An integrated approach to diagnosis of complex hybrid systems

Sriram Narasimhan, Gautam Biswas, and Gabor Karsai

Institute for Software Integrated Systems, Dept. of EECS, Box 1679 Station B
Vanderbilt University, Nashville, TN 37235. USA.
nsriram, biswas, gabor@vuse.vanderbilt.edu

ABSTRACT

This paper presents a model-based approach to diagnosis of hybrid systems. We have developed a combined qualitative-quantitative diagnosis scheme that uses hybrid models of the system and a model of the supervisory controller. By applying the supervisory controller model to diagnostic analysis we significantly cut down on the complexity in tracking behaviors, and in generating and refining hypotheses across discrete mode changes in the system behavior. We present the algorithms for hybrid diagnosis: hypotheses generation by back propagation, and hypotheses refinement by forward propagation and parameter estimation. Example scenarios demonstrate the effectiveness of this approach.

Keywords: hybrid systems, supervisory controller, tracking hybrid behavior, fault isolation, qualitative and quantitative methods.

1. INTRODUCTION

Modern systems, such as aircraft and manufacturing plants, are complex and include supervisory control that switches modes of behavior of the system to increase reliability and improve performance. Consider a plant with a supervisory controller in Fig. 1. Actuators directly controlled by the supervisory controller govern the system input, and sensors measure system variables that are used to estimate the system state.

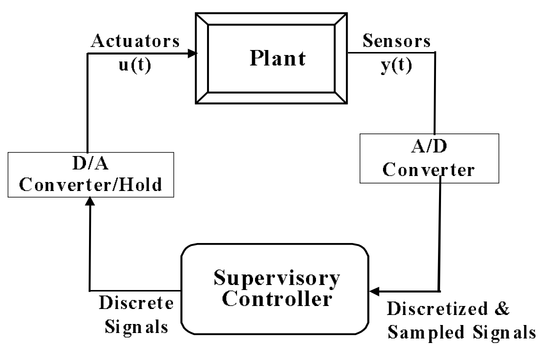

Figure 1: Hybrid system: Plant with supervisory controller

The supervisory controller is a software program running on a digital processor. Unlike lower-level regulators in feedback loops, this controller is not tightly meshed with the continuous plant dynamics. It maintains predefined system functionality by generating discrete actions at pre-determined points in time, and when predefined events occur in the plant dynamics. Variables values are directly sensed or computed from the measurements made on the system. The discrete actions of the controller change the input to the plant, or cause a reconfiguration of the plant. This changes the models that govern the continuous dynamics of the plant. We call such a system (plant + supervisory controller) a *hybrid* system, where the continuous behavior of the plant is interspersed by discrete changes triggered by the supervisory controller.

The continuous dynamics of the plant are defined by differential and algebraic equations:

$$\dot{x}(t) = f(x(t),u(t),q(t))$$
$$y(t) = g(x(t),u(t),q(t)),\ t >= 0,$$

where $x(t)$ is the continuous state vector, $u(t)$ is the input, $y(t)$ is the output vector, and $q(t)$ is the discrete *mode*. Mode changes in the plant are attributed to *controlled* and *autonomous* events [2]. The supervisory controller may change the discrete mode resulting in changes to the $u(t)$, $f(.)$, and $g(.)$ functions. This is called a *controlled event*. The discrete mode, $q(t)$, may also change when the state variables, *x(t)*, cross boundary values, which brings about a change in *f(.)* and *g(.)*. These are called *autonomous events*, typically attributed to modeling abstractions [9].

We study the fault detection and isolation (FDI) problem in hybrid systems with supervisory controllers. System faults may be component, actuator, sensor, and controller faults. When the controller issues commands that generate behavior in conflict with the desired functionality, the controller may be said to be faulty. We do not deal with these kinds of faults and make the assumption that the commands issued by the controller are consistent with the desired functionality. This paper develops a model-based methodology that combines qualitative and quantitative reasoning techniques to perform parameterized fault isolation of plant component faults.

2. MODELING FOR DIAGNOSIS

Model-based approaches to FDI in hybrid systems with supervisory control uses explicit models of the plant and controller to track system behavior and detect and analyze faults.

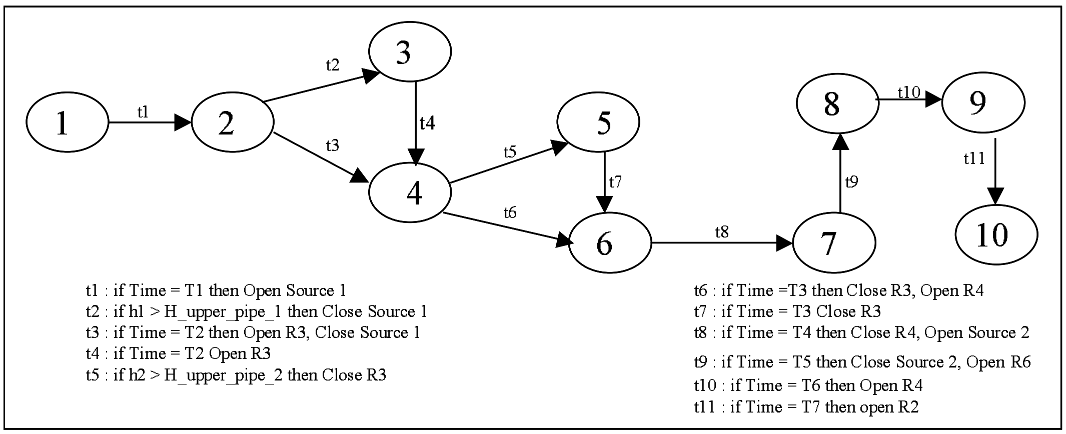

Figure 2: Controller mode for Three-tank system

2.1 Controller Model

The primary model of the controller is implemented as a timed finite state machine (FSM). States of the FSM correspond to the states of the controller. A state trajectory (sequence of states visited) of the controller defines the mode of the physical plant. The transitions determine the conditions for switching states and specify the discrete actuation signal generated when the transition is executed. As is illustrated in Fig. 2, transition conditions can be based on timing information, controller-defined signals, and expressions that are a function of the plant variables. The transitions also describe the initialization of variables in the new state based on values from the previous state (the reset function). They define a partial order of discrete

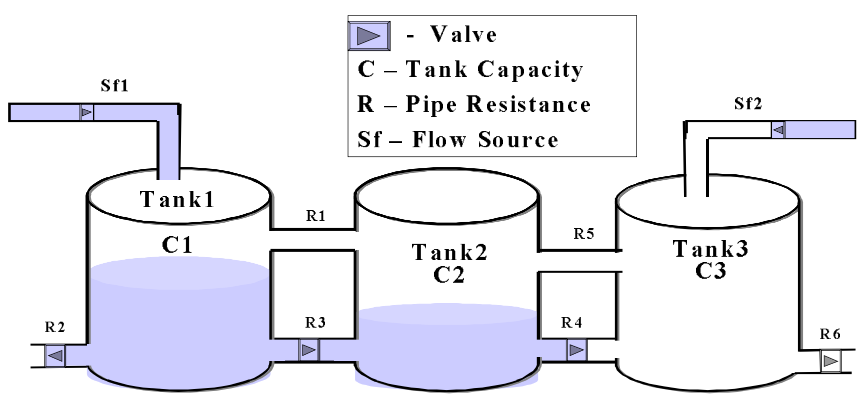

Figure 3: Three-tank system

events that may occur in the system. Fig. 2 illustrates a supervisory controller model for a connected three-tank system shown in Fig. 3. For the physical plant, tanks 1 and 3 can be filled and emptied independently. The supervisory controller directly controls the actuators that open and close the valves on the pipes, therefore, it takes the system through a defined sequence of modes. Autonomous events occur when liquid levels attain particular heights, which cause the intermediate connecting pipes to become active or inactive. Autonomous events may occur between controlled events.

2.2 Plant Model

Our approach to modeling the plant involves building hybrid automata that model the continuous and discrete parts in a unified framework [1]. We use a *hybrid bond graph* modeling paradigm for the hybrid automata. Bond graphs adopt an energy-based lumped parameter approach to modeling of physical systems. At any given point in time, the physical system is described as a distribution of energy over connected physical elements. This energy distribution reflects the history of the system, and, therefore, defines its *state*. Future behavior is determined by its current state description, and subsequent *input* to the system. Changes in state of a physical system are attributed to energy exchange among its components, which can be expressed in terms of the time derivative or flow of energy, *power*. Irrespective of domain (i.e., mechanical, fluid, pneumatic, electric, etc.) power is the product of two conjugate variables: effort, e (e.g., force, pressure, and voltage), and flow, f (e.g., velocity, flow rate, and current). The primary constituent elements in the bond graph modeling language are energy storage elements, *capacitors*, C, and *inductors*, I, *and a dissipative* element, R. These elements exchange energy with other elements via *ports*. To connect more than two basic elements together, a *junction* structure is required. Junctions typically allow an arbitrary number of components to be connected together. They preserve continuity of power by adhering to the generalized forms of Kirchoff's voltage and current laws, which define the two forms of junctions, *0*- and *1*-junctions, respectively.

In the bond graph framework, discontinuities in behavior are dealt with at a *meta-model* level, where the energy model embodied in the bond graph scheme is suspended in time, and discontinuous model configuration changes happen instantaneously. Therefore, the meta-model describes a control structure that causes changes in bond graph topology using idealized switches that do not violate the principles of energy distribution in the system imposed by the bond graph. After a new model configuration is derived, the model state is transferred from the previous configuration to the new one. Further switches may occur, and the meta-model is active until a bond graph configuration where no more switches occur is derived. At this point, the principles of conservation of energy and continuity of power govern the evolution of continuous system behavior. To keep the overall behavior generations consistent, the meta-model control mechanism and the energy-related bond graph models are kept distinct. The configuration changes are implemented as local structural changes, where model components get connected or disconnected at junctions controlled by the meta-model mechanism. The switching structure is implemented as *controlled junctions* that facilitate the modeling of discrete mode transitions in system behavior [7]. This provides a compact representation of the system model across all its nominal modes of operation. Instead of pre-enumerating the bond graph for each mode, the hybrid bond graph uses individual junctions to model local mode transitions. The controlled 0- and 1- junctions represent idealized discrete switching elements that can turn the corresponding energy connection on and off. A finite state machine determines the ON/OFF physical state of the junctions. The transitions in this automaton depend on both control signals and internal variable values.

Fig. 4 illustrates the hybrid bond graph model of the three-tank system shown in Fig. 3. The two flow sources into tanks 1 and 3 are indicated by Sf1 and Sf2, respectively, the tank capacities are shown as C1, C2, and C3, and the pipes are modeled by simple resistances, R1 through R6. Valves are modeled by controlled junctions, which are shown in the figure as junctions with subscripts. The control signals for turning these junctions on and off are generated by the finite state automata shown in Fig. 4. The toggling signal for the automata comes directly from the supervisory controller. For autonomous transitions in the system, also modeled by controlled junctions, the transition conditions computed from system variables (e.g., see the transition conditions for junction 6). A mode in the system is defined by the state of the eight controlled junctions in the hybrid bond graph model. Therefore, theoretically the system can be in 256 different modes.

When all the switch conditions are specified, the hybrid bond graph reduces to a simple bond graph. All controlled junctions whose switch condition is OFF are replaced by corresponding 0 source elements. The resulting bond graph may be used to systematically derive state equation and temporal causal graph (TCG) models of the system. State equations and TCGs can be systematically derived from the bond graph representation of the system [8,14]. When mode changes occur, the appropriate controlled junctions are toggled, a new bond graph model is derived corresponding to the current system configuration, and a new state equation model and TCG can be derived for this mode. The state equations simulate system behavior, and they along with the temporal causal graphs constitute our diagnosis models.

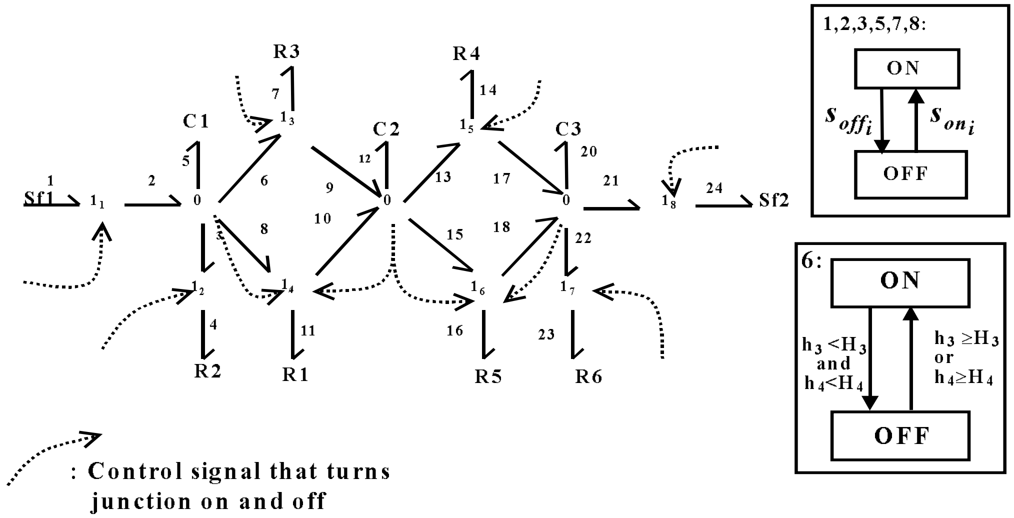

Figure 4: (a) Hybrid Bond Graph for Three Tank System
(b): Automata for the Switched Junctions

3. OUR METHODOLOGY FOR HYBRID DIAGNOSIS

Our model-based approach (Fig.5) uses a hybrid observer to track normal system behavior, a fault detection mechanism, and a fault isolation unit. The observer uses a quantitative hybrid model of the plant to follow the dynamics of the plant within a continuous operating mode. Mode transition conditions defined by the model are checked, and when they are satisfied, the

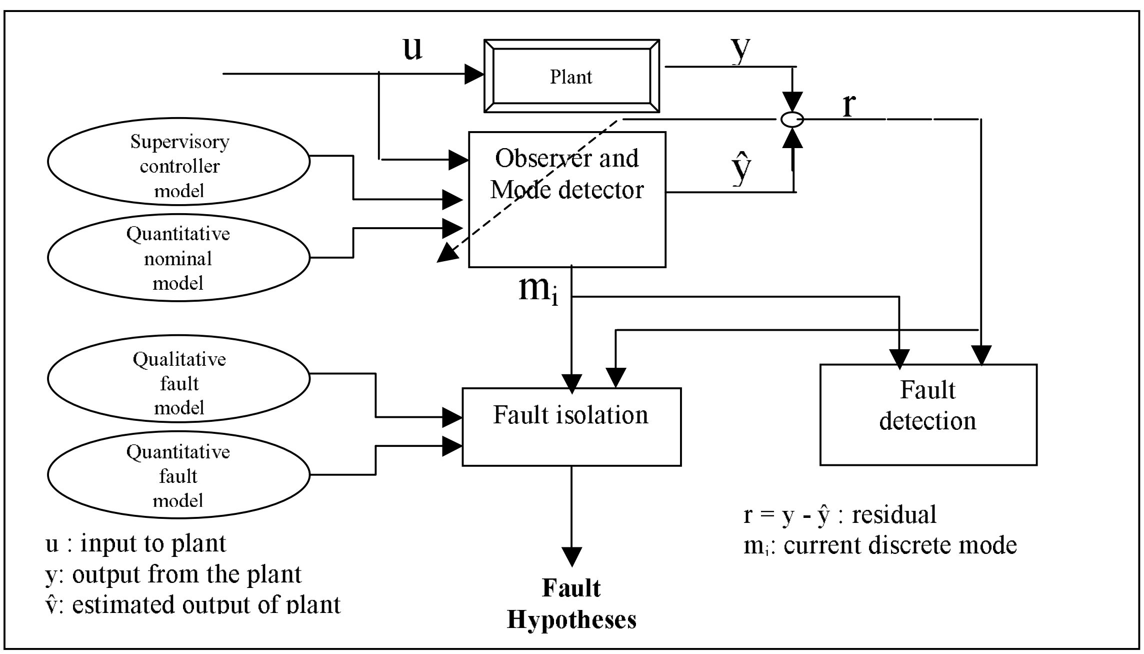

Figure 5: Hybrid System FDI Architecture

observer makes the switch from the current to the new mode of system operation by updating the plant model and its continuous system vector. We have adopted a combination of a hybrid automata and the Kalman filtering approach to design our hybrid observer [12]. Small differences, attributed to minor imperfections in the model and noise in the measurements, are compensated for in the observer mechanism. Significant differences that the observer cannot compensate for cause the fault detection unit to signal the presence of a fault in the system. In most cases, noise and the complexity of the signals, and the imperfections in the sensors and the system model, require the use of sophisticated signal analysis techniques to detect discrepancies in the observed measurements [5]. The fault isolation unit generates candidate faults and refines them with the hybrid model and measurements from the system.

4. OBSERVER

The schematic of our hybrid observer scheme based on the hybrid bond graph models describe earlier is shown in Fig. 6. To accommodate modeling uncertainties and noise in measurements, we propose the use of a Kalman filter scheme to track continuous behavior within a mode of operation. For a given state space equation model a Kalman gain matrix, K, the enhanced state space equations using a Kalman filter is given by,

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y})$$
$$\hat{y} = C\hat{x} + Du$$
$$\dot{P} = AP + PA^T + BQB^T - KRK^T$$
$$K = PC^T R^{-1}$$

where A, B, C, and D are the system matrices, Q is the input noise covariance, R is the output noise covariance, and P is the error covariance matrix. A mode switch requires the re-computation of the model and the Kalman gain matrix. To avoid a lag in tracking, we propose pre-computation of the models and the initial gain matrix for potential successor modes from the current mode. Potential next modes can be determined by analyzing the controller model and looking at the direction of change of the state variable values to predict which autonomous transitions may occur next. The combination of the two gives us all possible candidates for mode transitions. It should be noted that as the behavior evolves in the current mode, more autonomous transitions might become candidates. If such a situation occurs, then the new models and the Kalman gain matrix are computed for the destination mode based on the new autonomous transition.

Fig. 7 illustrates a sample run of our hybrid observer as it tracks the three-tank system through three modes. In the first mode

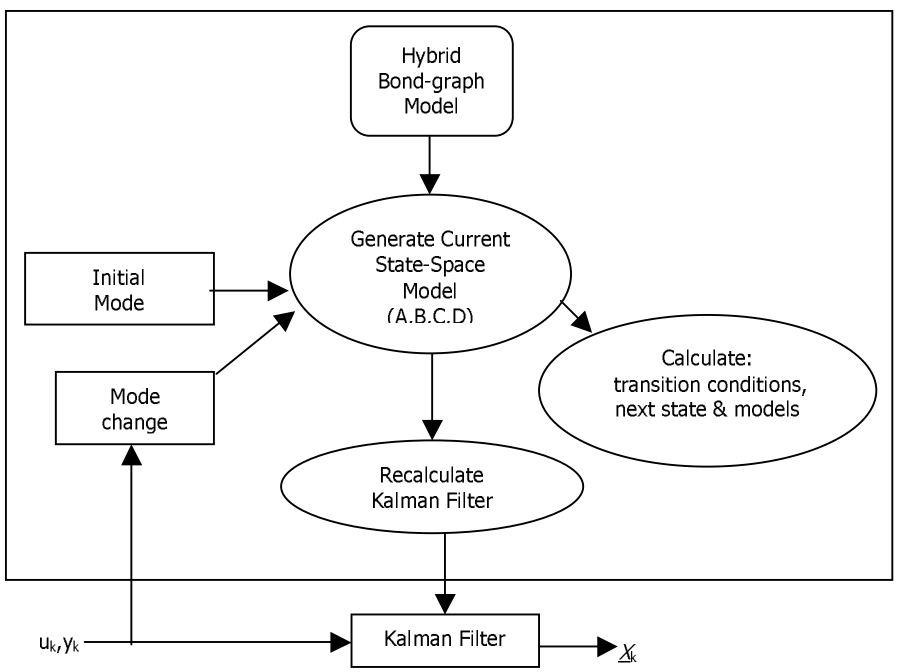

Figure 6: Hybrid observer

(0-10 seconds in Fig. 7), there is an inflow to tank 1, and R3 and R4 are open (all three tanks filling up). In the second mode (10-20 seconds in Fig. 7), there is no inflow to tank 1 and R2 through R6 are open (i.e., all three tanks are being drained). In the third mode (20-30 seconds in Fig. 7), there is no inflow to tank 1, and only R3 is open (isolating tank 3). We assume an input noise and output noise covariance of 1% for each variable. Fig. 7 displays the actual and tracked heights in the three tanks over time. It is interesting to note that the Kalman filters accurately track the system behavior through and across modes 1 and 2. In mode 3, there is an abrupt change in flow value (because of the closing of valve 4). As a result, the predicted level values for tanks 1 and 2 are initially inaccurate, but as time progresses, the Kalman filter converges to its true value. This implies that mode transitions with abrupt changes can cause initial problems in tracking system behavior. This can become a problem in systems with quick autonomous transitions, because the hybrid observer may make errors in predicting mode changes in the system behavior.

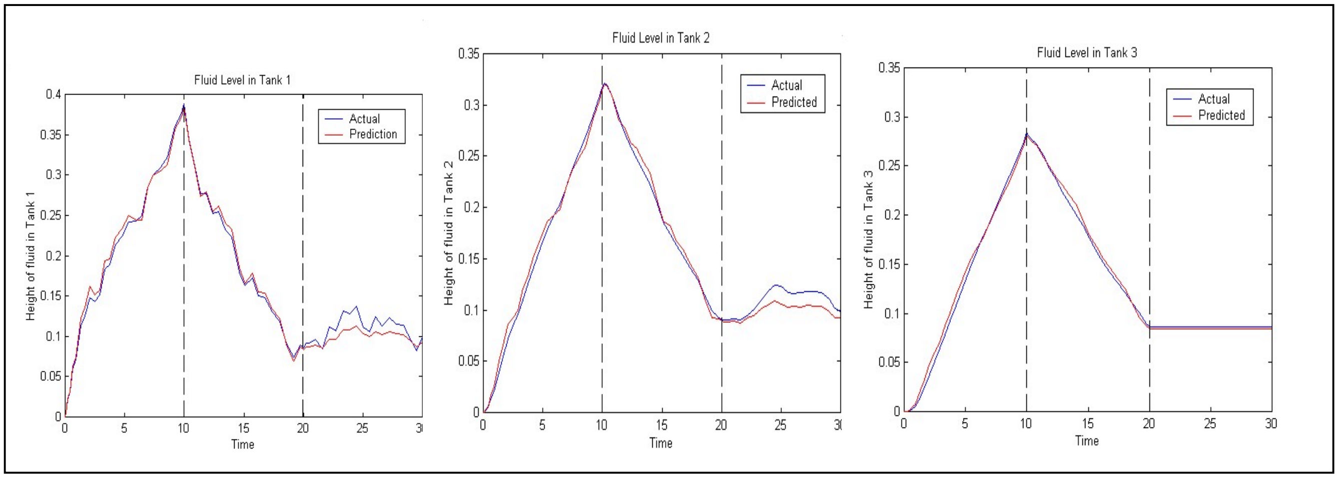

Figure 7: Sample run of hybrid observer

5. FAULT ISOLATION

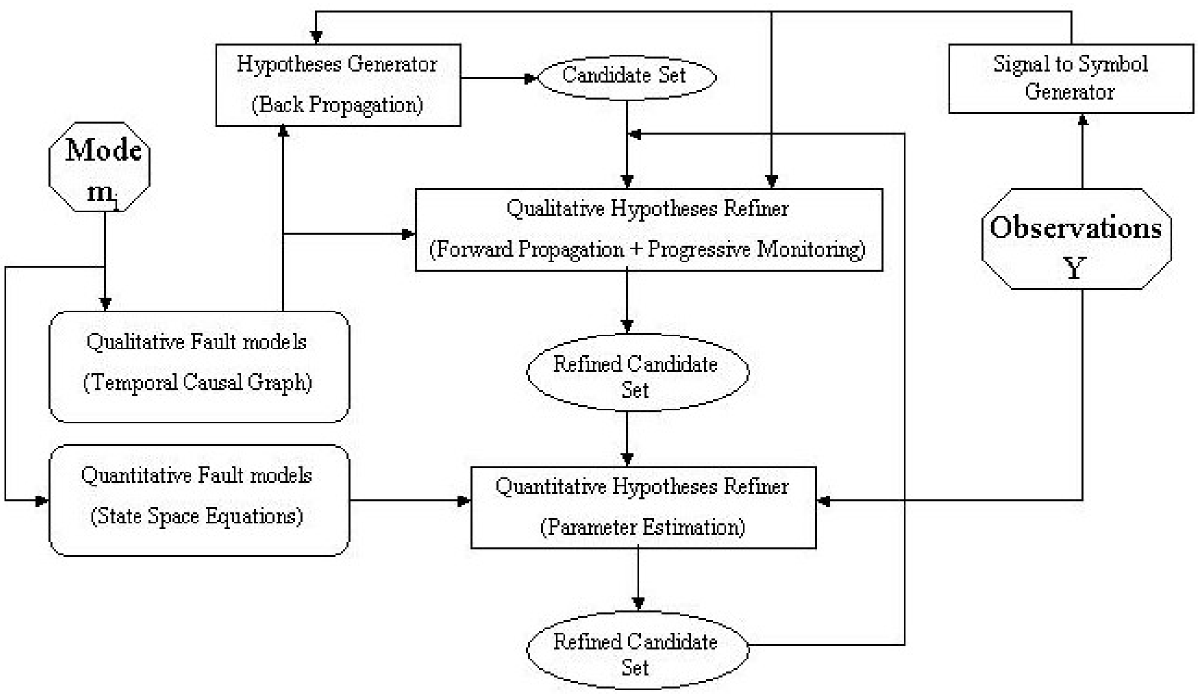

Fig 8: Fault Isolation Scheme

The type of plant model employed determines the scheme to be employed for fault isolation. Discrete event approaches precompile fault models and fault trajectories into finite state automata for tracking nominal and faulty system behavior [4,15]. Traditional fault observer schemes in the continuous domain use structured and directional residual approaches. An algebraic function transforms the raw residual generated by the observer scheme into a form where there is a one-to-one mapping between hypothesized faults and the observed residual vectors [13]. These techniques mainly apply to linear systems, though recently there have been some applications to non-linear dynamic systems [3]. Extending these continuous methodologies to hybrid systems becomes intractable because the residual transformation functions have to be pre-computed for all modes of operation. Further, when faults occur, predicting the true system mode in itself becomes a challenging task. The fault isolation problem becomes even more complex, when the fault is detected in a later mode of operation. The predicted mode sequence may no longer be the true mode sequence the system goes through after the occurrence of the fault. Additional methods have to be introduced for detecting mode transitions, switching the system model when such transitions occur, and correctly initializing the system state so that the fault observers perform correctly. Mode changes introduce discrete effects and transients making it difficult to analyze fault transients across mode changes. Therefore, extending continuous FDI schemes to hybrid systems is a non-trivial task.

Our approach to fault isolation involves two primary steps: *hypotheses generation* and *hypotheses refinement*. We use a qualitative framework for hypotheses generation, and a combined qualitative-quantitative approach for hypothesis refinement. Qualitative analysis requires a symbol generation methodology (SIGNAL_TO_SYMBOL) presented in [5]. The use of qualitative methods to initiate hypotheses generation and refinement overcomes a number of the limitations of quantitative schemes, such as convergence and accuracy problems in dealing with complex non-linearities and lack of precision of parameter values in system models. We use a combination of extended versions of the backward and forward propagation algorithms [8] that operate on the TCGs representing the system dynamics to generate fault hypotheses and predict their consequences over time as fault signatures. When used in conjunction with the supervisory controller model, this can provide a computationally simpler mechanism to track faulty behavior across discrete mode changes. The qualitative reasoning scheme is fast and effective, but it has limited discriminatory ability. To overcome this, we use parameter estimation schemes that initiate a fault observer for every hypothesized fault [6]. The use of a reduced set of individual observers mitigates computational convergence and complexity problems of quantitative analysis methods. In addition, the quantitative estimate provides detailed information about the extent of degradation in faulty components.

We use an example to explain our hypotheses generation and refinement algorithms. Fig 9. illustrates the TCG model that is used in the qualitative hypotheses generation and hypotheses refinement algorithms. In all our experiments, system behavior is tracked with a sampling rate of 0.1 second. In this example, our measurements are the height of liquid in tank 3, i.e., h_3 (this corresponds to pressure e11 in the bond graph) and out flow from pipe 1, i.e., f_{R1} (this corresponds to flow f3 in the bond graph). Fig. 10 displays the results of tracking the level of fluid in tank 3 using the observer algorithm. We introduced a fault in the system at time 20 (mode 9) but the fault detection unit detects the fault only at time 21 (mode 10). At this point the mode trajectory seen so far (1,2,3,4,5,6,7,8,9,10) is saved and the fault isolation algorithms are invoked.

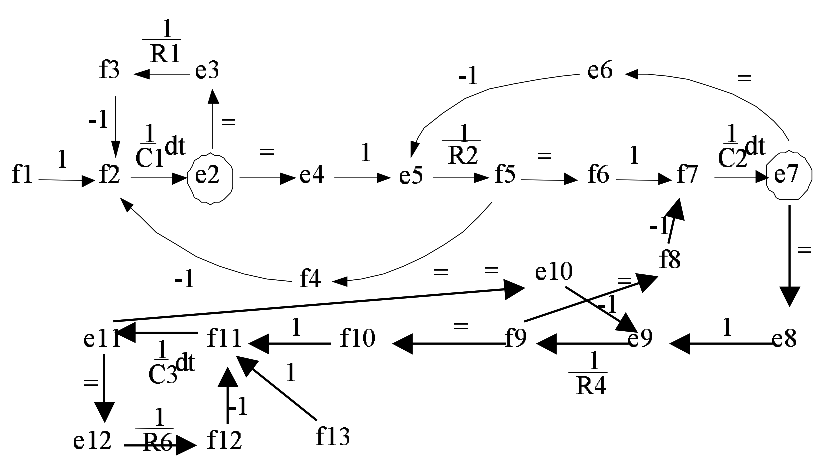

Figure 9: Temporal causal graph for the three tank system

4.1 Hypotheses Generation

Back propagation for initial hypotheses generation has to be performed across modes the system has traversed through because the fault may have occurred in a previous mode but the manifestations are not seen until a later mode. This may happen when none of the observed variables are affected in the mode in which the fault occurs. The problem is that once a fault occurs the predicted mode sequence of the observer may no longer be correct, and a worst-case analysis may require considering all possible modes in generating fault hypotheses. However, the assumption that the controller model is correct implies that *the observer predicted the correct mode sequence till the fault occurred. Therefore, the mode in which the fault occurred must be in the predicted trajectory of the observer*. Back propagation is applied to each of the modes in the mode trajectory predicted by the observer. This ensures that the true fault hypothesis, which includes the fault and the mode in which the fault occurred (<mode, fault>) will be included in our initial hypothesis set.

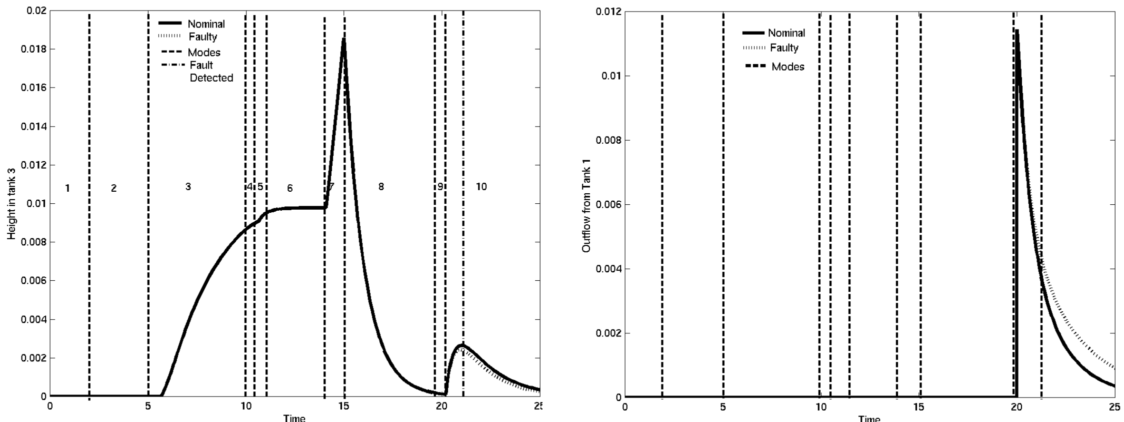

Figure 10: Experiment 1 – fault C2- introduced in mode 9, but detected in mode 10. h_3 and f_{R1} are measured. Mode changes in the system are shown by vertical lines

The hybrid back propagation algorithm goes back through the mode trajectory, and in each mode the continuous back prop algorithm [12] is applied to the corresponding TCG to identify the initial candidates. For the example, given discrepancy h3+, the hybrid backprop algorithm yields the candidates <10, C3- R6+ R4+ C2- R2- C1- R1+ > (10 indicates the mode number) in mode 10, and additional initial candidates <9, C3- R6+ R4+ C2- > and <8, C2- R4+> in modes 8 and 9. Candidates in the earlier modes are not listed.

4.2 Hypotheses Refinement

Hypotheses refinement uses qualitative forward propagation in TCG to generate fault signatures for every hypothesized fault [8]. A progressive monitoring scheme is applied to match the fault signature with the current observations. At some point, the hypothesis refinement step switches to quantitative parameter estimation in an attempt to isolate the true fault.

In the qualitative step, the forward propagation for the hypothesized faults is first performed in the mode in which the fault occurred. The progressive monitoring algorithm is used to check if the signatures match the actual observations. If the qualitative predictions of the fault signature do not match the current observations, we have to consider one of two possible situations. Either the fault candidate is invalid or a mode change has occurred in the system. To take into account both possibilities, *all possible mode changes from the current mode are hypothesized. This involves using the model of the controller plus additional information from the system to limit the number of possible mode transitions*. For each of the hypothesized modes, the forward propagation algorithm is repeated to generate the qualitative fault signatures. If the predicted signatures still do not match the observations, further mode transitions are considered. When applied a number of times, this may generate a large number of trajectories that one has to track. If the discrepancies persist for a period of time then the corresponding candidate is dropped from the list. Otherwise, quantitative parameter estimation is initiated to continue the tracking process.

In the hybrid hypothesis refinement algorithm, all candidates generated by the back prop algorithm are maintained in a list. The list is multidimensional because multiple trajectories may be initiated for a given fault candidate. Our goal is to continue to run this algorithm until the size of this list is reduced to the desired size (typically 1). It may turn out that the progressive monitoring technique quickly prunes the list to a small number of candidates, but then is unable to reduce it any further. In other situations, the numbers of trajectories associated with the fault hypotheses start growing. The limited discriminative ability of the qualitative analysis [6] requires us to switch to quantitative parameter estimation to uniquely isolate the fault. This approach works within a single continuous mode. If there is a mode switch during the estimation process, we face two problems. (i) How we do know that a mode change has occurred, and if a mode change has occurred what is the new mode? It depends on the estimated parameter value. (ii) Even if we can identify the new mode of the system, how do we continue parameter estimation since the state space model for the new mode is different? In our work we perform parameter estimation in a single mode. When a mode change occurs we switch the mode and start the estimation afresh.

In the parameter estimation algorithm, for each candidate we derive the state space equations of the system in the current mode. The only unknowns in the equations are the parameters corresponding to the selected candidate. A system identification approach is used to estimate the fault parameter using the known input and measured values of the output. Finally a statistical test is carried to check to see if the estimation process has converged. This can be done by predicting the output using the estimated parameter value, and then checking to see of the prediction error is statistically close to 0.

Continuing with the example, the 2nd order signatures for the fault hypotheses derived by forward propagation in the TCG of mode 10 are listed in Table 1.

Fault	Tank 3 Height	Tank 1 Outflow
C1-	00-	+-+
C2-	0-+	0+-
C3-	-+-	00+
R1+	000	+-+
R2-	00-	0-+
R4-	0-+	00-
R6+	-+-	000

Table 1: Candidates generated and their signatures

For faults C3- and R6+, the signatures imply a discontinuous change for the height in tank 3. Since this is not observed (see Fig. 10), these candidates are dropped. Similarly, C1- and R1- are dropped, since they predict a discontinuous change for the outflow from tank 1 (again this is not observed in Fig. 10). The outflow measurement signature for R2- (0 - +) does not match the actual outflow. For a purely continuous system, this fault hypothesis would be eliminated at this stage. However, when tracking hybrid behavior this cannot be done. Instead the algorithm assumes a mode change could have occurred, and it generates all possible transitions that are feasible, given the direction of change of variables, the autonomous transition definitions, and the predictions of the supervisory controller model Progressive monitoring is continued, but in each of the new hypothesized modes the outflow retains the same signature for fault R2-. Hypothesizing more autonomous changes would result in the same signature and hence R2- is dropped when we reach the predefined level of recursion for hypothesizing additional mode changes.

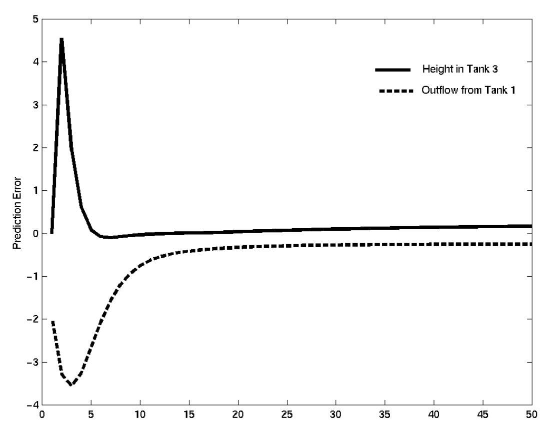

Figure 11: <8, C2-> identified as the true candidate by parameter estimation and statistical convergence

The remaining candidates <10, C2- R4+>, <9, C2- R4+> and <8, C2- R4+>, cannot be distinguished further by qualitative analysis, therefore, the quantitative parameter estimation algorithm is invoked. A fault observer is initiated for each of the remaining candidates. Estimation and a statistical check for convergence shows that that <8,C2-> is the true fault (see Fig. 11). For all other fault candidates the parameter estimation diverges.

6. EXPERIMENTS

We have studied the effectiveness of our hybrid algorithms by running extensive experiments on a three-tank system test-bed in our Modeling and Analysis of Complex Systems (MACS) laboratory. The controller model for the three-tank system (Fig. 3) is illustrated in Fig 2. Three fault scenarios are illustrated below. A fourth fault scenario was used as an illustrative example in Sec. 4.

The first experiment illustrates the importance of parameter estimation. The measure variable is the height of tank 3 (pressure e11 in the bond graph in Fig. 9). In this case, the hypothesis generation procedure generates two candidates <5,C2-> and <6,C2-> among other candidates. The fault is detected in mode 6 (at 11.2seconds). A number of fault hypotheses are generated, but progressive monitoring reduces it to the two candidates listed above. The qualitative signatures for the two faults are identical (similar to table 1), and Fig. 12 illustrates the measured height in tank 3 for the two faults. The solid curve represents the behavior of the system under nominal conditions. The dotted and dashed curves represents the two faulty behaviors Parameter estimation, however, establishes <5, C2-> as the true fault.

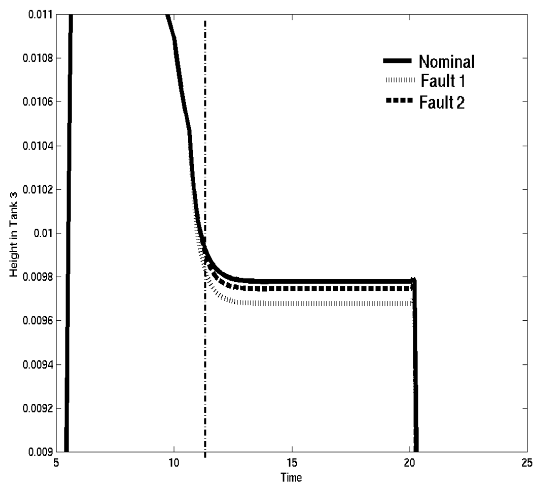

Figure 12: Experiment 2: Measurement is the height of liquid in tank 3

The second experiment shows that the fault isolation procedure works even if the observer predicts an incorrect mode trajectory after the fault occurs. For the case in which the abrupt fault, C1- occurs in mode 2 (t=4s), and the outflow from tank 1, *f3,* is the only measured variable, the fault is not detected until mode 8 (t=20.5s), when the deviation in out flow from tank 3 becomes significant (see Fig. 13). Fig. 13 also shows the height of liquid in tank 1 (this corresponds to pressure e2 in the bond graph in Fig. 9, and this height is not measured) to illustrate that the fault occurred earlier. An autonomous event corresponding to flow in the upper connecting pipe occurs after t=4s, but the observer is unable to predict this event since the height is estimated assuming nominal conditions. The observer predicts the opening of the pipe connecting tanks 1 and 2 as the next controlled event. Therefore, the actual mode sequence to the point when the failure is detected is 1, 2, 3, 4, … 8 (see Fig. 2), whereas the predicted mode sequence is 1, 2, 4, …, 8, i.e., the observed mode sequence is different from the actual mode sequence. Since the mode in which the fault occurs is part of the predicted observer trajectory, BACK_PROP adds the <2,C1-> fault candidate to the hypothesis list. Fault candidates from modes 4 through 8 are also generated, but they are all eliminated very early in the hypothesis refinement process described in the first experiment. When the parameter estimation procedure is invoked, the system succeeds in isolating the true fault.

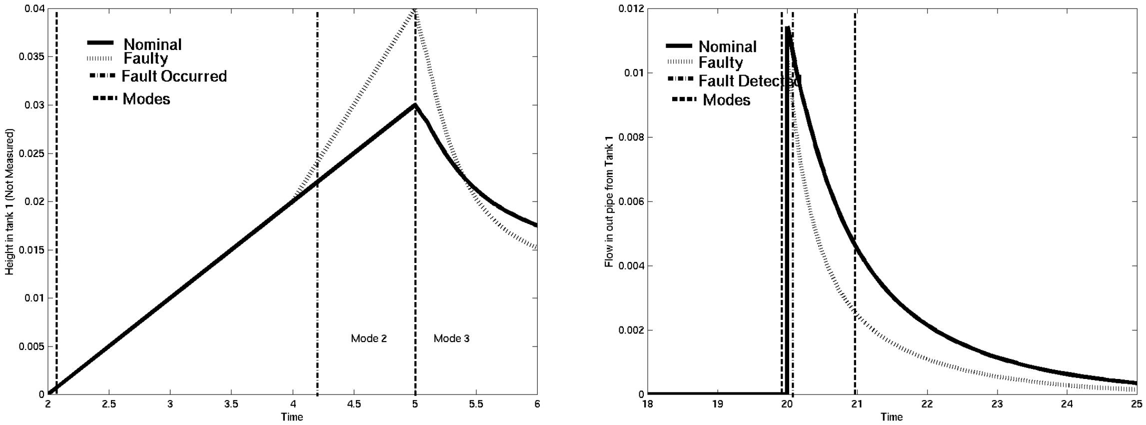

Figure 13: Experiment 3 – The true fault is isolated even though the observer misses a mode transition

In the third experiment (Fig. 14), we illustrate that even if the fault cause our observer to go through modes that system does not actually go through, the hypothesized faults in these modes are eventually dropped in the fault isolation phase. We introduce a fault (C1 +) in the system at time 4 (mode 2). Again we measure only the outflow from tank 1. Fig.11 also shows the height in tank 1 to indicate that trajectory is affected significantly even though the difference between actual and observed mode sequences is one mode (3).

The actual system goes through the mode sequence (1,2,4,5,6,7,8,9,10) but the observer predicts the mode sequence (1,2,3,4,5,6,7,8,9,10). The sudden increase in the capacity of the tank 1 causes the height in tank 1 to drop and due to this mode 3 (corresponding to the height in tank 1 crossing the upper connecting pipe) never occurs. Since the observer is not aware of the fault, it uses the nominal value of the height and predicts the occurrence of the autonomous events and hence goes through mode 3. At the time of fault detection, back propagation selects candidates from mode 3 also but these candidates get eliminated either in the hybrid forward propagation or parameter estimation steps.

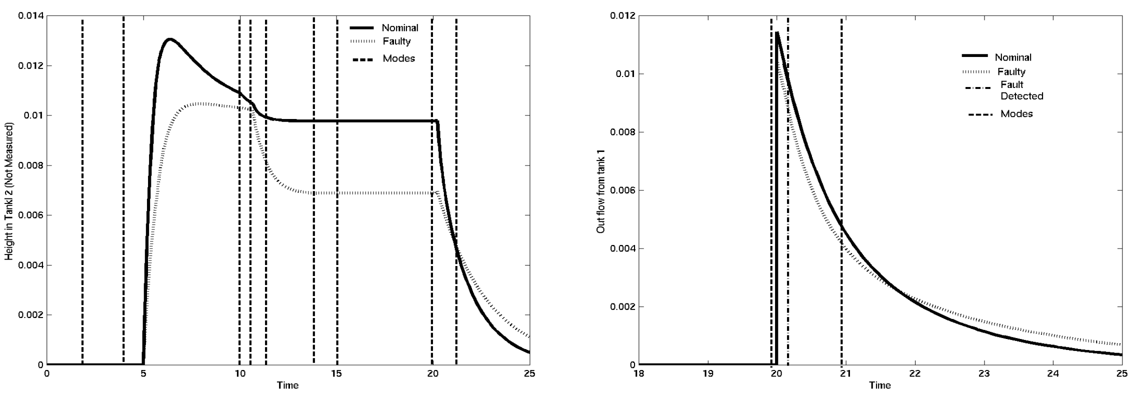

Figure 14: Experiment 4 -- Correct fault isolation occurs even though the observer predicts additional modes after the fault occurrence

7. CONCLUSIONS

In this paper, we extend previous work in FDI of hybrid systems [10,11], which assume the mode sequence in behavior evolution is known even after faults occur in the system. Relaxing this assumption could create an exponential blow up in the number of mode transitions that need to be considered after fault occurrences (Number of trajectories that have to be considered is the number of modes in the system x the number of potential faults). In previous work (e.g., [15,16]) this is avoided by assuming the fault is detected in the mode in which it occurs, or it is known when exactly the effect of a fault will be observed. In either case the controller model is then used to pre-compile the fault trajectories of the system. In this work, since we are dealing with a mostly continuous system, the pre-enumeration of fault trajectories is computationally intractable for the reasons mentioned above. To avoid the intractability problem, like [15,16] and other work we assume that have a correct model of the supervisory controller. We do not pre-enumerate fault trajectories, but by incorporating the supervisory controller model into our approach, we significantly cut down on the search space that we explore during the back propagation (hypothesis generation) step of our FDI algorithm. This is based on the observation that the mode in which the fault occurred must lie in the trajectory hypothesized by the hybrid observer, therefore, we generate fault hypotheses only for the modes predicted by the observer using the controller model. Similarly, the controller model helps cut down search during forward propagation to generate predicted behaviors. Instead of considering all possible transitions from a mode, we consider only the ones specified by the controller and autonomous transitions that may occur.

A significant component of our hybrid diagnosis system involved the design of the hybrid observer for tracking continuous behaviors across mode changes. An efficient scheme that compiles our hybrid bond graphs [7] into hybrid automata [1] was described in [12]. Future work will involve building observers that can perform mode identification based only on measurements, under nominal and faulty conditions. This will permit us to identify actuator, sensor and plant faults. We also need to extend our work to derive more robust online parameter estimation techniques. The observer will then be integrated with our qualitative and quantitative diagnosis algorithms for fault detection and isolation in hybrid systems, and also for fault-adaptive control of complex systems.

Acknowledgements. The research is supported by grants from DARPA SEC under contract number #F33615-99-C-3611 and Xerox PARC. The authors benefited a lot from discussions with Dr. Feng Zhao at Xerox PARC.

8. REFERENCES

1. Alur, R., et al., 1993. *Hybrid Automata: an algorithmic approach to the specification and verification of hybrid systems,* in R.L. Grossman, et al, eds., Lecture Notes in Computer Science, Springer, Berlin, 736, pp. 209-229.
2. Branicky, M.S., V. Borkar, S. Mitter, 1994. *A Unified Framework for Hybrid Control: Background, Model, and Theory*, Proc. 33rd IEEE Conf. on Decision and Control, FL, Paper No. LIDS-P-2239.
3. Garcia, E.A., Frank, P.M. 1997. *Deterministic non-linear observer based approaches to fault diagnosis – A survey*, Control Engg. Practice, Vol. 5 (5), pp. 663-670.
4. Lunze J., 1999. *A timed discrete-event abstraction of continuous-variable systems*. Intl. Jour. of Control, 72, 1147-1164.
5. Manders E.J., P.J. Mosterman, G. Biswas, 1999. *Signal to symbol transformation techniques for robust diagnosis in TRANSCEND,* Tenth Intl. Workshop on Principles of Diagnosis, Loch Awe, Scotland, pp. 155-165.
6. Manders E.J., et al., 2000. *A combined qualitative quantitative approach for efficient fault isolation in complex dynamic systems*, 4th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS, pp. 512-517.
7. Mosterman P.J., G. Biswas, 1998. *A theory of discontinuities in physical system models*, Jour. of the Franklin Inst., 335B, pp. 401-439.
8. Mosterman P.J., Biswas G., 1999. *Diagnosis of Continuous Valued Systems in Transient Operating Regions*. IEEE Trans. on Systems, Man and Cybernetics: 29, pp. 554-565.
9. Mosterman P.J., Biswas G., 2000. *Towards Procedures for Systematically Deriving Hybrid Models of Complex Systems*. Third Intl. Workshop. on Hybrid Systems: (HSCC'00), pp. 324-337.
10. McIllraith S., et al., 2000. *Towards Diagnosing Hybrid Systems*. Third Intl. Wkshp. on Hybrid Systems: (HSCC'00), pp. 282-295.
11. Narasimhan, S., et al., 2000. *Fault Isolation in Hybrid Systems Combining Model-based Diagnosis and Signal Processing*, 4th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS, pp. 1074-1079.
12. Narasimhan, S., et al., 2000. *Building Observers to Handle Fault Isolation and Control Problems in Hybrid Systems*, Proc. IEEE Intl. Conf. on SMC, Nashville, TN, pp. 2393-2398.
13. Patton, R.J., Frank, P.M., and Clark, R.N. (eds.), 2000. *Issues of Fault Diagnosis for Dynamic Systems,* Springer-Verlag, London, U.K.
14. Rosenberg, R.C., Karnopp, D. 1983. *Introduction to Physical System Dynamics,* McGraw Hill, New York.
15. Sampath M., et al., 1996. *Failure Diagnosis using Discrete Event Models*. IEEE Trans. On Control Systems Tech., 4(2), pp. 105-124.
16. Sampath, M, et al, 2000. *Combining Qualitative and Quantitative Reasoning: A Hybrid Approach to failure Diagnosis of Industrial Systems.* 4th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS, pp. 494-501.