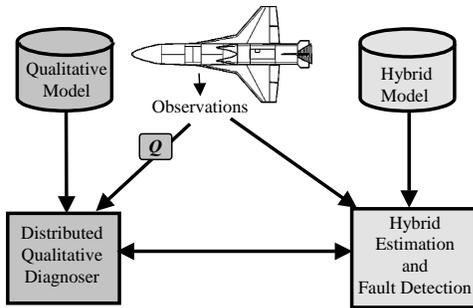


Figure 2: Diagnosis architecture

make use of models at two different levels of abstraction, a qualitative model and a hybrid model of the plant. By using both qualitative diagnosis and hybrid estimation and fault detection, we leverage the speed of the qualitative diagnoser and the resolution of the hybrid model.

The qualitative model provides a discrete abstraction of the plant model by first discretizing the range of each variable representing the system into a discrete, finite domain. For example, temperatures in the LOX system might be discretized into the range $\{low, high\}$ where *low* describes temperatures in the range of LOX (-184 degrees) and *high* describes temperatures around the ambient air temperature. A small software module, marked "Q" in the figure, maps continuous observations from the plant into the discrete space. The qualitative model specifies the states of the system and an abstraction of the dynamics. Given a set of observations in the discrete space, these qualitative relationships are sufficient to very quickly rule out many possible states of the plant as inconsistent with observations and yield the most likely, consistent states. For example, if the solenoid valve is commanded to close, we can predict the pneumatic valve should close and LOX flow through the vent should cease. Further observations might further reduce the set of consistent diagnoses. However, due to the loss of resolution introduced when the plant model is discretized, the qualitative diagnoser will not be able to distinguish failures that can be detected with a hybrid model.

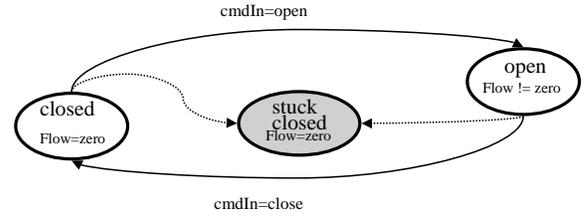


Figure 3: Automaton Representing A Single Valve

The hybrid model represents analytically the physical phenomena that govern the dynamic evolution of the plant. For example, the continuous states of the system include the temperature and mass of the GOX and LOX inside the tank and their evolution is governed by analytical equations derived by mass and energy conservation laws. It also relates the sensor measurements with the state of the system and models input and measurement noise. Controlled and autonomous events that affect the evolution of the system are also modeled. Hybrid estimation is the task of computing the most likely trajectory of continuous states given the observations. Estimation of the unobservable states is necessary to detect failures that are caused by subtle failures, for example, a leakage in the pneumatic valve. The next three sections of the paper describe our approach to distributed, qualitative diagnosis, our hybrid estimation and fault detection techniques, and options for integrating the capabilities of the two.

Distributed Qualitative Diagnosis

Our approach to distributed qualitative diagnosis follows the centralized diagnostic formalism developed in (de Kleer & Williams 1989) and extended in (Williams & Nayak 1996) and (Kurien & Nayak 2000). To motivate our distributed algorithms, we begin with a brief overview of the centralized technique, summarized from (Kurien & Nayak 2000). Suppose we would like to diagnose the state of a single component, a valve, which is qualitatively modeled via the finite state machine illustrated in Figure 3. We refer to each possible discrete state of a component as a *mode*. A valve v has three modes, *open*, *closed*, and *stuckClosed*. The behavior of the flow of the valve within each mode, which has the discrete domain $\{zero, nonzero\}$, can be captured with the following propositional formulae.

$$\begin{aligned}
 v = open & \Rightarrow flow_v = nonzero \\
 v = closed & \Rightarrow flow_v = zero \\
 v = stuckClosed & \Rightarrow flow_v = zero
 \end{aligned}$$

If $flow_v$ is observable from the physical plant, we will refer to this variable as an *observation*. In order to represent the non-determinism of the automaton within a propositional framework, the encoding introduces an *assumption* variable a . Intuitively, a_v represents the choice that Nature makes as to whether valve v will behave normally or experience a failure when it is commanded. The transition portion of the

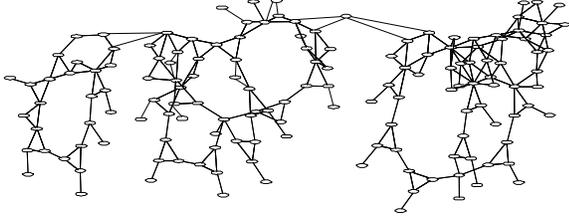


Figure 4: Variable Connectivity In a Global Model

automaton can thus be captured by the following formulae.

$$\begin{aligned}
a_v = normal &\Rightarrow \\
v_t = closed \wedge cmd_t = open &\Rightarrow v_{t+1} = open \\
v_t = closed \wedge cmd_t \neq open &\Rightarrow v_{t+1} = closed \\
v_t = open \wedge cmd_t = close &\Rightarrow v_{t+1} = closed \\
v_t = open \wedge cmd_t \neq close &\Rightarrow v_{t+1} = open \\
v_t = stuckClosed &\Rightarrow v_{t+1} = stuckClosed \\
a_v = stuck &\Rightarrow v_{t+1} = stuckClosed
\end{aligned}$$

Intuitively, the diagnostic task is to find a set of assignments to the assumptions, here $\{a_v\}$, such that the model is consistent with the observations, here $\{flow_v\}$. For example, suppose $v_t = closed$, we command the valve open, represented by $cmd_t = open$. The plant assigns O as $flow_v = zero$. The only consistent assignment to a_v is $a_v = stuck$ and we diagnose valve is stuck closed. If we wish to model multiple automata, we introduce a mode and assumption for each automaton and compile all automata into a set of formulae that may share variables. For example, two valves in series share the same flow. Figure 4 illustrates such a compilation for a small but typical physical plant model. Each node represents a variable. Two nodes are connected by an edge if the two variables appear in a formula together, and hence directly constrain each other. Note that a realistic model such as that of Figure 4 contains many observations and assumptions, and many assignments may be consistent. More formally, let A denote the set of assumptions, O denote the set of observations, and F denote the formulae describing the plant. Given an assignment Ω to O created by observing the plant, a diagnosis D is an assignment to A such that

$$\bigwedge_{a_i \in A} (a_i = d_i) \wedge_{o_j \in O} (o_j = \omega_j) \models F.$$

In this paper, we propose splitting the global diagnostic process into a number of cooperating local diagnostic processes. This will allow us to apply multiple processors to the diagnostic problem. It will also enable several properties we believe are even more important to scalability. First, we can arrange that two diagnosers need communicate only if the subsystems of the physical plant they correspond to are physically interconnected or share data. Thus our diagnostic architecture will have scalability properties similar to those of the physical topology of the system being diagnosed. Second, we can arrange that each diagnoser locally produce a superset of the diagnoses that a global diagnoser would produce for the local components, then use communication with other diagnosers only to refine its diagnosis. This will make the diagnostic architecture extremely robust to failure and

able to operate in an anytime and any information manner. If diagnosers fail, then the remaining diagnosers will simply produce coarser (more conservative) estimates of the possible states of their components. If the system is bifurcated due to a communication failure, then each half will produce all diagnoses consistent with the reachable diagnosers and any state of the other half of the system. Finally, the ability to coordinate multiple localized diagnosers introduces the possibility of making use of a heterogeneous set of diagnostic algorithms, so long as a coordination protocol has been established. We believe these properties will be of particular utility as we investigate applications involving very large numbers of embedded processors communicating via networks.

In order to distribute the problem, we divide the global diagnoser which produces assignments to A into a set of local diagnosers which make assignments to subset of A . Intuitively, we partition the edges of Figure 4. If a node is connected to edges in more than one partition, it is replicated and the partitions must reach consensus on its value. More formally, a local diagnoser L is described by $(F_L, V_L, A_L, O_L, R_L)$ where F_L is the subset of F assigned to L , V_L denotes the set of variables that appear in F_L , A_L denotes $A \cap V_L$, O_L denotes $O \cap V_L$ and R_L denotes the union of $V_L \cap V_M$ over all other diagnosers M . Given a fixed number of diagnosers, we can use a graph partitioning algorithm (Sanchis 1989) to find a partitioning of the graph that attempts to minimize R_L for each diagnoser.

We define the relationships *conservative* and *feasible* between the diagnoses produced by the global diagnoser and the diagnoses produced by a local diagnoser. A local diagnosis set D_L is conservative with respect to the global diagnosis set D_G if $\forall \delta_G \in D_G \Pi_{A_L}(\delta_G) \in D_L$, where Π is the projection operator. That is, the assignments made to the assumptions local to L by a global diagnosis must also be made by a local diagnosis. A local diagnosis set D_L is feasible if the assignments made to the local assumptions are contained in a consistent global diagnosis. More formally, $\forall \delta_L \in D_L \exists \delta_G \in D_G : \Pi_{A_L}(\delta_G) = \delta_L$.

We are developing several strategies for distributed qualitative diagnosis. In this paper we discuss the 'bottom-up' approach, similar in spirit to Waltz's algorithm (Waltz 1975). Each diagnoser begins with a conservative local diagnosis set. Each set is monotonically reduced toward a feasible set as a side effect of spreading consensus on the value of shared variables. The algorithm has the following high-level steps. Details of achieving distributed consensus and leveraging parallel communication have been omitted for the sake of brevity.

1. Given observation set Ω , if $o_j \in O_L$, assign $o_j = \omega_j$ in L .
2. $\forall L$, if $O_L \neq \emptyset$, compute all assignments to $A_L \cup R_L$ s.t. $\bigwedge_{o_j \in O_L} (o_j = \omega_j) \wedge_{a_i \in A_L} (a_i = d_i) \wedge_{r_i \in R_L} (r_i = \rho_i) \models F_L$
3. For each $r \in R_L$, for each other diagnoser M , if $r \in V_M$ send all R_L assignments to M .
4. In each such M , compute all assignments such that $\bigwedge_{r_i \in R_L} (r_i = \rho_i) \wedge_{a_k \in A_M} (a_k = d_k) \wedge_{r_k \in R_M} (r_k = \rho_k) \models F_M$
5. If the consistent R_M assignments decreased in step 4, return to step 3, substituting M for L .

The algorithm operates by incrementally reducing the possible assignments to A_L for all L , first by introduction of observations and second by communication between diagnosers. In Step 1, observations are assigned in every diagnoser which has constraints involving an observation. In Step 2, the observation assignments are used to compute all assignments to $A_L \cup R_L$ that are consistent with F_L and the observations received by L . Note that the projection of A_L from these assignments is a conservative diagnosis set. Intuitively, suppose an assignment to A_L appears in a global diagnosis but is not computed by L . If it is not computed, it must be inconsistent with F_L and the assignments to O_L . It is therefore inconsistent with F and the assignments to O , and could not appear in a global diagnosis. In Step 3, the assignments to R_L are projected out of the consistent assignments of L and forwarded to each other diagnoser M that references these variables. In Step 4, M eliminates a subset of its assignments that are not feasible. Intuitively, an assignment α to A_M is not feasible if there is no assignment to A containing α that is consistent with F and O . If α constrains a variable in R_L to have a value that was not received from L , then α is inconsistent with all consistent assignments to A_L . Thus, each time Step 4 is performed, infeasible assignments to A_M are eliminated. Each diagnoser begins with a conservative set of assignments to A_L , and as rounds of communication are performed, the local diagnoses are moved toward feasibility in an anytime manner. Per Step 5, the algorithm continues as long as consistent assignments are eliminated. In the worst case, each loop would eliminate one of an exponential number of possible assignments.

We performed preliminary experiments with one possible implementation of this distributed algorithm, using using finite-state automata to prune inconsistent assignments to V_L (Steps 2 and 4) and a distributed consensus algorithm (Steps 3 and 5) shown to converge to feasible diagnoses (Su *et al.* 2002). Table 1 compares performance with L2, a centralized diagnoser available from NASA. The first three columns are the name of the diagnostic scenario, the diagnoses found by L2, and the time required. The physical plant has few sensors, so the number of consistent diagnoses grows with the complexity of the scenario. The fourth column is the number of local diagnosers reached via Step 3 of the algorithm, out of 24. The distributed diagnoser never computes global diagnoses, but the fifth column lists the number of combinations of local diagnoses. Intuitively, each local diagnosis is a projection of a global diagnosis, but not every combination of local diagnoses is a consistent global diagnosis. Note that even given the global diagnosis set, each node in a distributed system without a centralized controller could act upon a different diagnosis. We therefore do not believe that feasible local diagnoses are a significant issue for the types of embedded systems we envision. However, one could rule out inconsistent combinations by performing a linear-time, unit-propagation on each combination. The fifth column is the time to compute the diagnoses, demonstrating the dramatic speed advantage of computing feasible local diagnoses via a pre-compiled FSA representation versus global, on-line inference. The current implementation runs each diagnoser serially on a single processor, and we believe a parallel im-

Independent Faults In	L2		Distributed		
	Diag	Time	Spread	Diag	Time
First module	6	0.02	9	21	0
Two modules	12	0.18	14	49	0
Three modules	84	13.28	20	343	0.05
All modules	108	27.08	24	637	0.22

Table 1: Comparison of distributed diagnoser and L2

plementation will provide a greater speed advantage.

Hybrid Estimation and Fault Detection

In this section, we use particle filtering based methods for state estimation of hybrid systems. Hybrid systems contain interacting discrete and continuous dynamics. The discrete dynamics are usually described by discrete event models with a finite state space. Every discrete state (or mode) corresponds to a unique differential/difference equation that governs the continuous dynamics. Mode transitions may occur either upon receiving an external control command or when the continuous state satisfies certain guard conditions. Mode transitions that depend on the continuous behavior of the system are called autonomous. The main idea in our algorithm is to focus on the mode transitions that cover most of the probability space. Of course, the probability of each mode transition changes dynamically based on the continuous behavior of the system and has to be re-computed at every time step.

A hybrid system is described by $H = (Q, X, \Sigma, I, Inv, E, f)$ where Q is the set of discrete states or *modes* of the system, $X = \mathbb{R}^n$ is the continuous state space, Σ is a finite set of transition labels or *events*, $I \subseteq Q \times X$ is the set of initial conditions, $Inv : Q \rightarrow 2^X$ is the invariant associated with each mode q , $E \subset Q \times X \times \Sigma \times Q \times X$ is the set of discrete transitions, and $f : Q \times X \rightarrow X$ is the flow condition for every mode.

The state of the hybrid system is described by $s = (q, x)$. The state can change either by a discrete transition or by a time delay. A discrete (or mode) transition may change both the mode and the continuous state, while a time delay changes only the continuous state according to the flow condition. Each transition consists of a source mode q_i , a target mode q_j , a labeling event σ (denoted as $q_i \xrightarrow{\sigma} q_j$), a guard set $G_{ij} \subset X$, and a reset map $R_{ij}(x) = x'$. If the condition described by the guard is satisfied, then the transition can fire. Upon firing of the transition, the continuous state may be reset according to the reset map.

In the hybrid system literature, it is often assumed that the state is directly observable. However, in real-world applications, the state has to be reconstructed from the observations. The continuous dynamics of the system can be represented by the discrete-time model

$$\begin{aligned} x_{t+1} &= f_q(x_t) + \nu_t \\ y_t &= h_q(x_t) + v_t \end{aligned}$$

where ν_t and v_t denote process and measurement noise respectively. It should be noted that the estimation algorithm can be applied to systems with time-varying dynamics as

well as continuous control inputs. A discrete transition occurs when either the controller issues an appropriate command or when the continuous state satisfies the guard of the transition. The hybrid estimation problem is to compute the most likely hybrid state $s_t = (q_t, x_t)$ given the observation sequence $Y_t = (y_0, y_1, \dots, y_t)$ and the history of control events $(\sigma_1, \sigma_2, \dots)$ up to time t .

The most challenging aspect of every hybrid estimation algorithm is how to monitor the autonomous mode transitions in order to use the appropriate mode q for updating the belief of the continuous state x . The probability of mode transitions triggered by control commands can be usually computed by discrete estimation techniques based, for example, on Hidden Markov models. Let's focus on autonomous transitions and define the mode transition probability matrix with elements

$$T_{ij}(x_{t-1}) = p(q_t = j | x_{t-1}, q_{t-1} = i), \quad i, j = 1, \dots, |Q|.$$

Let G_{ij} be the guard corresponding to the transition from mode i to mode j . Assuming that the probability of the transition $q_i \rightarrow q_j$ is equal to the probability the guard G_{ij} is satisfied, we have

$$T_{ij}(x_{t-1}) = \int_{G_{ij}} p(x_{t-1} | Y_{t-1}, q_{t-1} = i) dx_{t-1} \quad (1)$$

where $p(x_{t-1} | Y_{t-1}, q_{t-1} = i)$ is the conditional density of the continuous state at time $t - 1$.

The probability of occurrence of the autonomous transitions is represented by the transition probability matrix that can be computed at every time step as a function of the continuous state. The estimation algorithm will be robust if small changes in the continuous state do not result in large changes in the probabilities T_{ij} . Practically, it is desirable to (1) avoid chattering phenomena, where the probability mass oscillates between modes at every time step, and (2) allow enough time after a mode change for the transient to converge to the steady state behavior for that particular mode. These aspects of the algorithm can be considerably improved by transforming the guard conditions so that they form a cover of the continuous state space as explained in the following.

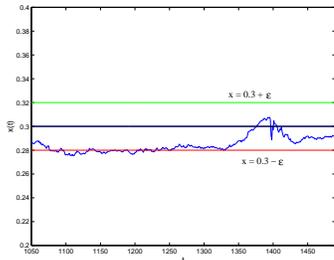


Figure 5: Guard conditions that cover the state space

Figure 5 shows the estimated fluid level $x(t)$ of a two tank system. This is typical when the continuous state is not directly observed but inferred from the sensor measurements. Let's assume that the system has two modes and switches

from q_1 to q_2 if $x(t) > .3$ and from q_2 to q_1 if $x(t) < .3$. Such a guard condition arises if we connect two tanks with a pipe at $h = .3$. Further details about the actual system are not important. In the estimation algorithm, $x(t)$ is approximated at every time step with a probability distribution. If the transition probability matrix T is computed using the original guard conditions, the performance of the algorithm is degraded by the fast switching around $t = 1380$ and leads to chattering between modes q_1 and q_2 . While the most likely discrete state oscillates between q_1 and q_2 , the estimation of the continuous state is unreliable.

Hybrid estimation can be considerably improved by transforming the guard conditions to form a cover of the state space as illustrated in Figure 5. The transition $q_1 \rightarrow q_2$ occurs if $x(t) > 0.3 + \epsilon$. Similarly, the transition $q_2 \rightarrow q_1$ occurs if $x(t) < 0.3 - \epsilon$. The small variations of the state $x(t)$ around $x(t) = 0.3 - \epsilon$, for example, will not trigger any transitions since the system is not in mode q_2 . The design parameter ϵ depends on the process and measurement noise. The transition probability matrix can be represented by the transformed guard conditions by equation (1). It should be noted that the continuity of analog-to-digital maps based on covers of the state space has been studied using small topologies in (Nerode & Kohn 1993).

We have implemented a particle filtering based algorithm for hybrid estimation that computes the mode transition probability matrix at every time step and updates the belief of the continuous state using the most likely mode. In particle filters, complex integrals as that of equation (1) are computed efficiently by approximating the belief state by finitely many samples. Detailed descriptions of particle filtering methods can be found in (Doucet, Freitas, & Gordon 2001). Our approach is similar to algorithms with mixed-state and automatic model switching that have been successfully applied for tracking of motion boundaries in video images (Isard & Blake 1998; Black & Fleet 2000).

Let $\{s_{t-1}^{(k)}, w_{t-1}^{(k)}, k = 1, \dots, N\}$ denote the sample set at time $t - 1$ where $s_{t-1}^{(k)} = (q_{t-1}^{(k)}, x_{t-1}^{(k)})$ is the k^{th} sample of the hybrid state and $w_{t-1}^{(k)}$ its probability weight. The estimation algorithm consists of the following steps:

1. Initialization $t = 0$.

- i. sample $s_0^{(k)}, k = 1, 2, \dots, N$ from $p(q_0), p(x_0)$ and set $t = 1$.

2. Prediction

- i. sample from $p(s_t | s_{t-1}^{(k)})$ to select each $\tilde{s}_t^{(k)}$.
- ii. evaluate the importance weights $w_t^{(k)} = p(y_t | \tilde{s}_t^{(k)})$.
- iii. normalize the weights.

3. Resampling

- i. resample N particles $s_t^{(k)}$ from $\tilde{s}_t^{(k)}$.
- ii. set $t \leftarrow t + 1$ and go to step 2.

Consider that at time t the prediction $p(q_{t-1}, x_{t-1} | Y_{t-1})$ is represented by the sample set $\{q_{t-1}^{(k)}, x_{t-1}^{(k)}, w_{t-1}^{(k)}, k = 1, \dots, N\}$. The mode transition probabilities can be computed by

$$T_{ij}(x_{t-1}) = \begin{cases} \frac{\sum_{k \in \tilde{G}_{ij}} w_{t-1}^{(k)}}{\sum_{k \in \tilde{I}} w_{t-1}^{(k)}} & i \neq j \\ 1 - \sum_{\ell \neq i} T_{i\ell}(q_{t-1}, x_{t-1}) & i = j \end{cases}$$

where $k \in \hat{G}_{ij} \Leftrightarrow q_{t-1}^{(k)} = i \wedge x_{t-1}^{(k)} \in G_{ij}$ and $k \in \hat{I} \Leftrightarrow q_{t-1}^{(k)} = i$. Let $(q_{t-1}^{(k)}, x_{t-1}^{(k)}, w_{t-1}^{(k)})$ be the k^{th} particle and assume $q_{t-1}^{(k)} = i$, then we sample from the i^{th} row of the mode transition probability matrix $[T_{i1}, T_{i2}, \dots, T_{i|Q|}]$ to select the k^{th} sample $q_t^{(k)}$ for the discrete mode. Suppose that $q_t^{(k)} = j$, then the density $p_{ij}(x_t|x_{t-1}^{(k)}) = p(x_t|x_{t-1}^{(k)}, q_{t-1} = i, q_t = j)$ is used to sample in order to compute the k^{th} sample $x_t^{(k)}$ for the continuous state. Next, we compute that importance weights, normalize, reinforce the predicted state using the observations, and resample the particles as described in the above algorithm.

The estimated discrete and continuous states are computed using the samples by

$$\hat{q}_t = \arg \max_i \sum_{k \in \hat{Q}_i} w_t^{(k)} \text{ and } \hat{x}_t = \frac{\sum_{k \in \hat{Q}} w_t^{(k)} x_t^{(k)}}{\sum_{k \in \hat{Q}} w_t^{(k)}}$$

where $\hat{Q}_i = \{k | q_t^{(k)} = i\}$ and $\hat{Q} = \{k | q_t^{(k)} = \hat{q}_t\}$.

The particle filtering based algorithm can be used for fault detection using an observer-like scheme as shown in figure 6. The particle filter algorithm plays the role of a hybrid observer which is computing the most likely discrete mode \hat{q} and continuous state \hat{x} and is generating the expected output \hat{y} based on the plant model. The residual signal $r_t = y_t - \hat{y}_t$ is then thresholded, after low-pass filtering, to detect possible failures. Fault detection and isolation can be performed by considering both the residual r_t and the mode \hat{q}_t . For example, the observer may not be able to perfectly track fast transients after each mode transition and therefore, the residual exceeding the threshold immediately after a mode transition does not necessarily correspond to a fault. Also information about the modes for which the discrepancy is present can be used for fault isolation. A leakage in the pneumatic valve of propulsion system, for example, will cause a discrepancy only if the valve is closed. In the following, we present simulation results for the propulsion system for two scenarios (1) normal behavior, and (2) leakage in the pneumatic valve.

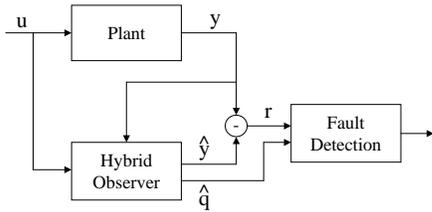


Figure 6: Fault detection using hybrid observer

Simulation Results We have tested extensively the above algorithm for state estimation of the propulsion system domain. Here, we present simulation results for a subsystem of the propulsion domain consisting of the LOX tank and the pneumatic valve. This subsystem interacts with the pneumatic part only via the command that controls the solenoid valve that in turn, drives the pneumatic valve. The subsystem is best described as a hybrid system with two discrete

modes corresponding to the oxygen boiling or not boiling in the tank. The continuous dynamics of the subsystem are described by a set of 4th nonlinear differential equations that are discretized using a sampling period $T = 100ms$. The discrete modes correspond to the oxygen boiling or not, which is determined by a nonlinear guard of the form $P_{sat} \geq P_{GOX}$ where the saturation pressure P_{sat} is approximated using a 5th order polynomial of the LOX temperature and the GOX pressure P_{GOX} that depends on the GOX mass and temperature using the ideal gas law. The outputs are the GOX pressure and temperature and are contaminated with Gaussian noise.

Normal behavior. We have demonstrated that the algorithm can track the state in the case when there are no faults in the system. The continuous states corresponding to the LOX and GOX masses and the discrete mode are shown in figure 7. The expected venting pressure as computed using the estimated state is plotted versus the actual venting pressure is also shown. The simulation was performed using 100 particles in approximately 2500s on a PC using MATLAB.

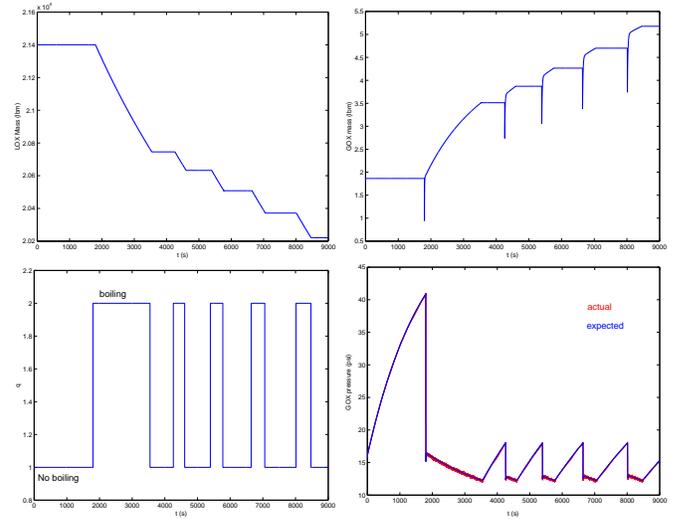


Figure 7: Estimated continuous state, mode, and output

Pneumatic valve leakage. The estimation algorithm can be used also to detect continuous faults such as leakage in the pneumatic valve. The valve leakage was simulated by including an additive term in the equation that represents the flow balance when the pneumatic valve is closed. Figure 8 shows the expected and the actual venting pressure. The estimated discrete mode and the residual signal computed as the difference between the actual GOX pressure and the expected are also shown. Whenever there is no boiling then the actual pressure is less than the expected one and a fault is detected.

Discussion and Future Work

We propose two methods for leveraging the combination of qualitative diagnosis and hybrid estimation. In the first, the two techniques are run in parallel, so that a single run of the

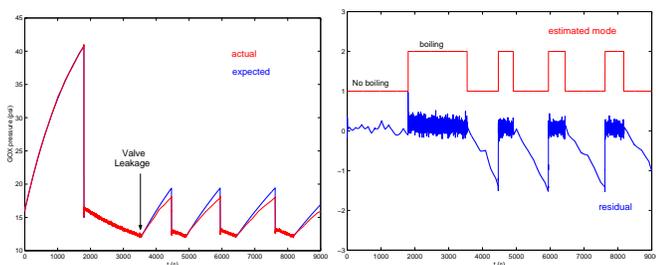


Figure 8: Estimation in the case of leakage

hybrid observer is used to resolve diagnoses that the qualitative diagnoser could not resolve alone. This scheme has the advantage that observations are incorporated into the qualitative diagnoser and hybrid estimator as soon as they are generated by the plant. Thus the hybrid model is selected before the qualitative diagnoses are determined. As a result the hybrid estimation may not be useful in updating the probability estimates of all qualitative diagnoses. For example, the behavior of the system after an abrupt fault such as sudden loss of pressure in the pneumatic tank is not described adequately by the nominal analytical model. In such cases, we may still be unable to distinguish all qualitative diagnoses. The second alternative is to generate multiple hybrid models by parameterizing the fault hypotheses. The hybrid model has the expressive power to parameterize both abrupt failures and subtle degradation of components, see for example (Koutsoukos *et al.* 2001). Given a large set of fault hypotheses, the qualitative diagnoser can be used to focus the hybrid observer on a relatively small set of possible failures that will allow estimation based on the particle filter. For each qualitative diagnosis, we invoke a copy of the hybrid observer with the appropriate fault parameters and biases to its model, for example representing the increased belief that a specific valve is leaking. In this scheme the hybrid estimator is used to refine every qualitative diagnosis. However, the hybrid estimation does not begin until after a fault in the plant has become visible in the qualitative abstraction used by the qualitative diagnoser. Depending upon the dynamics of the plant, this could represent a significant time delay and a significant drawback. We are currently investigating variations of these two techniques in search of a method of integration that is both mathematically sound and practical.

Acknowledgment This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract F33615-99-C3611. Rong Su implemented the distributed diagnoser as a PARC intern. NASA Ames Research Center provided L2 and collaborated with PARC on HCC, simulation software used for the LOX system.

References

- Baroni, P.; Lamperti, G.; Pogliano, P.; and Zanella, M. 1999. Diagnosis of large active systems. *Artificial Intelligence* 110(1):135–183.
- Black, M., and Fleet, D. 2000. Probabilistic detection and tracking of motion boundaries. *Int'l J. of Comp. Vision* 38(3):231–245.

Chu, M.; Haussecker, H.; and Zhao, F. 2001. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int'l J. High Performance Computing Applications*. To appear.

de Kleer, J., and Williams, B. C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32(1):97–130.

de Kleer, J., and Williams, B. C. 1989. Diagnosis with behavioral modes. In *Proc. of IJCAI-89*, 1324–1330.

Debouk, R.; Lafortune, S.; and Teneketzis, D. 2000. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications* 10(1/2):33–86.

Doucet, A.; Freitas, N. D.; and Gordon, N., eds. 2001. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer.

Frank, P.; Ding, S.; and Köppen-Seliger, B. 2000. Current developments in the theory of FDI. In *4th IFAC SAFEPROCESS*, 16–27.

Goodrich, C., and Kurien, J. 2001. Continuous measurements and quantitative constraints - challenge problems for discrete modeling techniques. In *Proc. of iSAIRAS-2001*.

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

Isard, M., and Blake, A. 1998. A mixed-state condensation tracker with automatic model switching. In *Proc. of the 6th International Conference on Computer Vision*, 107–112.

Jackson, W.; Fromherz, M.; Biegelsen, D.; Reich, J.; and Goldberg, D. 2001. Constrained optimization based control of real time large scale systems: Airjet movement object system. In *Proc. of the 40th IEEE CDC*, 4717–4720.

Koller, D., and Lerner, U. 2001. Sampling in factored dynamic systems. In Doucet *et al.* (2001). 445–464.

Koutsoukos, X.; Zhao, F.; Haussecker, H.; Reich, J.; and Cheung, P. 2001. Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems. In *Proc. of the 40th IEEE CDC*, 793–801.

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

McIlraith, S.; Biswas, G.; Clancy, D.; and Gupta, V. 2000. Hybrid systems diagnosis. In Lynch, N., and Krogh, B., eds., *Hybrid Systems: Computation and Control*, vol. 1790, LNCS. Springer. 282–295.

Nerode, A., and Kohn, W. 1993. Models for hybrid systems: Automata, topologies, controllability, observability. In Grossman, R. L.; Nerode, A.; Ravn, A. P.; and Rischel, H., eds., *Hybrid Systems*, vol. 736, LNCS. Springer-Verlag. 317–356.

Sanchis, L. A. 1989. Multiple-way network partitioning. *IEEE Transactions on Computers* 38(1):62–81.

Su, R.; Kurien, J.; Koutsoukos, X.; and Wonham, W. M. 2002. Distributed diagnosis for qualitative systems. Technical Report SPL-01-071, Palo Alto Research Center.

Waltz, D. L. 1975. Understanding line drawings of scenes with shadows. In Winston, P. H., ed., *The Psychology of Computer Vision*. McGraw-Hill. 19–91.

Williams, B. C., and Nayak, P. P. 1996. A model-based approach to reactive self-configuring systems. In *Proc. AAAI-96*, 971–978.

Williams, B. C.; Hofbaur, M.; and Jones, T. 2002. Mode estimation of probabilistic hybrid systems. MIT AI Memo.

Zhao, F.; Koutsoukos, X.; Haussecker, H.; Reich, J.; Cheung, P.; and Picardi, C. 2001. Distributed monitoring of hybrid systems: A model-directed approach. In *Proc. IJCAI'2001*, 557–564.