Building observers to address fault isolation and control problems in hybrid dynamic systems

[1]Sriram Narasimhan, [1]Gautam Biswas, [1]Gabor Karsai, [1]Tal Pasternak, and [2]Feng Zhao

[1]Vanderbilt University, Nashville, TN 37235.
[2] Xerox Palo Alto Research Center, Palo Alto, CA

Abstract. Model based approaches to diagnosis for dynamic systems have been based on continuous and discrete event models. Systems that combine continuous and discrete behaviors, i.e., hybrid systems have been typically abstracted into discrete event models or approximated by continuous models with steep slopes so that existing algorithms can be applied for fault isolation tasks. This approach runs into problems when both discrete events and continuous behaviors provide vital diagnostic information. We propose a diagnostic methodology that uses hybrid models of the system to perform diagnosis.

1. Introduction

Modern systems are complex in nature. They can include supervisory control that switches modes of behavior of the system to optimize system performance. In other words, discrete control actions change the operating region of behavior. The implication of this is that multiple models of the system have to be employed and model switching has to be performed at run time to execute monitoring, fault isolation, and control tasks. The hybrid nature of systems requires new forms of analysis because discrete changes are not handled well by continuous algorithms, and abstracting system behavior to discrete models may result in loss of information critical for fault isolation and control.

Model based diagnosis provides a framework for analyzing the global behavior of the system. Given that we have a limited number of sensors, model-based analytic redundancy methods have to be applied to derive non-local interaction between the faults and observations. Current model based diagnosis is based on discrete event and continuous techniques [SSLSD96; L99; MB99; G98]. Hybrid system diagnosis is performed by abstracting the system models to continuous or discrete event form. This approach is not sufficient when both continuous behavior and discrete events together are required to generate the necessary diagnostic information. In a hybrid system, the behavior evolves continuously until a discrete event causes it to move to a different continuously evolving region in the behavior space. Coming up with continuous representations of hybrid systems can result in very complex non-linear functional relations that are hard to analyze in real time. On the other hand, pure discrete event systems require a lot of simulation and can diagnose only qualitative faults. We propose a diagnosis methodology that uses hybrid models of the system and thereby performs hybrid diagnosis.

We motivate the need for hybrid model based diagnosis with an example. Consider the three-tank system illustrated in Fig. 1. The system consists of tanks, flow sources, outlet pipes, and connecting pipes. Some pipes contain valves that can be opened or closed by an external controller. Two types of discrete events may occur in the system. External control actions change the configuration of the system by opening and closing valves. Autonomous jumps are also possible in the system. For example, when the fluid level in tank 1 is higher than pipe R3, there exists a flow in pipe R3.

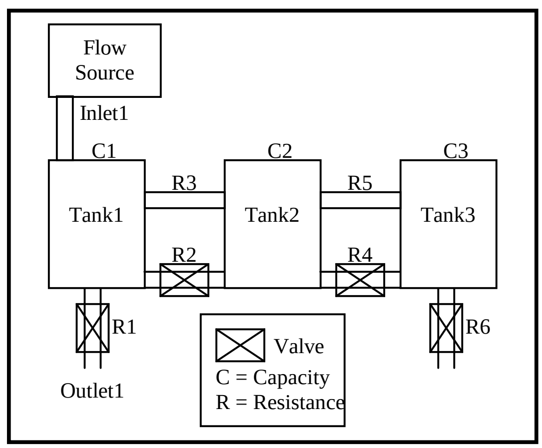


Figure 1: Three-tank system

System behavior, i.e., the flows in the pipes and the levels of fluid in the tanks are all governed by continuous differential equations. These equations change as the valve configurations change. For example, when all the valves are open and the fluid level in the tanks are higher than pipe R3 and R5, the state equations of the system are given by,

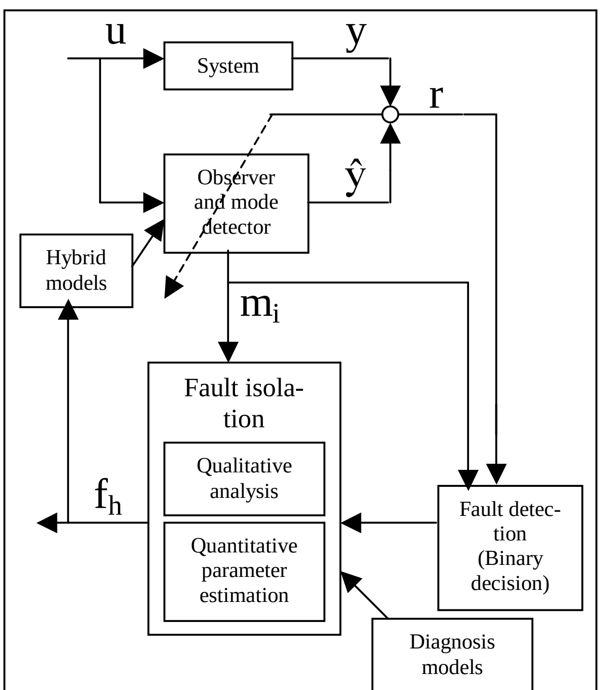
$$\begin{bmatrix}\dot{h}_1\\ \dot{h}_2\\ \dot{h}_3\end{bmatrix} = \begin{vmatrix} -\frac{1}{C_1R_1}-\frac{1}{C_1R_2}-\frac{1}{C_1R_3} & \frac{1}{C_1R_2}+\frac{1}{C_1R_3} & 0 \\ \frac{1}{C_2R_2}+\frac{1}{C_2R_3} & -\frac{1}{C_2R_2}-\frac{1}{C_2R_3}-\frac{1}{C_2R_4}-\frac{1}{C_2R_5} & \frac{1}{C_2R_4}+\frac{1}{C_2R_5} \\ 0 & \frac{1}{C_1R_1}+\frac{1}{C_1R_1} & -\frac{1}{C_1R_1}-\frac{1}{C_1R_1}-\frac{1}{C_1R_1} \end{vmatrix} \begin{bmatrix} h_1\\ h_2\\ h_3\end{bmatrix} + \begin{vmatrix}\frac{1}{C_1}\\ 0\\ 0\end{vmatrix} f_1$$

where h_1, h_2, and h_3 correspond to the heights in the three tanks, and f_1 is the flow into tank 1. A specific valve configuration and the existence or non-existence of flow in pipes R3 and R5 determine the mode of the system. Since there are 4 valves and two pipes where autonomous jumps may occur, there are 64 (2^6) total modes of the system. Each mode is governed by a different set of equations. Some of the faults that may occur in the system are leaks in the pipes and change in tank capacities.

Continuous system diagnosis methods break down across mode changes. To apply discrete event diagnosis techniques, we would have to enumerate all modes, perform simulation in each mode, and derive relations between faults and their manifestation in a discrete framework. It is clear that even for a small system like this, this approach becomes impractical. Hence there is a need to approach the diagnosis problem in a hybrid framework.

We propose a diagnosis methodology that uses hybrid models. This approach can overcome some of the problems associated with continuous system diagnosis or discrete event diagnosis. We suggest a combination of qualitative and quantitative strategies to efficiently identify a candidate fault set and refine this set. We will focus on the three-tank example described above to explain our methodology.

2. Hybrid system diagnosis architecture



u = inputs, y = measured output from system,
$\hat{y}$ = predicted output from model,
r = residual $y - \hat{y}$, m_i = current mode,
f_h = fault hypotheses

Figure 2: Hybrid diagnosis methodology

We make the following assumptions in our work.

- Only single faults occur.
- Each fault is associated with a parameter in the system model. A fault causes an abrupt change in the parameter value.

Our diagnosis methodology illustrated in Fig. 2 consists of three mains steps, (i) using a hybrid observer to track system behavior, (ii) detecting a fault occurrence, and (iii) isolating the fault in the system.

The hybrid observer uses the models of the system to track system behavior. We use switched bond graphs [MB98] as the primary modeling language for building hybrid system models. The observer uses the state equations models for tracking continuous behavior in a mode, and a hybrid automata for detecting and making mode transitions as system behavior evolves. Detection of mode changes requires access to controller signals for controlled jumps, and predictions of state variable values for autonomous jumps. If a mode change occurs in the system, the observer switches the tracking model (different set of state space equations), initializes the state variables in the new mode, and continues tracking system behavior with the new model. The fault detector compares the observations from the system and the predictions from the observer to look for significant deviations in the observed signals. We use a simple decision scheme that signals a fault, if the discrepancy between an observation and prediction exceeds a pre-specified threshold for a few time steps, or if an abrupt change is detected in a signal value that cannot be explained by a mode change [MMB99].

Once the fault has been detected, qualitative and quantitative techniques are used to isolate the fault in the system. We use temporal causal graphs derived from switched bond graphs for the qualitative analysis, and state equations derived from bond graphs for the quantitative analysis. In the qualitative analysis, we first identify an initial candidate set to explain the discrepancy in the observations and predictions. This is achieved by back propagating the qualitative value of the discrepancy (-, 0 or +) through the temporal causal graph of the system. The back propagation may have to be continued in previous modes to identify all possible candidates [MB99]. We can qualitatively predict future behavior of the system under each of the hypothesized fault conditions by forward propagating through the causal graph. These predictions include magnitude and higher order derivatives of the variables of the system. The predictions can be compared against the qualitative value of the observations to refine the candidate set. We use a technique called progressive monitoring [MB99] to achieve this.

For the quantitative analysis, we estimate the deviated parameter values for each of the remaining fault candidates. To do this, we rewrite the state space equations in terms of the parameter associated with the fault candidate and use system identification techniques to estimate the parameter value [MNBM00]. These estimated parameter values could be used to quantitatively predict future behavior of the system, which can be compared to the observations from the system to eliminate some candidates.

In this paper we focus on developing the hybrid models and building the hybrid observer. The fault detection and fault isolation tasks are still work in progress and will not be discussed in this paper. In the next section we describe our modeling paradigms namely, hybrid automata, switched bond graphs, temporal causal graphs and state space equations. In the subsequent section we show how these models can be used to build a hybrid observer for the system.

3. Modeling for diagnosis

Earlier attempts at modeling tried to abstract hybrid systems as either discrete event systems or continuous systems. Traditionally discrete event systems have been modeled by finite state automata, Petri nets, and directed graphs, whereas continuous systems are typically represented by differential equation models, circuit diagrams, and block diagrams. Although it is true that most hybrid systems are continuous systems at the lowest level of detail, building hybrid models proves useful in analyzing the system at a level of detail that is useful for diagnosis.

3.1 Hybrid automata

Our approach to hybrid modeling involves building hybrid automata that combines finite state automata (FSA) with continuous representations [A93]. The FSA, whose states correspond to the modes of operation of the system, captures the possible mode transitions in the system. The FSA representation is enhanced by including a continuous system model in each state that governs behavior evolution for that state. The number of modes of a system may be large enough to make it infeasibile to exhaustively generate the complete hybrid automata. We avoid this computational problem by enumerating states of the hybrid automata (which correspond to the modes) dynamically as system behavior evolves. When the system is any given mode, we compute all possible transitions from the current state and enumerate only those transitions and corresponding destination states. This step involves computing three things

- Destination state and model
- Conditions for transition to the destination state
- Reset conditions (i.e., the state vector value) in the new state

The conditions for transition can be specified in terms of the occurrence of control signals and the state variables of the system reaching certain boundary conditions. The reset conditions initialize the values of variables of the system model in the destination state.

We look at the example of the three-tank system presented earlier. We represent the mode of the system as a 6-tuple corresponding to whether there is a nonzero flow through pipes R1 to R6. For pipes R1, R2, R4 and R6 this corresponds to the valve on the pipe being open or not (controlled jumps). For pipes R3 and R5, this corresponds the level of fluid in the tanks exceeding a predefined value so as to cause a flow in the pipes (autonomous jumps). If in the current mode (state) all pipes are closed then Fig. 3.1 illustrates the relevant part of the FSA where only neighboring states are enumerated under the assumption that only one switch is activated at a time.

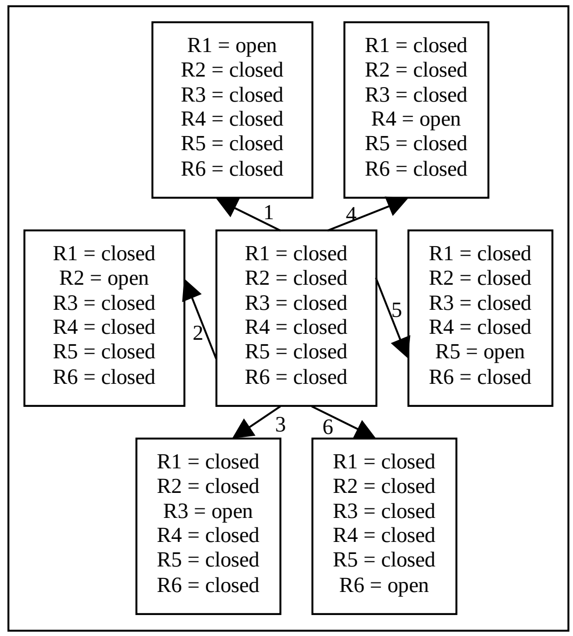


Figure 3.1: Finite State Automata

Fig. 3.2 shows the transition conditions and reset conditions for each of the six transitions in Fig. 3.1.

1 : OPEN Valve R1
reset = none
2 : OPEN Valve R2
reset = none
3 : h1 >= Position_R3 or h2 >= Position_R3
reset = none
4 : OPEN Valve R4
reset = none
5 : h2 >= Position_R5 or h3 >= Position_R5
reset = none
6 : OPEN Valve R6
reset = none
h1, h2, h3 – height of fluid in tanks 1,2 and 3 respectively
reset = none implies there is no discontinuous change in state vector across mode transitions

Figure 3.2: Transition and reset conditions

3.2 Continuous models

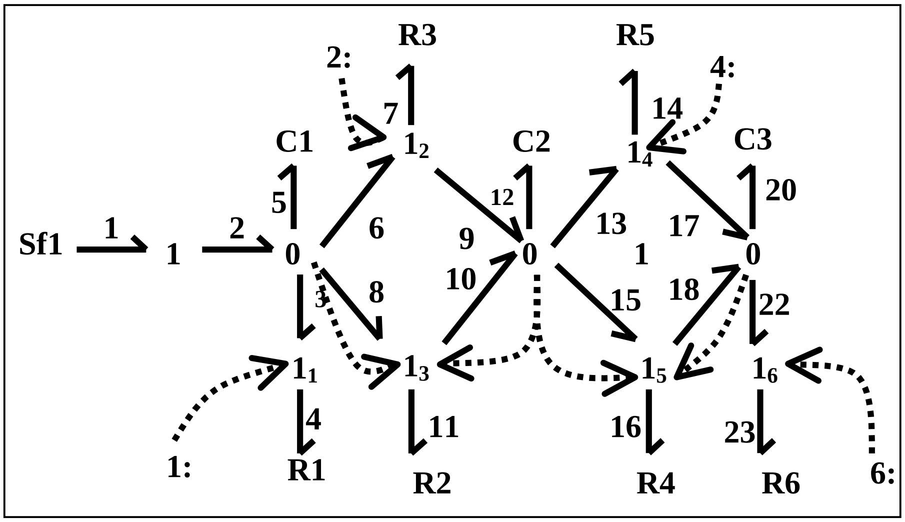


Figure 4.1: Switched bond graph of three-tank system

As mentioned before, our models for continuous behavior in a given mode are bond graphs [RK83]. Bond graphs are energy-based models of the system in terms of the effort and flow variables of the system. Bonds specify interconnections between elements that exchange energy, which is given by the rate of flow of energy, *power = effort x flow*. Bond graphs represent a generic modeling language that can be applied to a multitude of physical systems, such as electrical, fluid, mechanical, and thermal systems. There exist standard techniques to build bond graph models of systems based on physical principles. State equations can be systematically derived from the bond graph representation of the system. In addition, we can also systematically derive temporal causal graphs from bond graphs. This is very important since state equations can be used to simulate system behavior and state equations and temporal causal graphs constitute our diagnosis models.

We use an enhanced form of bond graphs through the use of switched junctions that facilitate the modeling of discrete mode transitions in system behavior [MB98]. In general, the number of modes and possible transitions in a system model can be quite large. Instead of pre-enumerating the bond graph for each mode to build a complete hybrid automata, the complete system model is developed as a switched bond graph, where individual junctions model local mode transitions. The switching 0- and 1- junctions represent idealized discrete switching element that can turn the corresponding energy connection on and off.

The physical on/off state for each of these switched junctions is determined by external control signals and continuous variables crossing pre-specified thresholds. These can be specified as finite state sequential automata.

Fig. 4.1 represents the switched bond graph model of the three-tank system. Fig. 4.2 shows the sequential automata for the switched junctions 1_1 (externally controlled) and 1_3 (autonomously controlled) in Fig. 4.1.

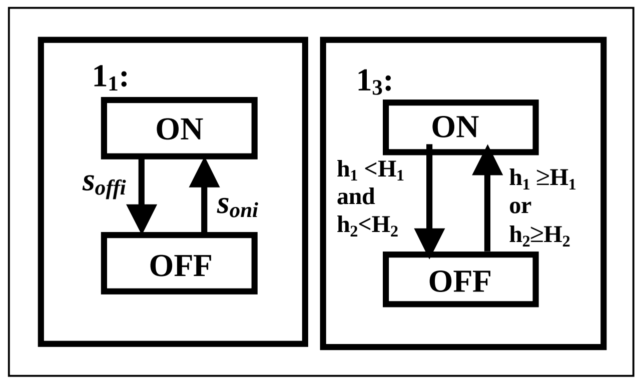


Figure 4.2: Automata for switched junctions

As mentioned before we enumerate states of the hybrid automata dynamically as system behavior evolves. In the bond graph framework, this can be achieved in the following fashion. For any given state we identify all its neighboring states by changing the status of each switch in the switched bond graph one at a time. We can determine the transition and reset conditions for these transitions from the local sequential automata of the corresponding switched junction. This approach is illustrated in Figs. 3.1 and 3.2

4. Hybrid Observer

The hybrid observer tracks the system behavior across different modes of operation. This involves two main steps,

- Simulating continuous system behavior in individual modes of operation
- Identifying and executing all mode changes including controlled and autonomous jumps.

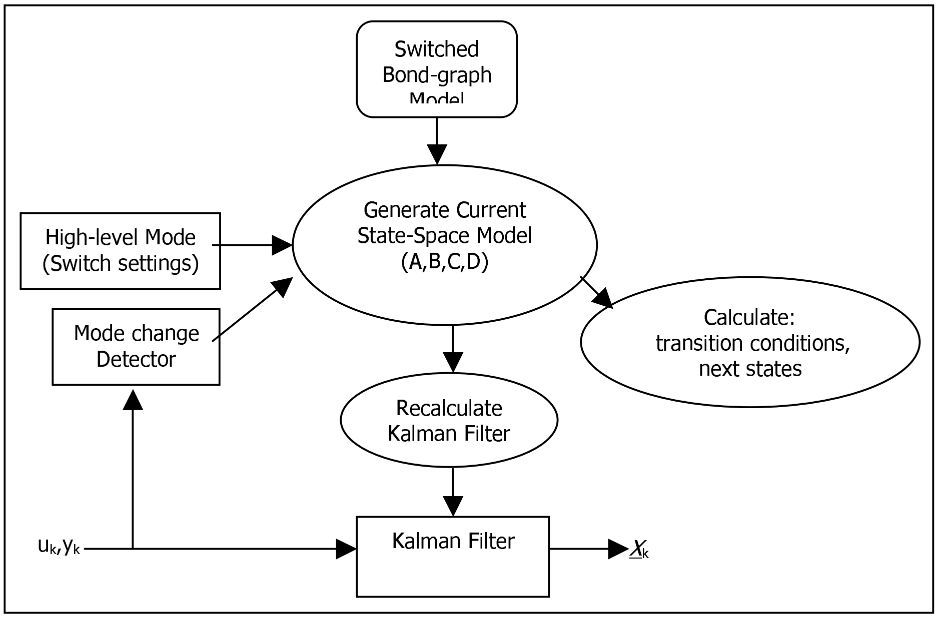


Figure 5: Hybrid observer

As described before, our models for continuous behavior in a mode are bond graphs. These bond graphs can be used to derive the state equations of the system in the current operating mode. Once initial conditions are specified or derived, these equations provide the model for tracking system behavior. We assume that we have access to the controller commands and hence controlled jumps can be identified. We rewrite all autonomous jump conditions in terms of state variables of the systems. Since we can estimate the values of the state variables from the state space equations, autonomous jumps can also be identified. When a mode change occurs, the hybrid observer needs to switch models and initialize the new model accordingly. For this purpose, we use the hybrid automata model of the system that tells us the destination state (model) and the reset conditions for any specific transition. Assuming that we know all nominal parameter values and the initial conditions when the observer is started we can track the system behavior by following the hybrid automata as mode changes are determined and switching models with appropriate reset conditions when doing so. Fig. 5 illustrates our approach to building a hybrid observer.

Since the input and output of the system may be affected by noise and our state space models may not be accurate we use a Kalman filter [G79] to track system behavior in a single mode of operation. For a given state space equation model a Kalman gain matrix, K, can be computed if we know the covariance of the noise that affects the input and output. The difference in the actual observations and predicted observations ($y - \hat{y}$) is multiplied by this gain matrix and used to estimate the state variables of the system. The enhanced state space equations using a Kalman filter is given by,

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

where A, B, and C are the system matrices, Q is the input noise covariance, R is the output noise covariance, P is the error covariance, and K is the Kalman gain matrix.

Fig. 6 illustrates a sample run of our hybrid observer as it tracks the three-tank system through three modes. In the first mode (10 seconds), there is an inflow to tank 1 and R2 and R4 are open (filling up tanks). In the second mode (10 seconds), there is no inflow to tank 1 and R1, R2, R4 and R6 are all open (draining tanks). In the third mode (10 seconds), there is no inflow to tank1, and only R2 is open (isolating tank 3). We assume that the input noise and output noise covariance is 0.001. In the figure, we see the heights in the three tanks over time. It is interesting to note that the Kalman filters in modes 1 and 2 have no trouble in accurately tracking system behavior, but in mode 3, where there is an abrupt change in flow value, the predicted level values for tanks 1 and 2 are initially inaccurate, but the system slowly converges to its true value. This implies that mode transitions with abrupt changes can cause problems 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.

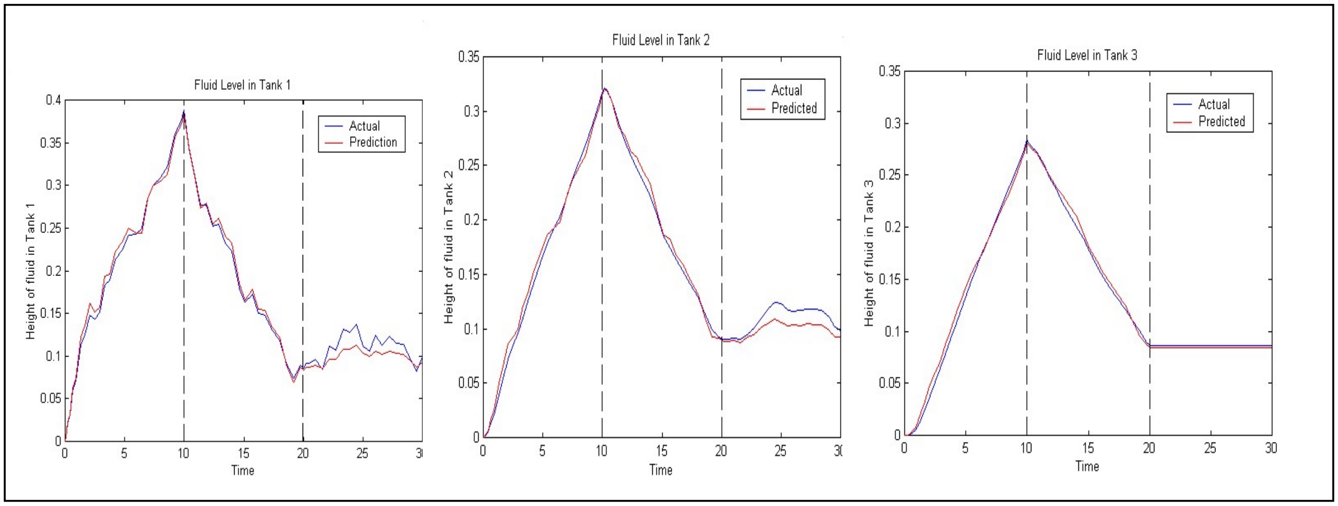


Figure 6: Sample run of hybrid observer

5. Conclusions

Our research goals are to build a diagnosis methodology that applies to hybrid systems. In this paper, we have presented a hybrid modeling paradigm that is used to construct observers for tracking hybrid behaviors in complex systems. This work bridges the gap between purely discrete event and continuous system modeling. We do not need the extensive simulation (for pre compilation) required by some of the discrete event systems [L99; SSLSD96]. On the other hand, we reduce some of the computational complexity of continuous system model by breaking it up into different modes of continuous operations.

Future work would involve building observers that can perform mode identification based only on measurements, under nominal and faulty conditions. Robust online parameter estimation techniques need to be developed. 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.

References:

[A93] **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.

[G79] **Gelb, A., ed.,** 1979. *Applied Optimal Estimation*, MIT Press, Cambridge, MA.

[L99] **Lunze J.**, 1999. *A timed discrete-event abstraction of continuous-variable systems*. Intl. Jour. of Control, **72**, pp. 1147-1164.

[G97] **Gertler, J.J.,** 1998. *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, NY.

[MB98] **Mosterman P.J., G. Biswas,** 1998. *A theory of discontinuities in physical system models*, Journal of the Franklin Institute:**335B**, pp. 401-439.

[MMB99] **Manders E.J., P.J. Mosterman, G. Biswas**, 1999. *Signal to symbol transformation techniques for robust diagnosis in TRANSCEND,* Tenth International Workshop on Principles of Diagnosis, Loch Awe, Scotland, pp. 155-165.

[MNBM00] **Manders E.J., S. Narasimhan, G. Biswas, P.J. Mosterman**, 2000. *A combined qualitative/quantitative approach for efficient fault isolation in complex dynamic systems*. 4th symposium on fault detection, supervision and safety processes, pp. 512-517.

[MB99] **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.

[MB00] **Mosterman P.J., Biswas G.,** 2000. *Towards Procedures for Systematically Deriving Hybrid Models of Complex Systems*. Third International Workshop on Hybrid Systems: Computation and Control (HSCC'00), pp. 324-337.

[MBCG00] **McIlraith S., Biswas G., Clancy D., Gupta V.,** 2000. *Towards Diagnosing Hybrid Systems*. Third International Workshop on Hybrid Systems: Computation and Control (HSCC'00), pp. 282-295.

[RK83] **Rosenberg, R.C., D. Karnopp,** 1983. *Introduction to Physical System Dynamics,* McGraw Hill, New York.

[SSLSD96] **Sampath M., Sengupta R., Lafortune S., Sinnamohideen K., Demosthenis T.C.,** 1996. *Failure Diagnosis using Discrete-Event Models*. IEEE Trans. On Control Systems Technology, **4(2),** March 1996.