In Proc. of IJCAI-2001

Distributed Monitoring of Hybrid Systems: A model-directed approach

Feng Zhao and **Xenofon Koutsoukos** and **Horst Haussecker**
James Reich and **Patrick Cheung**
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304, U.S.A.

Claudia Picardi
Dipartimento di Informatica
Universita di Torino
Corso Svizzera 185, 10149 Torino, Italy

Abstract

This paper presents an efficient online mode estimation algorithm for a class of sensor-rich, distributed embedded systems, the so-called hybrid systems. A central problem in distributed diagnosis of hybrid systems is efficiently monitoring and tracking mode transitions. Brute-force tracking algorithms incur cost exponential in the numbers of sensors and measurements over time and are impractical for sensor-rich systems. Our algorithm uses a model of system's temporal discrete-event behavior such as a timed Petri net to generate a prior so as to focus distributed signal analysis on when and where to look for mode transition signatures of interest, drastically constraining the search for event combinations. The algorithm has been demonstrated for the online diagnosis of a hybrid system, the Xerox DC265 printer.

1 Introduction

Many man-made electro-mechanical systems such as automobiles or high-speed printers are best described as hybrid systems. The dynamics of a hybrid system comprises continuous state evolution within a mode and discrete transitions from one mode to another, either controlled or autonomous. A mode of an automobile could be an acceleration phase or a cruising phase. A printer may have a paper feeding phase followed by a registration phase. Within each mode, the dynamics of a system is governed by a continuous behavioral model. Under a control signal, such as gear shift, the system may transition to a different operating mode. Certain transitions are autonomous due to system state reaching a threshold value. For example, when a paper feed roll contacts a sheet of paper, the paper starts to move.

Diagnosis of a hybrid system requires the ability to monitor and predict system behaviors and detect and isolate faults. The monitoring task involves estimating and tracking system state and is at the heart of hybrid system diagnosis. In a sensor-rich environment, the monitoring task is significantly complicated by the need to associate data from multiple sensors with multiple hypotheses of states, modes, or faults. This is particularly true for distributed detection and diagnosis in large complex systems such as highway traffic monitoring or large print shop fault diagnosis where the numbers of sensors and system components can potentially be very large (1,000 – 10,000 or more). Recent advances in micro-electro-mechanical systems (MEMS) and wireless networking have enabled a new generation of tiny, inexpensive, wirelessly connected MEMS sensors. As shown in Section 2, the complexity of brute-force monitoring schemes is exponential in the numbers of sensors and measurements over time and is clearly not scalable. Our algorithm addresses this computational problem.

Monitoring of hybrid systems has two components, mode estimation and (continuous) state tracking. Once a system is estimated to be in a particular mode, a continuous state estimator such as Kalman filter could be used to track the continuous state. This paper focuses on the more difficult problem of mode estimation and its application to sensor-rich, distributed hybrid system monitoring and diagnosis.

Example. Consider the problem of workflow identification and fault diagnosis in a document processing factory (or print shop), where multiple printing, collating, and binding machines may be placed in proximity of each other. The objective is to identify critical printing job and machine operating parameters for online workflow scheduling and fault diagnosis. An example of the printing equipment is the Xerox Document Center DC265 printer, a multifunction system that prints at 65 pages per minute (Fig. 1). The system is made of a large number of moving components such as motors, solenoids, clutches, rolls, gears, belts and so on. A fault of "no paper at output" may be caused by abrupt failures such as a broken transfer belt. Paper jams are often caused by subtler component degradation such as roll slippage or timing variations of clutch, motor or solenoid due to wear, some of which is not directly observable with the system's built-in sensors and must be estimated using system behavioral model and additional sensor information. The printer is an example of a hybrid system as is illustrated here using its paper feed subsystem. A component such as the feed motor may be in any one of the ramp-up, rotating with constant speed, ramp-down, stationary states, each of which is governed by continuous dynamics. Mode transitions are induced by either control events or evolution of the continuous dynamics. For example, the transition from stationary to ramp-up for the motor is caused by "turn_motor_on" control event and can be estimated using the control event and sensor signal. However, a transition such as acquisition roll contacting a paper is autonomous and must be estimated using model and sensor data.

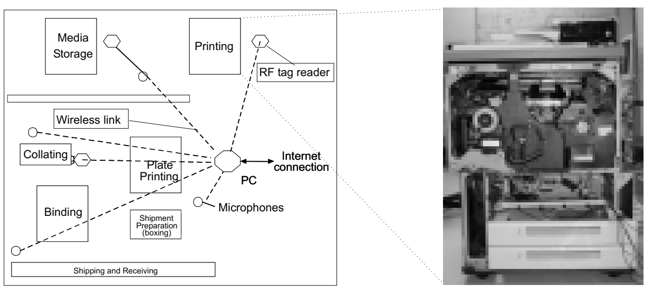

Figure 1: Print shop with multiple machines such as a Xerox DC265 printer.

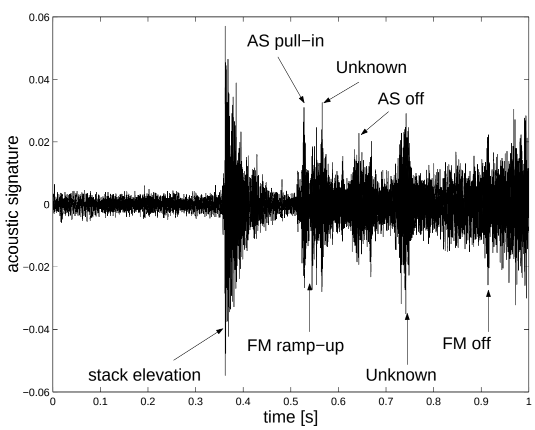

Figure 2: Acoustic signal for a one-page printing operation of DC265 printer.

In this example, estimating the timing of the roll contacting paper requires to single out the solenoid pull-in event from incoming sensor data streams. The paper path system of the printer has 3 motors, 10 solenoids, 2 clutches, and a large number of gears connecting the motors to different rolls and belts. The testbed to be detailed in Section 4 uses 18 sensors, each sampled at 40,000 times per second. Many of the system components may be active around the time of solenoid pull-in – the so-called cocktail party phenomenon in speech processing (Fig. 2). Moreover, other machines may be active in an environment such as a print shop. As the number of event hypotheses scales exponentially with the numbers of sensors, system components, and measurements (Section 2), *pulling the relevant events out of a large number of high-bandwidth data streams from a multitude of simultaneous sources is a tremendous computational challenge* – the main **computational problem** this paper addresses. Signature analysis techniques such as [Hung and Zhao, 1999] could not immediately be applied to mode estimation without using a model to focus on when to acquire data and where to look for events.

This paper describes an efficient mode estimation algorithm for hybrid systems. The algorithm integrates model-based prediction with distributed signature analysis techniques. A timed Petri net model represents the temporal discrete-event behavior of a hybrid system. The model generates event predictions that focus distributed signal processing on when and where to look for signatures of interest, and estimation of signature in turn refines and updates model parameters and states. The algorithm is novel in its use of model knowledge to drastically shrink the range of a time-domain search for events of interest, and has been experimentally validated on a multi-sensor diagnostic testbed.

Monitoring and diagnosis of hybrid systems are an active research area. Existing approaches to hybrid system monitoring and diagnosis do not address the computational data association problem associated with distributed multi-sensor systems [Bar-Shalom and Fortmann, 1999] and assume sensor output has already been properly assembled to form likelihood functions of system output. Moreover, they assume either no autonomous mode transition or autonomous transition without signal mixing. [Lerner *et al.*, 2000] described a Bayesian network approach to tracking trajectories of hybrid systems. They introduced a method of smoothing that backward propagates evidence to re-weigh earlier beliefs so as to retain weak but otherwise important belief states without explicitly tracking all the branches over time. [Kurien and Nayak, 2000] addressed the temporal branching problem in tracking using a consistency-based generation and revision of belief states in a partially observable Markov decision process formulation. [McIlraith, 2000] described a particle filter approach to tracking multiple models of behaviors. Qualitative diagnosis techniques such as [McIlraith *et al.*, 2000] are used to provide a temporal prior to focus the sampling of particle filter on part of the distribution consistent with the model prediction. In contrast, our approach exploits model knowledge of control and discrete-event behaviors of hybrid systems to address the exponential blow-up in data association of multi-sensor observation, as well as the complexity due to multiple measurements over time.

The rest of the paper describes three main **contributions** of this work: Section 2 presents a *formulation* of the mode estimation problem for distributed monitoring of hybrid systems and its computational challenges. Section 3 describes the mode estimation *algorithm* for both controlled and autonomous mode transitions. The algorithm has been demonstrated as part of a *diagnosis system* for the Xerox DC265 multifunction printer and *experimental results* are presented in Section 4.

2 Hybrid System Mode Estimation Problem

In the mode estimation problem we consider, a hybrid system is described by a 6-tuple: (X, Q, Σ, Y, f, g), where X is the continuous state space of the system, Q is the mode space (set of discrete states), Y is the space of sensor signals, Σ is the set of possible control inputs to the system, $f : X \times Q \times \Sigma \rightarrow X \times Q$ is the transition function, and $g : X \times Q \times \Sigma \rightarrow Y$ is the observation function.

The mode space Q can be understood as the product of individual component modes of an n-component system, with each mode vector denoted as $\mathbf{q} = [q_1, \ldots, q_n]^T \in Q$. For an l-sensor system, the sensor output vector is $\mathbf{y} = [y_1, \ldots, y_l]^T \in Y$, where y_i is the output of sensor i. Each y_i could be a measure of a signal from component mode q_j alone or a composite signal of multiple components.

The problem of mode estimation in a multi-sensor environment can be stated as follows. At each time step t, given the previous mode estimate at $t-1$ and current observation, mode estimation is a mapping:

$$\mathcal{E} : Q^{t-1} \times Y^t \rightarrow Q^t \tag{1}$$

Equivalently, the mode estimation problem is to estimate τ such that $\mathbf{q}^{\tau+1} = \mathcal{E}(\mathbf{q}^\tau, \mathbf{y}^{\tau+1})$, and $\mathbf{q}^{\tau+1} \neq \mathbf{q}^\tau$, i.e., the time instance when one or more component modes have changed.

Mode transitions induced by external control events can be estimated using the control events and sensor signals. Autonomous transitions must be estimated using a combination of system model, control event sequence, and sensor signals.

To estimate $\mathbf{q}^t \in Q^t$, components of y_i contributed by mode components q_j's must be associated with the q_j's in order to determine if there is a transition for q_j, and if so, what the parameters (such as transition time) are. We illustrate the *computational difficulties* of data association for the hybrid system mode estimation problem for two cases.

Case I. Assume there is no signal mixing and each y_i measures a signal $s_j \in S$ from system component j only. The number of possible associations of y_i's with the corresponding q_j's is n^l, that is, it is exponential in the number of sensors at each time step.

Case II. More generally, each sensor signal y_i measures a composite of s_j's through a mixing function: $\mathcal{H} : S^t \rightarrow Y^t$. Without prior knowledge about $\mathcal{H}$, any combination of s_j's could be present in y_i's. Pairing each y_i with s_j's creates $n!$ associations. The total number of associations of $\mathbf{y}$ with $\mathbf{q}$ is $(n!)^l \propto 2^{nl}$, i.e., exponential in the numbers of sensors and signal sources.

For applications such as diagnosis, it is often necessary to reason across multiple time steps and examine the history of mode transitions in order to identify a component fault occurred in an earlier mode. Each pairing of observation with mode vector in the above single-step mode estimation creates a hypothesis of the system mode transition sequence. As more observations are made over time, the total number of possible mode transition sequences is exponential in the numbers of sensors *and* measurements over time.

3 An Online Mode Estimation Algorithm

The objective of mode estimation is to estimate the mode transition sequence of a hybrid system:

$$\mathbf{q}^{\tau_1} \rightarrow \mathbf{q}^{\tau_1+1} = \mathbf{q}^{\tau_2} \rightarrow \mathbf{q}^{\tau_2+1} \ldots \mathbf{q}^{\tau_k} \rightarrow \mathbf{q}^{\tau_k+1} \ldots$$

where $\mathbf{q}^{\tau_i} \neq \mathbf{q}^{\tau_i+1}$. Each transition is caused by one or more mode transitions of components of $\mathbf{q}$.

Assuming each sensor output y_i is a linear superposition[1] of possibly time-shifted s_j's

$$y_i(t) = \sum_{j=1}^{n} \alpha_{ij} s_j(t - \tau_{ij}), \quad i = 1, \ldots, l \tag{2}$$

or more compactly,

$$\mathbf{y}^t = D(\alpha_{ij}, \tau_{ij}) * \mathbf{s}^t \tag{3}$$

where $D(\alpha_{ij}, \tau_{ij})$ is an $l \times n$ mixing matrix with elements $d_{ij} = \alpha_{ij}\delta(t - \tau_{ij})$ and $\delta(t - \tau_{ij})$ is the sampling function. The operator $*$ denotes element-wise convolution in the same way matrix-vector multiplication is performed.

In particular, when s_j represents the signal event characteristic of the mode transition $q_j^{\tau_i} \rightarrow q_j^{\tau_i+1}$, the mode estimation problem is then to determine τ_{ij}, the onset of the signal event s_j, and α_{ij}, the contribution of s_j to the composite sensor output y_i. A common physical interpretation for the mixing parameters τ and α is that τ characterizes signal arrival time at each sensor, and α sensor gain for each sensor.

The following mode estimation algorithm computes $P(D(\alpha, \tau)|\mathbf{y}^t)$, the posterior probability distribution of τ and α given observation $\mathbf{y}^t$, iterating through the following three steps: (1) Use a model of system behaviors to generate a temporal prior $P(D(\alpha, \tau))$ of transition events within the time window associated with the current time step; (2) Decompose sensor observation as a sum of component signal events $\mathbf{y}^t = D(\alpha, \tau) * \mathbf{s}^t$, and compute the likelihood function $P(\mathbf{y}^t|D(\alpha, \tau))$; (3) Compute the posterior probability distribution of the mode transition $P(D(\alpha, \tau)|\mathbf{y}^t)$ using Bayesian estimation and update the mode vector. The algorithm is suited for a distributed implementation. Assume each node stores a copy of signal component templates $\hat{s_j}(t)$. At each step, a few global nodes broadcast the model prediction, and each node locally performs signal decomposition, likelihood function generation, and Bayesian estimation.

Mode Estimation Algorithm

Initialize $\mathbf{q}^0$;
for $n = 1, 2, \ldots,$
(1) **Prediction:**
$P(D(\alpha^n, \tau^n)) = ModelPrediction(\mathbf{q}^{n-1})$
(2) **Signal decomposition and likelihood generation:**
$\mathbf{r}^n(t) = \mathbf{y}^n(t) - \hat{\mathbf{y}}^n(t)$
where $\hat{\mathbf{y}}^n(t) = D(\alpha^n, \tau^n) * \hat{\mathbf{s}^n}(t)$;
$P(\mathbf{y}^n|D(\alpha^n, \tau^n)) = (2\pi)^{-\frac{l}{2}}|R|^{-\frac{1}{2}} exp\left(-\frac{1}{2}(\mathbf{r}^n)^T R^{-1}(\mathbf{r}^n)\right)$
where R is the covariance matrix for $\mathbf{r}^n$;
(3) **Update:**
$P(D(\alpha^n, \tau^n)|\mathbf{y}^n) \propto P(\mathbf{y}^n|D(\alpha^n, \tau^n))P(D(\alpha^n, \tau^n))$

[1] When the signals are nonlinearly superposed, then a nonlinear source separation method must be used.

$D(\alpha^n, \tau^n) = argmax_{(\alpha^n, \tau^n)} P(D(\alpha^n, \tau^n)|\mathbf{y^n})$
$\mathbf{q}^n = NextMode(D(\alpha^n, \tau^n), \mathbf{q}^{n-1})$
end

To address the problem of exponential blowup in data association described earlier, *ModelPrediction* uses a model to predict signal events that are present within a time window, thus focusing the signal event localization and association on just the predicted subset of events. A variety of models such as timed finite automata or Petri nets (Section 4.2) could be used to generate a prior. Other possible candidates include partially observable Markov decision processes and dynamic Bayesian nets suitably modified to encode both discrete and continuous variables. Signal decomposition and Bayesian estimation identify the signal events that are most likely present, thus eliminating the exponential factor in associating events with component modes. The tracking cost is linear in the number of measurements over time. *NextMode* updates the mode vector $\mathbf{q}$ with the identified mode parameters α and τ. Alternatively, instead of keeping only the most likely events according to posterior, the algorithm could be extended to maintain less likely events by propagating the full posterior distribution and using techniques such as backtracking [Kurien and Nayak, 2000] or smoothing [Lerner *et al.*, 2000] to manage the branching complexity.

The notation $\mathbf{y}(t)$ in the algorithm represents the observation within a time window of interest. In the signal decomposition $\hat{y_i}(t) = \sum_{j=1}^{n} \alpha_{ij} \hat{s_j}(t - \tau_{ij})$, $\{\hat{s_j} | j = 1, \ldots, l\}$, are the so-called signal event templates that characterize s_j's and are constructed from training data.

The model predicts what combinations of signal components are present (the α's) and how they are appropriately shifted (the τ's) within the time window of interest. Given the parameters α and τ, the likelihood functions for sensors are assumed to be independent of each other. Since each signal template has a non-zero finite length, it is necessary to account for adjacent signal events spilling from the previous time step into the current time window. Given an observation, the parameters τ^n and α^n are determined by maximizing the posterior in Bayesian estimation.

For simplicity, the likelihood functions are assumed to be Gaussian. For non-Gaussian, multi-modal priors and likelihood functions, techniques such as mixture models or particle filter (also known as sequential Monte Carlo or Condensation) could be used to represent and propagate probabilistic evidence.

The algorithm exploits a temporal prior to manage the computational complexity in mode estimation. Likewise, a spatial prior could also be exploited to associate each y_i with one or a small number of identifiable signal sources s_j's, using techniques such as beamforming in a multi-sensor system.

4 Experiment: An application to diagnosis of DC265 printer

We have prototyped a diagnosis system comprising three main components: timed Petri net model, mode estimation, and decision-tree diagnoser (Fig. 3).

Discrete-event data from built-in sensors and control commands of the printer are used to drive the Petri net model. The

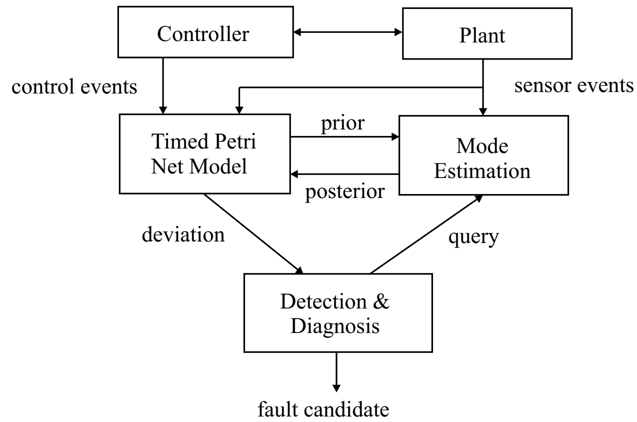

Figure 3: Architecture of the prototype diagnosis system.

model compares observed sensor events with their expected values. When a fault occurs, the deviation from the Petri net simulation triggers the decision-tree diagnoser. The diagnoser either waits for the next sensor event from the Petri net or queries the mode estimator for a particular event, depending on the next test. The mode estimator requests a temporal prior from the Petri net, uses the prior to retrieve the segment of the signal from appropriate sensors, and computes the posterior of the event. The Petri net uses the event posterior to update model parameters, generate a deviation of the event parameter for the diagnoser, and the process iterates until there are no more sensor tests to perform and the diagnoser reports the current fault candidates.

4.1 Experimental testbed

We have instrumented an experimental testbed, the Xerox Document Center 265ST printer (Fig. 1), with a multi-sensor data acquisition system and a controller interface card for sending and retrieving control and sensor signals. The monitoring and diagnosis experiment to be discussed in this section will focus on the paper feed subsystem (Fig. 4).

The function of the paper feed system is to move sheets of paper from the tray to the xerographic module of the printer, orchestrating a number of electro-mechanical components such as feed and acquisition rolls, feed motor, acquisition solenoid, elevator motor, wait station sensor, and stack height sensor. The feed motor is a 24V DC motor that drives the feed and acquisition rolls. The acquisition solenoid is used to initiate the feeding of the paper by lowering the acquisition roll onto the top of the paper stack. The elevator motor is used to regulate the stack height at an appropriate level. The wait station sensor detects arrival of the leading or trailing edge of the paper at a fixed point of the paper path. The stack height sensor is used to detect the position of the paper stack and controls the operation of elevator motor.

In the experimental setup, in addition to the system built-in sensors, audio and current sensors are deployed for estimating quantities not directly accessible (so-called virtual sensors [Sampath *et al.*, 2000]). A 14-microphone array is placed next to the printer. Ground return currents of various subsystems of the printer are acquired using three 0.22Ω inline resis-

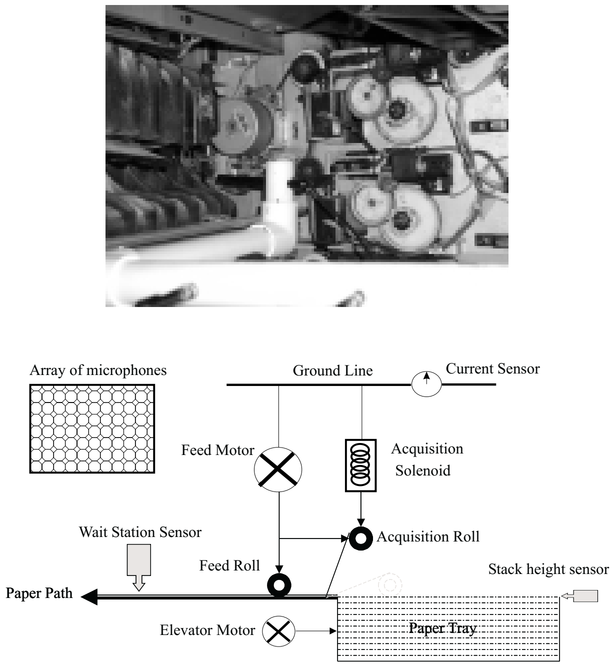

Figure 4: Paper feed system of Xerox DC265 printer

tors. Sensor signals are acquired at 40K samples/sec/channel and 16 bit/sample by a 32-channel data acquisition system.

The printer is designed such that control and sensor signals are passed between the controller and printer components through a common bus. By using an interface card these control and sensor signals can be accurately detected and mapped to the analog data acquired by the data acquisition system. Another controller interface card is used to systematically exercise components of the printer one at a time in order to build individual signal templates required by the mode estimation algorithm.

4.2 Prediction using a timed Petri net model

We use a timed Petri net to model temporal discrete-event behavior of hybrid systems instead of finite automata for the following reasons. First, Petri nets offer significant computational advantages over concurrent finite automata when the physical system to be modeled contains multiple moving objects. For example, it is desirable for a model of printer to compactly describe multiple sheets of paper and a variable number of sheets in a printing operation. Second, Petri nets can be used to model concurrency and synchronization in distributed systems very efficiently without incurring state-space explosion. Hybrid system models based on Petri nets have been developed, for example, in [Koutsoukos *et al.*, 1998].

The dynamics of a Petri net is characterized by the evolution of a marking vector referred to as the state of the net. The marking vector represents the mode of the underlying hybrid system and is updated upon firing of transitions synchronized with system events. In a timed Petri net, transition firings can be expressed as functions of time. A timed Petri net can be used to monitor a physical system by firing some of the transitions in synchronization with external events. In this case, a transition is associated with an external event that corresponds to a change in state of the physical system. The firing of the transition will occur when the associated event occurs and the transition has been enabled.

Here, we associate with each transition a firing time domain $[\tau_{\min}, \tau_{\max}]$. The transition is enabled when all its input places are marked, but the firing of the transition occurs at a specific time instant within the time domain. The advantage of this formalism is that it takes into consideration stochastic fluctuations in the time duration of physical activities in the system. If statistical information for the firings of the transition is provided, then the firing time domain can be augmented with a probability distribution characterizing the time instant the transition fires after it has been enabled. The model can be used to generate temporal prior probability distribution for the occurrence of autonomous events.

The Petri net model of the normal operation of the paper feed system is derived from the control specification of the system (shown in Fig. 5). Control commands issued by the controller and outputs of built-in sensors are external events for the appropriate transitions of the Petri net. For example, the transition labeled by "Ac_sl_on" corresponds to the event "acquisition solenoid on" and will fire when the controller issues a command to energize the solenoid if it is enabled. The transition labeled by "Dr_ac_rl" corresponds to the autonomous event "drop acquisition roll" that for the normal operation of the system should occur within a specified time interval $[\tau_{\min}, \tau_{\max}]$ from the time it was enabled. The transition labeled by "LE@S1" corresponds to the event the wait station sensor detects the leading edge of the paper and should also occur in a specified time interval. This time interval is derived using the motion dynamics of the paper according to the specifications. This transition is synchronized with the corresponding sensor signal from the physical system and is used to detect if the paper arrives late at the wait station sensor or does not arrive at all. This is accomplished by implementing a watchdog timer for the event based on the specifications of the paper feed system. It should be noted that the Petri net of Fig. 5 models the control logic of the paper feed system and can capture concurrent behavior for multiple sheets and multiple components in an efficient manner.

4.3 Diagnoser

The diagnostic process consists of the following two steps. First, a fault symptom table is generated by a simulation of the hybrid system model of the paper feed system that parameterizes both abrupt and gradual failures. Due to space limitations, interested readers should refer to [Koutsoukos *et al.*, 2001] for details of the fault parameterization and the fault symptom table generation. Alternatively, the fault symptom table could be derived from experimental methods such as FMEA when feasible. Second, a decision tree is compiled from the fault symptom table and it is then used as the diagnoser. The fault symptom table contains qualitative deviations of the sensor variables for different failure modes. Individual measurements are labeled as normal (0), above normal $(+)$, below normal $(-)$, maximum value $(\max)$, and minimum value $(\min)$. The minimum and maximum values are

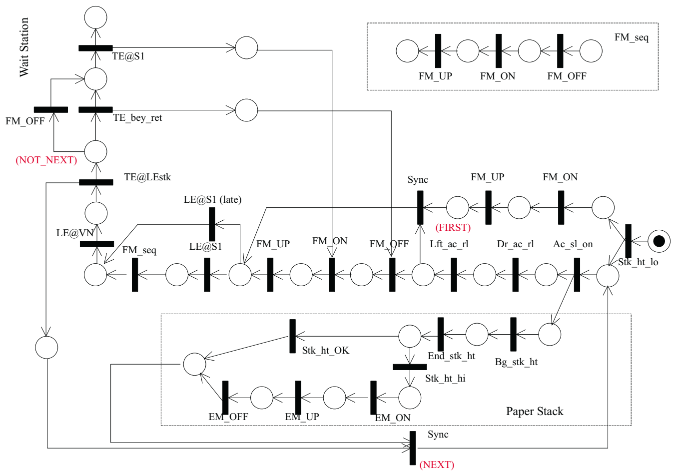

Figure 5: Petri net model of the paper feed system.

used to distinguish, for example, between a slow motor and a stalled motor. For real-time, embedded applications, the fault symptom table can be compactly represented by a corresponding decision tree using, for example, the ID3 algorithm [Quinlan, 1993].

In our diagnosis system we have two types of sensors, built-in sensors that are always accessible with a low cost and virtual sensors that cannot be used directly in the diagnoser but require the invocation of the mode estimation algorithm. Thus, the built-in sensors can be used for fault detection and trigger the diagnosis algorithm. The diagnoser will try to isolate the fault using only the built-in sensors. If this is not possible, then it will use virtual sensors. In order to take into consideration the sensor characteristics, we associate with the built-in sensors a cost equal to 0 and with the virtual sensors a cost equal to $K > 0$. The objective of the decision tree generation algorithm is to minimize the weighted cost of the tree $\sum_{L \in \text{leaves}} P(L) \sum_{X \in path(L)} C(X)$, where $P(L)$ is the probability of a fault or faults corresponding to leaf L of the tree and $C(X)$ is the cost of sensor test at node X of the path to L. A decision tree minimizing the weighted cost is generated by applying the ID3 algorithm in two phases. First, ID3 builds a tree using only the built-in sensors. Next, ID3 is applied to leaf nodes of the tree with more than one faults, and generates subtrees for those leaves using the virtual sensors (see Fig. 6).

4.4 Experimental Results

The diagnosis system of Fig. 3 has been demonstrated on four test fault scenarios, using the Petri net model of the paper feed system, the automatically generated decision tree, and the mode estimation algorithm. The system, implemented in MATLAB running on a Win2000 PC, sequentially scans prerecorded data streams to emulate online monitoring. The four test cases involve a feed roll worn fault (labeled as "8" in the decision tree of Fig. 6)), a feeder motor belt broken fault ("5"), an acquisition roll worn fault ("11"), and a motor slow ramp-up fault ("2"), and cover an interesting subset of system-

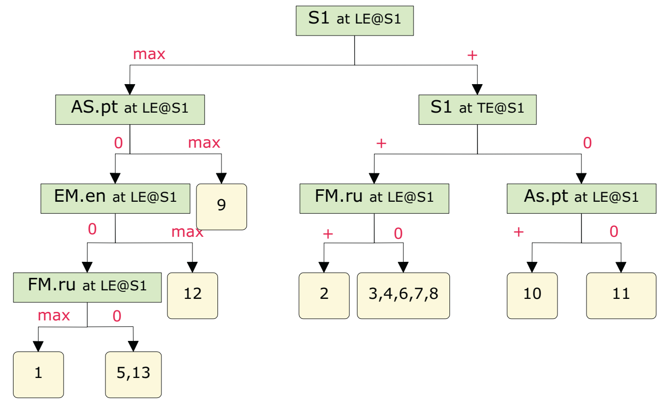

Figure 6: Decision tree for diagnosing faults in the paper feed system.

level faults of the printer. These faults may cause a delayed paper or no paper at subsequent sensors. Note the two "worn" cases are not directly observable. Our algorithm isolates the faults by reasoning across several sensor tests to rule out competing hypotheses using the decision tree. The motor slow ramp-up fault could be directly observed by the corresponding virtual sensor test only at the cost of substantial signature analysis. Instead, our algorithm uses less expensive system built-in sensors to monitor and detect faults and only invokes virtual sensor tests on a when-needed basis.

Let's examine the trace of the diagnosis output for one of the fault scenarios. The paper arrives late at wait station sensor LE@S1. The arrival time is compared with the expected time to generate a qualitative deviation "+", which triggers the diagnosis. The value of LE@S1 rules out faults such as drive train broken. Reading off of the decision tree, the next test TE@S1, trailing edge arrival time, is then invoked and returns normal ("0"). This rules out feed roll worn and motor slow ramp-up faults since both would cause the tailing edge late. Next on the decision tree, the more expensive acquisition solenoid pull-in time test (AS.pt) is invoked. This calls the mode estimation algorithm to determine the transition time at which the acquisition roll contacts the paper (or equivalently, solenoid pull-in), an autonomous transition event. The composite signal of one-page printing is shown in Fig. 2. The estimation uses acoustic and current signal templates of solenoid (Fig. 7) and motor (Fig. 8) to compute a posterior probability distribution of the pull-in event. Using the Petri net model prediction [495ms,505ms] to localize the event search, the estimation algorithm determines that the event is 2.5 ms later than the nominal value, well within the permissible range (see the peak location of posterior in Fig. 9). Therefore, AS.pt returns "0", and the only candidate remaining is the acquisition roll worn fault, which is the correct diagnosis. Physically, the reduced friction between the worn acquisition roll and paper causes the leading edge of the paper late at LE@S1. But this does not affect the trailing edge arrival time since the paper stops momentarily when the sensor detects the leading edge, and moves again without using the acquisition roll. In contrast, a worn feed roll would cause the trailing edge to be late.

The cost of the mode estimation algorithm scales linearly with the numbers of sensors and measurements when the mostly likely hypothesis is kept after each mode estimation. The cost of estimating α is exponential in the number of active component sources predicted by the model, since it has to check combinations of active sources present in the signal. Estimating τ employs a search for the maximum peak in the posterior in the mode parameter space. A brute-force search of the space is complete but at the cost exponential in the number of predicted active component sources. A gradient-decent search significantly speeds up the search and usually terminates within a small number of steps, but at the risk of possibly converging to local maxima. Experimentally, the diagnosis of the fault scenario described above was completed in 5 seconds for a sensor data sequence of 1.5 seconds in length.

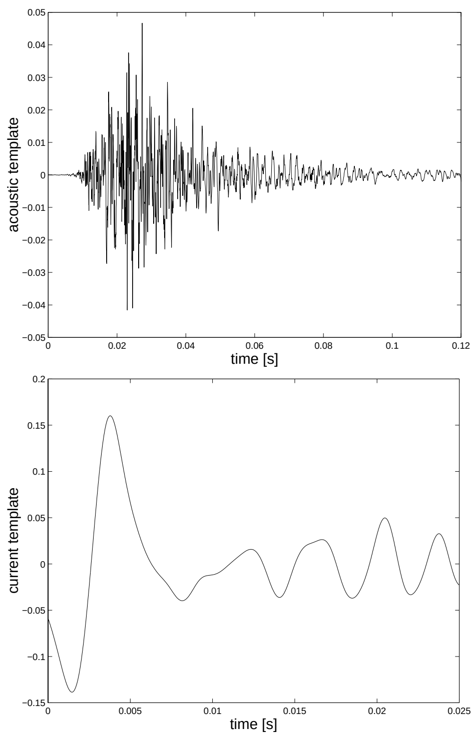

Figure 7: Acoustic and (high-pass filtered) current signal templates for AS_pull_in event.

5 Conclusions

This paper has presented a novel model-based mode estimation algorithm for monitoring and diagnosing multi-sensor distributed embedded systems. This work has demonstrated that monitoring of multi-sensor distributed hybrid systems can effectively exploit the knowledge of control and discrete-event behaviors of the system to drastically mitigate the exponential blowup due to the sensor data association problem.

There are a number of ways this work can be extended. The simple sensor cost function could be generalized to model more realistic distributed processing and communication characteristics in a distributed multi-sensor environment.

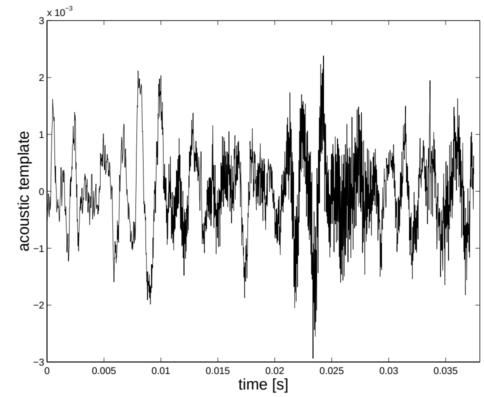

Figure 8: Acoustic signal template for FM_ramp_up event.

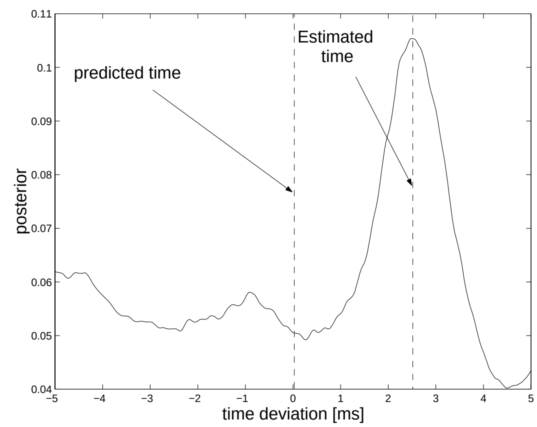

Figure 9: Posterior distribution of AS_pull_in time.

Currently, while mode estimation can be distributed, model simulation and diagnosis are performed centrally. Distributing the model and diagnostic reasoning would require maintaining and updating hypotheses on multiple nodes and remains as one of the topics for future research.

Acknowledgment

This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract numbers F33615-99-C-3611 and F30602-00-C-0139. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

We thank Sriram Narasimhan for his assistance in implementing a Petri net simulator during an internship at Xerox PARC, Bob Siegel for helping acquiring the test fixture, Meera Sampath, Ron Root, and Lloyd Durfey for their help in instrumenting the testbed, Johan de Kleer, Gautam Biswas and Bob Siegel for insightful discussions on diagnostics, and Jim Kurien for comments on drafts of the paper.

References

[Bar-Shalom and Fortmann, 1999] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1999.

[Hung and Zhao, 1999] E.S. Hung and F. Zhao. Diagnostic information processing for sensor-rich distributed systems. In *Proc. 2nd International Conference on Information Fusion (Fusion'99)*, Sunnyvale, CA, 1999.

[Koutsoukos *et al.*, 1998] X.D. Koutsoukos, K.X. He, M.D. Lemmon, and P.J. Antsaklis. Timed Petri nets in hybrid systems: Stability and supervisory control. *Journal of Discrete Event Dynamic Systems: Theory and Applications*, 8(2):137–173, 1998.

[Koutsoukos *et al.*, 2001] X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich, and P. Cheung. Fault modeling for monitoring and diagnosis of sensor-rich hybrid systems. Technical Report P2001-10039, Xerox Palo Alto Research Center, March 2001.

[Kurien and Nayak, 2000] J. Kurien and P. Nayak. Back to the future for consistency-based trajectory tracking. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'2000)*, 2000.

[Lerner *et al.*, 2000] U. Lerner, R. Parr, D. Koller, and G. Biswas. Bayesian fault detection and diagnosis in dynamic systems. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'2000)*, 2000.

[McIlraith *et al.*, 2000] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta. Hybrid systems diagnosis. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 282–295. Springer, 2000.

[McIlraith, 2000] S. McIlraith. Diagnosing hybrid systems: a bayesian model selection problem. In *Proceedings of the 11th International Workshop on Principles of Diagnosis (DX'2000)*, 2000.

[Quinlan, 1993] J.R. Quinlan. Combining instance-based and model-based learning. In *Proceedings of the 10th International Conference on Machine Learning*, 1993.

[Sampath *et al.*, 2000] M. Sampath, A. Godambe, E. Jackson, and E. Mallow. Combining qualitative & quantitative reasoning - a hybrid approach to failure diagnosis of industrial systems. In *4th IFAC Symp. SAFEPROCESS*, pages 494–501, 2000.