

# Monitoring and Fault Diagnosis of Hybrid Systems

Feng Zhao, *Member, IEEE*, Xenofon Koutsoukos, *Member, IEEE*, Horst Haussecker, *Member, IEEE*, Jim Reich, *Member, IEEE*, and Patrick Cheung, *Member, IEEE*

**Abstract**—Many networked embedded sensing and control systems can be modeled as hybrid systems with interacting continuous and discrete dynamics. These systems present significant challenges for monitoring and diagnosis. Many existing model-based approaches focus on diagnostic reasoning assuming appropriate fault signatures have been generated. However, an important missing piece is the integration of model-based techniques with the acquisition and processing of sensor signals and the modeling of faults to support diagnostic reasoning. This paper addresses key modeling and computational problems at the interface between model-based diagnosis techniques and signature analysis to enable the efficient detection and isolation of incipient and abrupt faults in hybrid systems. A hybrid automata model that parameterizes abrupt and incipient faults is introduced. Based on this model, an approach for diagnoser design is presented. The paper also develops a novel mode estimation algorithm that uses model-based prediction to focus distributed processing signal algorithms. Finally, the paper describes a diagnostic system architecture that integrates the modeling, prediction, and diagnosis components. The implemented architecture is applied to fault diagnosis of a complex electro-mechanical machine, the Xerox DC265 printer, and the experimental results presented validate the approach. A number of design trade-offs that were made to support implementation of the algorithms for online applications are also described.

**Index Terms**—Bayesian mode estimation, data association, hybrid systems, monitoring and diagnosis, printing systems.

## I. INTRODUCTION

**M**ANY 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. For example, an automobile can operate in multiple modes, such as the acceleration phase and the cruising phase. A printer may have a paper feeding phase followed by a registration phase. For each mode of operation, the system

dynamics is governed by a different continuous behavioral model. Control signals, such as gear shift, may transition the system from the current mode to a different operating mode. This is an example of a controlled transition. Other transitions are autonomous because they are governed by the values of internal state variables. For example, when a paper feed roll contacts a sheet of paper, a printer transitions to a new mode, where the paper starts to move.

A challenge addressed in this paper is modeling of both abrupt and incipient faults in hybrid systems. In a printer, for example, the operation can be interrupted either by abrupt failures such as a broken transfer belt or a stalled motor or by incipient faults describing subtle component degradation such as roll slippage or timing variations of clutches, motors or solenoids due to wear. Such events are not directly observable with the system's built-in sensors and must be estimated using system behavioral models and additional sensor information. Another problem addressed in the paper is monitoring of hybrid systems that consists of mode estimation and (continuous) state tracking. Once a system is estimated to be in a particular mode, a continuous state estimator such as a Kalman filter could be used to track the continuous state. This paper focuses on the difficult problem of mode estimation and its application to sensor-rich hybrid system monitoring and diagnosis.

The contributions of this paper are threefold.

- We present a fault modeling framework that is used to generate the online diagnoser. Faults affect the behavior of a hybrid system through both continuous and discrete dynamics as well as their interactions. We assume that there are no sensor faults and we present a framework for fault parameterization based on hybrid automata models that supports the diagnosis of both abrupt and incipient faults. Although we focus on hybrid automata with linear first-order dynamics, our printer example demonstrates that this class of hybrid systems can address realistic problems. We use the hybrid model to generate offline the fault symptom table by simulation which is then compiled into a decision tree which is used as the online diagnoser. The approach supports the diagnosis of single faults.
- We design an efficient online mode estimation algorithm for source separation and data association. The algorithm integrates model-based prediction with distributed signature analysis techniques. A timed Petri net model that is an abstraction of the hybrid system is used to represent temporal discrete-event behavior. The model generates event predictions that focus the signal processing algorithms. Mode estimation in turn refines and updates model states and event occurrences. The algorithm is novel in its use of model knowledge to drastically shrink the range

Manuscript received December 31, 2003; revised August 31, 2004 and January 26, 2005. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under contracts 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 U.S. Government. This paper was recommended by Associate Editor G. Biswas.

F. Zhao is with Microsoft Research, Redmond, WA 98052 USA (e-mail: zhao@microsoft.com).

X. Koutsoukos is with the Electrical Engineering and Computer Science Department, Vanderbilt University, Nashville, TN 37235 USA (e-mail: Xenofon.Koutsoukos@vanderbilt.edu).

H. Haussecker is with Intel Research, Santa Clara, CA 95054 USA (e-mail: horst.haussecker@intel.com).

J. Reich and P. Cheung are with the Palo Alto Research Center, Palo Alto, CA 94304 USA (e-mail: jreich, pcheung@parc.com).

Digital Object Identifier 10.1109/TSMCB.2005.850178

of a time-domain search for events of interest, and has been experimentally validated on a multisensor diagnostic test-bed.

- We develop an integrated software system architecture for online monitoring and diagnosis and experimental validation of a complex electro-mechanical machine, the Xerox DC265 multifunction printer. Our software architecture exemplifies two *generic principles* for the design of efficient diagnostic systems for networked embedded sensing and control systems: *model-driven signature analysis* and *utility-driven sensor querying*. The prototype diagnostic system demonstrates the integration of model-based and signature analysis techniques based on these principles. Finally, the experimental results illustrate the validity and the efficiency of our diagnostic approach.

The paper is organized as follows. Section II situates this work in the context of related approaches. Section III motivates this work using a significant problem from monitoring and diagnosis in a printer shop. Section IV overviews the diagnosis system and its main components. Section V presents the modeling framework and the offline design of the diagnoser. Section VI describes the online monitoring approach using a Petri net based monitoring algorithm. Section VII demonstrates the approach on the Xerox DC265 multifunction printer.

## II. RELATED WORK

Qualitative model-based diagnosis [1], [2] cannot isolate failures that manifest as small continuous variations in the plant's behavior, nor can they provide sufficient resolution to enable compensatory control of continuous degradations in the plant. These limitations render such discrete techniques ill-suited for diagnosis and control of many embedded systems, as demonstrated in practical applications [3]. Model-based approaches for continuous systems [4]–[8] are inappropriate for monitoring and diagnosing hybrid behaviors exhibited by many physical systems. These techniques typically derive a fault signature matrix based on analytical redundancy relations and compute fault statistics for raw sensor signals to form a diagnosis. The computation task may become prohibitively expensive for hybrid systems, which exhibit large numbers of possible mode transitions because models change with every mode transition, and the fault signature matrix has to be recomputed online.

An important class of approaches for hybrid systems diagnosis relies on discrete and/or temporal abstractions of the continuous dynamics. Hybrid diagnosis based on timed discrete-event representations has been presented in [9] where the continuous state is quantized and discrete methods are employed. The work of [10] extended TRANSCEND [11] for hybrid systems diagnosis. This framework uses qualitative algorithms for fault isolation based on temporal causal graphs. Our approach also uses discrete abstractions, namely a timed Petri net model and a decision tree diagnoser. One of the main differences is that we use the timed discrete event model to focus the signal processing of the discrete-time signal in order to estimate the mode of the system. The model is also used by the diagnoser to request sequences of sensor tests based on their discrimination power and computational cost. This is important for complex

hybrid systems, where the total number of modes and the cost of communication, sensing, and processing could be prohibitively high.

Another important direction of research is based on particle filtering methods. A particle filter approach to tracking multiple models of behaviors is described in [12]. Qualitative diagnosis techniques are used to provide a temporal prior to focus the sampling of particle filter consistent with the model prediction. Monitoring and diagnosis based on particle filtering has also been applied in [13] for a class of hybrid systems modeled by dynamic Bayesian networks, where the autonomous transitions between discrete states are only defined using the so-called soft-max conditional probability distributions. Another particle filtering approach for monitoring and fault detection of hybrid systems is presented in [14]. A distributed version of the algorithm applied to a cryogenic propulsion system can be found in [15]. A performance evaluation of the approach for hybrid systems with discrete sensors is presented in [16]. A similar approach has been applied for a planetary rover in [17]. The main weakness of such approaches is the computational complexity that limits their applicability to low-dimensional systems.

A different approach for diagnosing a special class of hybrid systems is presented in [18], where the fault hypotheses are modeled using a Markov chain with a Gaussian residual associated with each state and a Viterbi-like algorithm is used to find the most likely state trajectory. This approach does not consider the event-driven dynamics that are present in hybrid systems. A Bayesian network approach to tracking trajectories of hybrid systems is described in [19] based on a method of smoothing that backward propagates evidence to re-weight earlier beliefs so as to retain weak but otherwise important belief states without explicitly tracking all the branches over time.

A problem that is directly related to diagnosis is state estimation of hybrid systems. An approach based on banks of extended Kalman filters is presented in [20]. Keeping track of multiple models and the autonomous transitions between them is computationally very expensive, therefore, only a limited number of trajectories that have high probability are traced. A component-based framework for combining concurrent and continuous uncertain dynamical systems for simulation, state tracking, and diagnosis is presented in [21]. A state smoothing algorithm based on a moving horizon estimation of hybrid systems, modeled in the mixed logical dynamical form, is presented in [22]. Our work focuses only on mode estimation. Once an estimate of the mode is computed, a continuous state estimator such as a Kalman filter, can be used to track the continuous state. Also, our signature analysis approach considers multisensor systems with partially known observation models and signal mixing.

Finally, our Petri-net-based monitoring algorithm is motivated by the approaches presented in [23], [24] for fault monitoring of timed discrete event processes. We have modified these approaches to develop a monitoring method that can deal with continuous data streams, which contain signal contributions from multiple components and mode transitions. The approach has been demonstrated for the DC265 Xerox printer that is described in Section III.

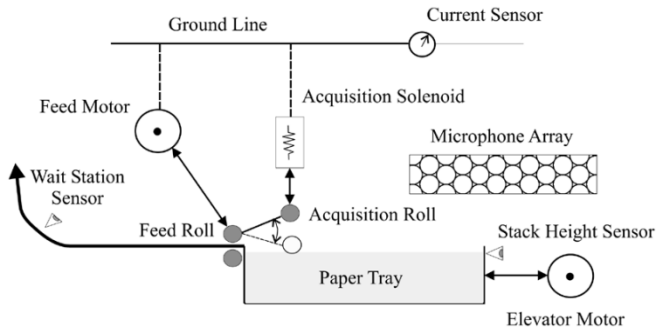


Fig. 1. Paper feed system of the Xerox DC265 printer.

### III. APPLICATION EXAMPLE

This work is motivated by the problem of work-flow 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. An example printer is the Xerox Document Center DC265 printer, a multifunction system that can print 65 pages per minute. The system is made of a large number of moving components that include motors, solenoids, clutches, rolls, gears, and belts. A fault, such as “no paper” may be caused by abrupt failures, such as a broken transfer belt, or as a result of subtler component degradation, such as roll slippage or timing variations of clutch, motor or solenoid operation caused by wear. None of these events are directly observable with the system’s built-in sensors.

The printer is an example of a hybrid system. For example, a component, such as the feed motor may be in the ramp-up, rotating with constant speed, ramp-down, and idle modes, each of which is governed by a different continuous model. Mode transitions can be attributed to control events, or continuous system variables crossing threshold values. For example, the transition from idle to ramp-up for the motor is caused by a “turn\_motor\_on” control event. However, a transition that represents the acquisition roll contacting the paper is autonomous, and must be estimated using model and sensor data. We have instrumented an experimental test-bed, the Xerox Document Center 265ST printer, with a multisensor data acquisition system and a controller interface card for sending and retrieving control and sensor signals. The monitoring and diagnosis experiments discussed in this section focus on the paper feed subsystem shown in Fig. 1.

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. The feed motor takes a 24-V dc input and drives the feed and acquisition rolls. The acquisition solenoid initiates the feeding of the paper by lowering the acquisition roll onto the top of the paper stack. The elevator motor regulates 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 on the paper path. The stack height sensor detects the position of the paper stack and controls the operation of the elevator motor.

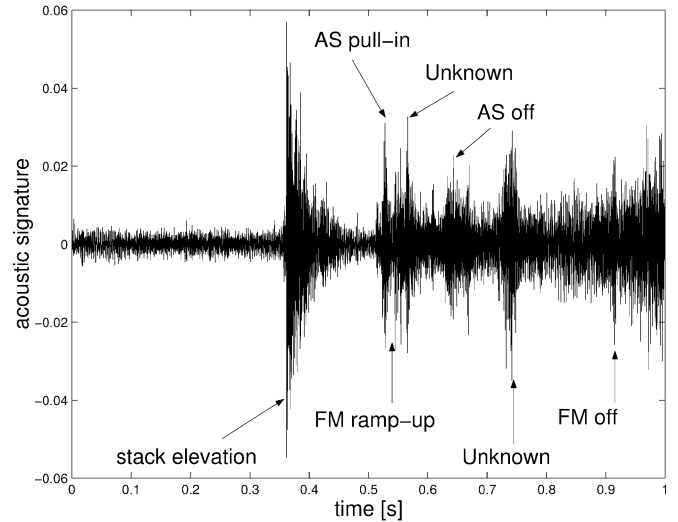


Fig. 2. Acoustic signal for a one-page printing operation of DC265 printer.

In the experimental setup, in addition to the built-in sensors, audio and current sensors are deployed for estimating events, which are not directly accessible otherwise. These sensors are called *virtual sensors* [25]. For example, estimating the time at which the acquisition roll contacts the paper can be estimated from audio and current data streams. We use a 14-microphone array placed next to the printer and three current sensors placed at the ground path of three printer subsystems. Ground return currents are acquired using three 0.22  $\Omega$  in-line resistors. All audio and current sensor signals are acquired at 40K samples/sec/channel and 16 bit/sample by a 32-channel data acquisition system. We assume that both the built-in and the current and audio sensors do not have faults during the experimental runs.

The control and built-in 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.

The paper feed system of the printer has three motors, ten solenoids, two clutches, and a large number of gears and belts connecting the motors to different rolls. Many of the system components may be active at the same time, and hence the current and audio measurements are the result of signal mixing—the so-called cocktail party phenomenon in speech processing, shown in Fig. 2. As the number of event hypotheses scales exponentially with the numbers of sensors, system components, and measurements (Section VI), *pulling the relevant events out of a large number of high-bandwidth data streams from a multitude of simultaneous sources is a significant computational challenge*. This is the main computational problem addressed in this paper. Signature analysis techniques, such as the one presented in [26] cannot be directly applied to the mode estimation task. A model is required to focus on when to acquire data and where to look for events.

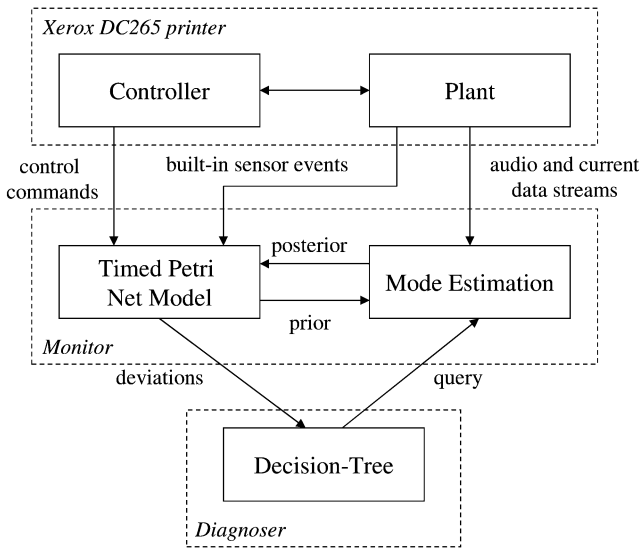


Fig. 3. Architecture of the prototype diagnosis system.

#### IV. ONLINE DIAGNOSTIC SYSTEM

We have developed an integrated software system for online monitoring and diagnosis of the Xerox DC265 printer. Our diagnostic system consists of two subsystems, the *monitor* and the *diagnoser*, as shown in Fig. 3. The objective of the monitor is to detect and estimate discrepancies between the actual and expected system behavior. The monitor employs a timed Petri net model and a mode estimation algorithm. The objective of the diagnoser is to isolate the fault based on the discrepancies between the actual and the expected measurements. The diagnoser is based on a decision tree that can be used efficiently for online diagnosis.

The Petri net has two main functions: 1) detecting faults based on the deviations between observed sensor events and their expected values and 2) providing prior probabilities to the mode estimation algorithm. Discrete-event data from built-in sensors and control commands of the printer are used to drive the Petri net model. The 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. This task is analogous to residual generation in observer-based diagnosis schemes. The Petri net also provides the temporal prior for the autonomous transitions that affect the fault to the mode estimation algorithm.

The mode estimation scheme estimates the occurrences of events that have to be estimated from measurements made by the virtual sensors. The mode estimation scheme requests a prior from the Petri net, which is used to retrieve an appropriate segment of the audio and current data stream in order to compute the posterior of the event. The Petri net uses this information to update model parameters and generate a deviation of the event parameter for the diagnoser. The process iterates until there are no more sensor tests to perform, then the diagnoser reports the current fault candidates.

The decision-tree based diagnoser is computed offline using a detailed hybrid system model of the paper feed system. The input to the decision-tree diagnoser consists of deviations between actual and expected measurements provided by the Petri

net model. Based on these discrepancies, the diagnoser either waits for the next sensor event from the Petri net or requests additional sensor tests from the mode estimation subsystem until the fault has been isolated. The task of the diagnoser is analogous to residual analysis in observer-based diagnosis.

Our prototype diagnostic system combines model-based diagnosis with statistical methods for signature analysis. Although, our implementation uses specific modeling formalisms and algorithms, the architecture is inspired by two generic principles, *model-driven signature analysis* and *utility-driven sensor querying*. Model-driven signature analysis drastically reduces the complexity of the estimation algorithms. In our approach, a timed Petri net model is synchronized with the actual system to detect deviations between the occurrences of the expected and actual events. The temporal behavior captured by the model provides a set of expectations for events and the time period they are supposed to occur, which are then used as prior in a Bayesian estimation algorithm. The timing of the events that are not directly observed are updated using the posterior of the estimation algorithm. Utility-driven sensor querying selects sequences of sensor tests based on their discrimination power and computational cost. Our diagnostic system uses a decision tree to capture the utility of the sensor tests. This is a fairly simple but computationally efficient way to query the sensors and enables online diagnosis for complex systems consisting of multiple components.

A timed Petri net model that abstracts away details of the continuous dynamics is used in the online monitor instead of the hybrid system model for computational reasons. The timed Petri provides the priors necessary for the estimation using asynchronous discrete event simulation. The design trade-off is that we cannot estimate continuous states such as the speed of a motor and therefore, we cannot distinguish between a “feed motor is slow” fault and a fault in the subsystem (clutch, belt, and gears) that transfers the drive force from the feed motor to the paper. The use of timed discrete event systems is reasonable since the estimation of the feed motor speed from the composite acoustic and current signals is a very difficult task that would require not only a more detailed model but also additional sensors. In the experimental test-bed, an optical encoder measures the feed motor speed. This measurement is not used by the diagnostic system but only for validation purposes since the cost of instrumenting all motors is prohibitive.

#### V. OFFLINE DESIGN OF THE DIAGNOSER

In this section, we present the design method for the diagnoser. The proposed method consists of three steps. First, we introduce a fault modeling formalism based on hybrid automata [27] for both abrupt and incipient faults. Second, we generate a fault-symptom table by simulation of the hybrid system model using specific values of the fault parameters chosen using domain specific knowledge (see Section V-B). Finally, we compile the fault-symptom table into a decision-tree that can be used efficiently as the online diagnoser. The diagnosability (the ability to discriminate among every fault pair) of the approach can be assessed using existing methods using the fault symptom table generated [28], [29].

### A. Modeling Faults in Hybrid Systems

We present a hybrid system model that contains all the features required for modeling the paper feed system. More general models can be found in [30] and the references therein.

A hybrid system is defined as  $H = (Q, X, \Sigma, Init, E, f, G)$  where  $Q$  is the set of discrete states or *modes* of the system,  $X$  is the continuous state space,  $\Sigma$  is a finite set of transition labels or *events*,  $Init \subseteq Q \times X$  is the set of initial conditions,  $E \subset Q \times \Sigma \times Q$  is the set of (controlled and autonomous) discrete transitions,  $f : \mathbb{R} \times Q \times X \rightarrow X$  is the flow condition for every mode defined by a differential equation, and  $G : E \rightarrow 2^X$  is a partial function that associates a guard condition (represented as a subset of  $X$ ) with each autonomous transition. The state of the hybrid system is described by the pair  $(q, x)$  where  $q \in Q$  and  $x \in X$ .

The behavior of the hybrid system is described by interleaving continuous evolution segments and discrete transitions. Continuous evolution corresponds to the progress of time and modifies the continuous state  $x$  according to the flow condition of mode  $q$ . Discrete (or mode) transitions change the mode  $q$  and are assumed to be instantaneous. Controlled mode transitions are induced by external control events (labeled by symbols from  $\Sigma$ ). Autonomous mode transitions are labeled by guard conditions  $G$  which are logical predicates over the continuous state space  $X$ . If the continuous state  $x$  satisfies the guard condition, then the system transitions to a new mode.

The acquisition solenoid, for example, is modeled as a hybrid system with the following modes: 1) idle; 2) pull-in; 3) roll-on-paper; and 4) pull-out. At each mode, the continuous dynamics are defined by a differential equation that describes the relative displacement of the acquisition roll. The transition from the mode “idle” to “pull-in” is triggered by an event issued by the controller. The transition from “pull-in” to “roll-on-paper” occurs when the acquisition roll contacts the paper. This transition is autonomous since it is triggered by a guard condition on the relative displacement of the roll (i.e., the continuous state).

The requirement for the hybrid system model is to capture all critical fault conditions for the paper feed system as derived a priori from reliability studies. We introduce three types of fault parameters for modeling incipient and abrupt faults. Fault parameters in the flow and guard conditions describe incipient faults and discrete failure modes describe abrupt faults. In the following, we consider the feed motor to motivate the fault modeling approach.

The most common incipient fault in a motor is a friction fault. Friction is modeled by introducing a fault parameter  $\theta_q \in \mathbb{R}$  in the differential equation of the mode  $q$ . To simulate the motor friction effects, the value of  $\theta_q$  is increased. The nominal condition with no motor friction implies  $\theta_q = 0$ .

In general, at every mode  $q$  we assume that the continuous dynamics are described by the parameterized system  $\dot{x} = f_q(x, \theta_q)$ , where the system’s behavior depends on the fault parameter  $\theta_q \in \Theta_q$  and  $\theta_{q0}$  represents the faultless system. Therefore, we use a finite set of subspaces  $\Theta_q$  representing the possible fault hypotheses  $\theta_q \in \Theta_q, q \in Q$ . This set of fault parameters is denoted by  $\Theta_1 = \bigcup_{q \in Q} \Theta_q$ .

Reliability studies for the printer have shown that motor aging affects also its steady-state angular velocity. The steady-state value is a reference signal used in a local PID controller. However, aging may slow down the motor causing an increasing steady-state error between the reference value and the actual angular velocity achieved by the motor. This incipient fault is a result of the discrepancy between the model used for designing the controller and the behavior of the degraded motor. In this case, the steady-state velocity of the motor is smaller than the nominal value, and therefore, the transition to the steady-state mode will occur at a lower speed. This is modeled by parameterizing the guard condition of the transition from the ramp-up to the steady-state mode using a fault parameter  $\theta_e \in \mathbb{R}$ .

In general, faults in the autonomous transitions are represented by parameterized guard conditions of the form  $G(x, \theta_e) \subset X \times \Theta_e$  where  $\theta_e \in \Theta_e$  is the fault parameter and  $\theta_{e0}$  describes the faultless system. Therefore, we use a finite set of subspaces  $\Theta_e$  representing the fault hypotheses  $\theta_e \in \Theta_e, e \in E$ . This set of fault parameters is denoted as  $\Theta_2 = \bigcup_{e \in E} \Theta_e$ .

In our approach, the model is used to generate the fault-symptom table using simulation. For generating the fault-symptom table we select a finite number of values for the fault parameters. The degradation of components such as the motor occurs very slowly in comparison with the printing operation. Further, for each page printed, the motor is activated for a short period of time (hundreds of msec). Thus, it is reasonable to assume that the value of the motor friction is constant for each print job. The values for these constants are selected from the specifications for normal arrival of the paper at the wait station sensor, late arrival, and no arrival (see Section V-B for details). A similar approach is used to model incipient faults of other components.

Finally, we introduce discrete states corresponding to faulty modes of the system that cannot be described by small deviations in the fault parameters. This modeling assumption arises naturally by the need to represent abrupt component failures. Reliability studies for the feed motor, for example, have shown that the most common abrupt fault is that the motor will not energize. Thus, we consider only one type of abrupt component faults that describe if the component works or not. Such abrupt faults are modeled as unobservable events that drive the system to the faulty modes. We assume that the set of modes of the hybrid system is partitioned as  $Q = Q_N \cup Q_F$  where  $Q_N$  and  $Q_F$  are the set of normal modes and faulty modes respectively. Similarly, we partition the set of transition labeling events as  $\Sigma = \Sigma_N \cup \Sigma_F$ . The set of failure events  $\Sigma_F$  labels transitions to faulty modes. Note that if information about the continuous dynamics for the faulty modes is available then a flow condition can be associated with these modes.

Let  $\epsilon$  denote the null event and  $\Sigma' = \Sigma_F \cup \{\epsilon\}$ . Then, the space of fault hypotheses for the hybrid system is defined as  $H = \Sigma' \times \Theta_1 \times \Theta_2$ . The null event,  $\epsilon$ , corresponds to the case when no discrete fault has occurred and it is included in  $H$  for describing hypotheses with only incipient faults. Fault hypotheses are described, by the function  $\mathbf{h}(\tau)$  where  $\mathbf{h} : \mathbb{R} \rightarrow H$  and  $\tau$  represents an independent time variable. The problem of

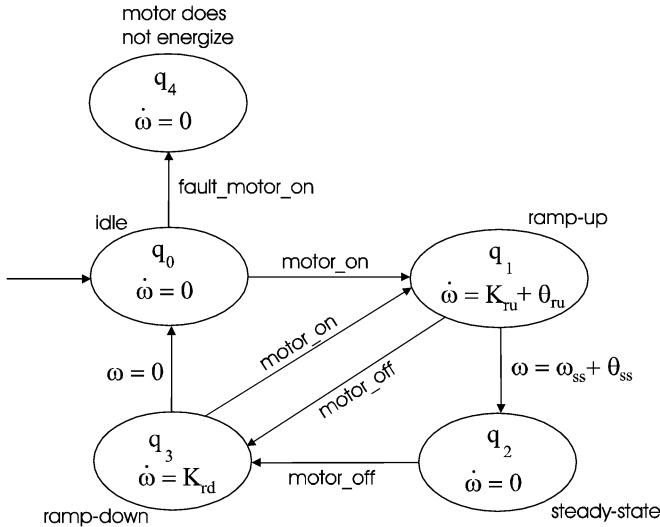


Fig. 4. Hybrid model for the feed motor.

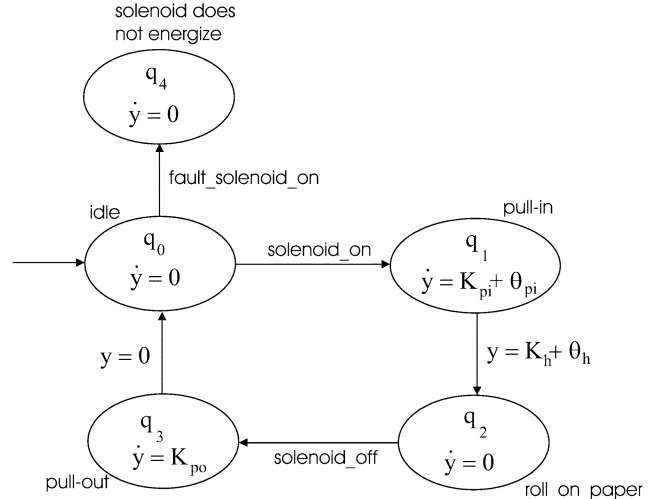


Fig. 5. Hybrid model for the acquisition solenoid.

hybrid system diagnosis is to find the most likely fault hypothesis  $h(\tau)$  for the observation history  $y(\tau)$ .

Complex systems like the DC265 printer consist of multiple components. Consider the set of components  $COMPS = \{c_1, \dots, c_N\}$  and assume that each component is modeled by a hybrid automaton. Then, the hybrid model of the printer is computed using the parallel composition of the individual hybrid automata [27]. In this case, the set of system modes  $Q$  can be understood as the product of individual component modes, i.e.,  $q = [q_1, \dots, q_N]^T \in Q$ . Similarly, for the continuous state we have  $x = [x_1, \dots, x_N]^T \in X$ . The set of events of the hybrid system can be written as  $\Sigma = \bigcup_{i=1}^N \Sigma^{(i)}$ , and the set fault hypotheses can be written as  $H = H_1 \times \dots \times H_N$  where  $H_i = \Sigma^{(i)} \times \Theta_1^{(i)} \times \Theta_2^{(i)}$ .

We present a hybrid model of the paper feed system. We consider only the feed motor, the acquisition solenoid, and a sheet of paper. Note that for our diagnostic system we also model the elevator motor that is used to place the paper stack at the correct position during the printing operation. The model of the elevator motor is similar to that of the feed motor and is omitted. The feed motor is a dc brushless motor controlled locally by a PID controller. Reliability studies have shown that the most common faults for the feed motor are the following: 1) the motor does not energize; 2) the nominal speed is not reached; and 3) it takes longer to ramp up. In our experiments, an external optical sensor was instrumented to measure the angular velocity of the motor to obtain “ground-truth”. This sensor is not used for diagnosis since it is very difficult and expensive to instrument all the motors of a printer in this way. The measurements obtained by this optical sensor show that the behavior of the motor and the local PID controller can be approximated by an integrator system with three distinct modes: ramp up, steady state, and ramp down. The behavior of the feed motor is captured in the hybrid automaton shown in Fig. 4.

Initially, the feed motor is idle and the angular velocity is  $\omega = 0$ . Upon receiving the control command “motor\_on”, the feed motor is ramping up according to the equation  $\dot{\omega} = K_{ru} + \theta_{ru}$ . The nominal behavior for the motor of the

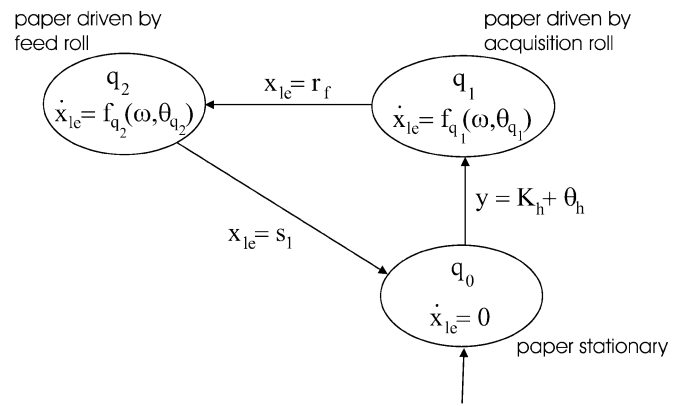


Fig. 6. Hybrid model describing the paper motion.

paper feed system is described by  $K_{ru} = 550 \text{ rad/s}^2$  and  $\theta_{ru} = 0$ , where  $\theta_{ru} \in \Theta_{ru} \subset \mathbb{R}$  parameterizes the acceleration, and therefore the “ramping-up” time of the motor. The transition from the ramp-up to the steady-state mode is labeled by the guard  $\omega = \omega_{ss} + \theta_{ss}$ . The nominal steady-state speed of the motor is  $\omega_{ss} = 16.5 \text{ rad/s}$  (and  $\theta_{ss} = 0$ ). Upon receiving a “motor\_off” control command, the motor ramps down ( $\dot{\omega} = K_{rd}$ ,  $K_{rd} < 0$ ) and returns to the idle position.

The most common faults for the acquisition solenoid are the following: 1) solenoid does not energize and 2) solenoid energizes slowly. The hybrid model of the solenoid describes its behavior using the relative displacement  $y$  of the acquisition roll that is attached to the solenoid. The hybrid automaton model is similar to the feed motor and is shown in Fig. 5.

A set of gears, belts, and clutches is used to transfer the drive from the feed motor to the feed and acquisition rolls that drive the paper. The motion of a sheet of paper in the paper path of the printer is described by the hybrid system shown in Fig. 6. The continuous state  $x_{le}$  represents the position of the leading edge of the paper. The modes for the paper motion correspond to the paper being stationary, and the paper being driven by the

acquisition roll or the feed roll. When the acquisition roll contacts the paper stack ( $y = K_h + \theta_h$ ), the top sheet starts moving toward the feed roll. As soon as the leading edge of the paper reaches the nip created by the feed roll ( $x_{le} = r_f$ ), the acquisition roll is lifted and the paper is driven by the feed roll. When the paper is driven by the feed roll, the paper motion is described by  $\dot{x}_{le} = \theta_1 \theta_2 R_{fr} \omega$ . The parameter  $\theta_1$  models the drive transfer from the feed motor to the feed roll through a set of belts, gears, and clutches and has nominal value  $\theta_{10} = 1$ . A common failure for the system is the degradation of the gears which affects the speed of the moving sheet and may result in paper jams. Such a degradation is represented in our framework by  $\theta_1 < 1$ . The parameter  $\theta_2$  represents the friction between the feed roll and the paper with nominal parameter  $\theta_{20} = 1$ . A roll that is worn will cause the paper to slip and may also lead to paper jams. Finally,  $R_{fr}$  is a constant that depends on the geometrical characteristics of the belt, the gears, and the rolls. Similarly, for the case when the paper is driven by the acquisition roll we have  $\dot{x}_{le} = \theta_1 \theta_3 \theta_4 R_{ar} \omega$ . Note that the acquisition roll is driven by the feed motor through the feed roll. Here,  $\theta_3$  represents the drive transfer from the feed roll to the acquisition roll, and  $\theta_4$  the friction between the acquisition roll and the paper. When the leading edge of the paper reaches the wait station sensor ( $x_{le} = s_1$ ) the feed motor is turned off and the paper stops.

The hybrid model of the paper feed system is derived using the parallel composition of the hybrid automata that model the feed motor, the elevator motor, the solenoid, and the paper motion. The mode  $\mathbf{q}$  of the overall system is the product of the component modes and the overall continuous state is  $\mathbf{x} = [\omega, y, x_{le}, \omega_{EL}]^T$  ( $\omega_{EL}$  denotes the angular velocity of the elevator motor). Therefore, the overall model has  $5 \times 5 \times 5 \times 3 = 375$  discrete modes and a four-dimensional continuous state space. The space of fault hypotheses for the paper feed system is the product of the fault hypotheses for the components. We have parameterized ten incipient and four abrupt faults (see Table I). However, as already explained in the diagnosis approach, we consider only single faults.

### B. Generation of the Fault-Symptom Table

The generation of the fault symptom table requires two steps: 1) computing a partition of the measurement and fault hypothesis spaces in order to discretize the continuous measurements and fault parameters and 2) simulating the hybrid system model to obtain a qualitative representation of the measurements. Let  $\tau$  denote the independent time variable. For given initial conditions, the state trajectory is denoted as  $(\mathbf{q}(\tau), \mathbf{x}(\tau))$  where  $\mathbf{q}(\tau)$  is a piecewise constant and  $\mathbf{x}(\tau)$  a piecewise continuous signal. For generating the fault symptoms, we consider a collection of measurements

$$y_i(\tau) = g_i(\mathbf{q}(\tau), \mathbf{x}(\tau)), \quad i = 1, \dots, l$$

where  $g_i$  describes the sensor model and we denote  $\mathbf{y} = [y_1, \dots, y_l]$ .

Our approach is based on a qualitative representation of the fault hypotheses and the sensor measurements. The abrupt fault

events are represented by the binary values “Y” and “N” (Yes, No) and the fault parameters  $\theta$  are labeled as normal (0), above normal (+), below normal (−), maximum value (max), and minimum value (min). The qualitative values were selected based on the frequent faults that appear in the printer. The (+) and (max) values are used to distinguish, for example, between the paper arriving late at the sensor and no paper at the sensor, respectively. The sensor variables  $y_i$  are also discretized and are represented appropriately either by qualitative values or binary values.

In the case when the continuous dynamics of the system are described by first-order integrators as in the paper feed system, a partition of the hypotheses space using rectangular constraints can be used to generate a fault symptom table. Assuming single faults, it is straightforward to show that the qualitative sensor values depend deterministically on the qualitative fault hypotheses. This follows from the fact that hybrid systems with first-order integrators, rectangular constraints, and without state jumps admit discrete abstractions that preserve reachability properties [31]. Therefore, after computing the partition the fault symptom table can be generated by simulation of the hybrid system model. Next, we focus on the paper feed system and illustrate the generation of the fault symptom table.

The most significant problems in the operation of the paper feed system are late and no arrival of the paper in the wait station sensor. In particular, there are two distinct events, the arrivals of the leading and trailing edge at the wait sensor that are used for the discretization of the fault hypotheses and the sensor measurements. According to the specifications of the printer, the paper is considered late at the wait station sensor if the (leading or trailing) edge arrives 30 ms after the expected time. Further, if the edge does not arrive within 100 ms after the expected time, a “no paper” fault is generated.

We have identified 14 faults that affect the arrival of the paper in the wait station sensor (Table I). All faults are linked to parameters in the hybrid system model. An abrupt fault is simulated by issuing the corresponding event. For the incipient faults, we first compute analytically the values of each fault parameter that result in late paper arrival and the “no-paper” fault. Note that this is straightforward for the hybrid model of the paper feed system since the continuous trajectories are piecewise linear. These values partition the fault hypothesis space into regions that are labeled as normal (0), above normal (+), below normal (−), maximum value (max), and minimum value (min) based on their effect on the paper arrival. An incipient fault is simulated by selecting an appropriate value of the corresponding fault parameter. Since the hybrid automaton model of the paper feed system contains only first-order linear dynamics and the fault parameters are either additive or multiplicative, the arrival times of the paper at the wait station sensor are monotone with respect to the continuous fault parameters. In addition, the abrupt faults will clearly cause the paper to stop and appropriate “no paper” events will be issued.

For the fault symptom table, we consider a group of sensor signals that can be measured either using the built-in sensors or the virtual sensors (the online estimation algorithm for the virtual sensors is presented in Section VI). The fault symptom table for the paper feed system is shown in Table I. Only the part that

TABLE I  
FAULTS FOR THE PAPER FEED SYSTEM

Id	Failure	Fault parameter	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
1	Feed motor does not energize	$\sigma_{f_{motor}}$	MAX	0	Y	N	MAX	MAX
2	Feed motor has high ramp up time	$\theta_{ru} = (-)$	+	0	Y	Y	+	+
3	Feed motor is slow	$\theta_{ss} = (-)$	+	0	Y	Y	0	+
4	Clutch has broken gears	$\theta_1 = (-)$	+	0	Y	Y	0	+
5	Belt is broken	$\theta_1 = (-)$	MAX	0	Y	Y	0	MAX
6	Belt is worn	$\theta_1 = (-)$	+	0	Y	Y	0	+
7	Gears are worn	$\theta_1 = (-)$	+	0	Y	Y	0	+
8	Feed roll is slipping	$\theta_2 = (-)$	+	0	Y	Y	0	+
9	Acquisition solenoid does not energize	$\sigma_{f_{solenoid}}$	MAX	MAX	Y	Y	0	MAX
10	Acquisition solenoid energizes slowly	$\theta_{pi} = (-)$	+	+	Y	Y	0	0
11	Acquisition roll is worn and slips	$\theta_4 := (-)$	+	0	Y	Y	0	0
12	Elevator motor does not energize	$\sigma_{f_{elev}}$	MAX	0	N	Y	0	MAX
13	Elevator motor is slow	$\theta_{el} = (-)$	+	0	Y	Y	0	0
14	No paper	$\sigma_{f_{paper}}$	MAX	0	Y	Y	0	MAX

affects the leading edge of the paper is shown due to space limitations. The columns of the table correspond to the quantized deviations of the sensor outputs from the nominal values. We consider the following sensor outputs: 1) the time  $y_1$  that the leading edge of the paper is detected by the wait station sensor; 2) the pull-in time  $y_2$  of the acquisition solenoid; 3) the measurement  $y_3$  that takes the value “Y” if the elevator motor energizes and “N” otherwise; 4) the measurement  $y_4$  that takes the value “Y” if the feed motor energizes and “N” otherwise; 5) the ramp-up time  $y_5$  of the feed motor; and 6) the time  $y_6$  that the trailing edge of the paper is detected by the wait station sensor.

We consider the case when the feed motor has high ramp-up time ( $\theta_{ru} < 0$ ) to illustrate our approach. The ramp-time  $\tau_{ru}$  represents the time interval in which the angular velocity of the feed motor increases from  $\omega = 0$  to  $\omega = \omega_{ss}$ . Assuming  $\tau_0 = 0$ , we have  $\tau_{ru} = \omega_{ss} / (K_{ru} + \theta_{ru})$ . The time instant  $\tau_{LE@S1}$  that the leading edge of the paper arrives at the location  $S_1$  of the wait station sensor is given by

$$\tau_{LE@S1} = \frac{S_1}{\theta_1 \theta_2 R_{fr} \omega_{ss}} + \frac{1}{2} \frac{\omega_{ss}}{k_{ru} + \theta_{ru}}. \quad (1)$$

The arrival time of the leading edge is clearly a decreasing function of the fault parameter  $\theta_{ru}$ . If  $\theta_{ru} < 0$  the paper will arrive at the wait sensor later than expected. To compute the values of the fault parameter  $\theta_{ru}$  that causes a late paper arrival or “no-paper”, we set  $\tau_{LE@S1} = \text{ExpectedTime}_{LE@S1} + 30$  ms and  $\tau_{LE@S1} = \text{ExpectedTime}_{LE@S1} + 100$  ms, respectively, and solve (1) with respect to  $\theta_{ru}$  to compute  $\theta_{ru}^{\text{late-paper}}$  and  $\theta_{ru}^{\text{no-paper}}$ . The fault for the feed motor with high ramp-up time

is simulated by selecting  $\theta_{ru} \in (\theta_{ru}^{\text{late-paper}}, \theta_{ru}^{\text{no-paper}})$  in the hybrid system model and is denoted by  $\theta_{ru} = (-)$  in the fault system table.

### C. Decision-Tree Diagnoser

For real-time, embedded applications, the fault symptom table can be compactly represented by a decision tree using, for example, the ID3 algorithm [32]. The use of a decision tree offers efficiency, however, it limits the approach to diagnosis of single faults. For applications like the printer, a fault cannot be masked by another fault since there are no faults that cause the paper to move faster and the probability of having two or more faults is very low.

The diagnoser receives as inputs two types of observations: 1) observations from built-in sensors which are always accessible with a low cost and 2) observations which are not directly sensed but are estimated by the mode estimation algorithm using the virtual sensors. The built-in sensors are used for fault detection and trigger the diagnosis algorithm. The diagnoser will initially 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 cost for using the sensors, 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

$$\sum_{L \in \text{leaves}} P(L) \sum_{X \in \text{path}(L)} C(X)$$



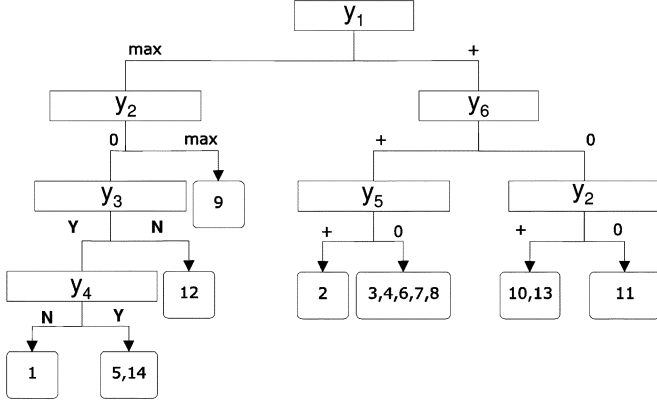


Fig. 7. Decision tree for the paper feed system.

where  $P(L)$  is the prior 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 fault, and generates subtrees for those leaves using the virtual sensors as shown in Fig. 7.

The diagnostic system traverses the decision-tree and requests sensor tests as indicated by its nodes. The monitoring algorithm computes the deviations between the expected and the actual sensor values. The process is continued until there are no more tests to perform and the diagnoser reports the fault candidates.

As it can be seen in Fig. 7, not all faults are diagnosable in our current system. The faults 5 and 14 (“belt is broken” and “no paper”) have exactly the same symptoms. The motor and the solenoid energize properly but no paper arrives at the wait station sensor. Currently this problem is addressed by instructing the user to check if there is paper in the tray. Diagnosing between faults  $\{3, 4, 6, 7, 8\}$  and  $\{10, 13\}$  requires the estimation of the speed of the motors in the steady-state mode, and this variable is not measured in our experimental setup.

## VI. ONLINE MONITORING AND DIAGNOSIS

Monitoring and diagnosis of systems such as the DC265 printer require estimating event occurrences such as the arrival of paper, the ramp-up of the motor, and the pull-in of the solenoid. Events related to the paper motion are directly observed by the built-in sensors. For the detection of the remaining events, we employ the virtual sensors using a Bayesian estimation method. A model of the system must be used to provide the prior probability necessary for the estimation. The estimation algorithm is computationally the most expensive software component of the diagnostic system since it employs the audio and current data streams. To perform online diagnosis, it is necessary to compute the prior quickly. For this reason, we use a timed Petri net model instead of the hybrid system model for monitoring the system. The Petri net model is a timed abstraction of the hybrid system model. The Petri net describes the normal operation of the system and it is used to detect deviations between the actual and expected events.

The basic idea is to synchronize the Petri net model with the actual system based on the controller commands, detect faults based on the sensor events, and provide the prior for the mode estimation algorithm. Thus, the Petri net provides the temporal prior for the estimation of the autonomous transitions. The prior distribution describes the expected time of occurrence for such a transition which corresponds to the nominal condition of the system as well as the time interval (support of the distribution) within which the transition may occur in the case of an incipient fault. Our approach does not assume any particular forms for the prior distributions, so parametric (e.g., uniform, normal, or exponential) or nonparametric distributions may be used.

### A. Prediction Using a Timed Petri Net Model

The Petri net generates a set of expectations describing events and the time periods during which are expected to occur. The set of expectations is maintained online by synchronizing the controller commands with the controlled transitions of the Petri net. Fault detection is performed by monitoring whether certain autonomous events occur within their expected time period. The expectations are also used as prior in the Bayesian mode estimation algorithm. New expectations are added for the events enabled by the controller commands and removed when the corresponding event occurs in the correct context. For example, an “acquisition\_solenoid\_on” event issued by the controller will generate an expectation for the “drop\_acquisition\_roll” event that will occur shortly after the control command. The estimation of the time stamp of this event is crucial for monitoring and diagnosis since the paper starts moving when it comes in contact with the acquisition roll and not when the controller issues the corresponding command.

Petri nets are used instead of automata based models for computational reasons. Petri nets offer significant computational advantages over concurrent automata when the physical system to be modeled contains multiple moving objects. For the printer, it is desirable to compactly describe the movement of multiple sheets of paper. Petri nets can be used to model concurrency and synchronization efficiently without incurring state-space explosion.

An ordinary Petri net structure [33] is represented by  $\mathcal{N} = (P, T, I, O)$  where  $P$  is a finite set of places,  $T$  is a finite set of transitions,  $I \subset P \times T$  is a set of input arcs (from places to transitions), and  $O \subset T \times P$  is a set of output arcs (from transitions to places). The marking of a Petri net is defined as a mapping  $\mu : P \rightarrow \mathbb{Z}^+$  from the set of places onto the nonnegative integers which assigns to each place  $p$  a number of tokens  $\mu(p)$ . We denote that a transition  $t$  may fire at marking  $\mu$  resulting in  $\mu'$  by  $\mu[t]\mu'$ . It is assumed that only a single transition can fire at any time instant (no concurrency assumption). A firing sequence from  $\mu_0$  is a sequence of transitions  $s = t_0 t_1 \dots t_k$  such that  $\mu_0[t_0]\mu_1 \dots [t_k]\mu_k$ .

In order to monitor the system, we label the transitions by events  $\sigma \in \Sigma$  using the labeling function  $\ell : T \rightarrow \Sigma$ . These labels are used to manage the transitions in all software components of the diagnostic system. The set of events is partitioned into controlled and autonomous events  $\Sigma = \Sigma_C \cup \Sigma_A$ . Controlled events describe the commands issued by the controller. The set of autonomous events is further partitioned into built-in

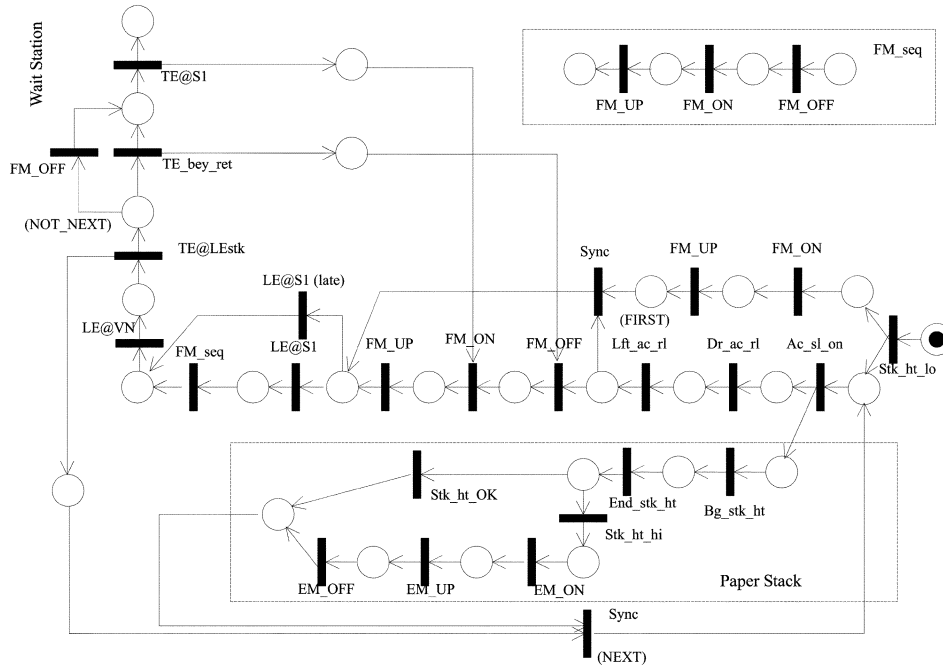


Fig. 8. Petri net model of the paper feed system.

sensor events and virtual sensor events that cannot be sensed by the built-in sensors but are important for monitoring the behavior of the system,  $\Sigma_A = \Sigma_B \cup \Sigma_V$ . For the paper feed system, the controller command “motor\_on” is modeled as a controlled event and the arrival of the leading edge of the paper at the wait station sensor “LE@S1” is modeled as a built-in sensor event. The event “drop\_acquisition\_roll” describes the time instant the acquisition roll comes into contact with the paper which is modeled as an event in  $\Sigma_V$ . We denote the occurrence of an event  $\sigma$  at time  $\tau$  by  $(\sigma, \tau)$  where  $\tau$  is understood with respect to a global clock.

The set of transitions is also partitioned into *controlled* and *autonomous* transitions  $T = T_C \cup T_A$  based on the labeling events. Controlled transitions are associated with commands issued by the controller and fire in synchronization with these commands. The occurrence of the transition  $t$  is denoted by  $(t, \tau)$  where  $\tau \in \mathbb{R}$  is the firing time. The firings of the autonomous transitions are not known a priori. Let  $\mathcal{I}$  be the set of closed intervals from the reals  $\mathbb{R}$ . We associate with each  $t \in T_A$  a *firing time domain*  $I = [\tau - \delta, \tau + \delta] \in \mathcal{I}$  where  $I$  is understood relative to the time the transition has been enabled. The firing of the transition is expected to occur at a time instant within the time domain.

The Petri net abstracts away the continuous dynamics and describes only the temporal discrete event evolution of the system. Hence, it can be used to provide the prior for the Bayesian mode estimation algorithm. We define the *mode*  $m \in M \subset 2^P$  of the Petri net as the set of places that are marked, that is  $m = \{p \in P : \mu(p) > 0\}$ . Consider the discrete state of the hybrid system model  $\mathbf{q} = [q_1, \dots, q_N]^T \in Q$ . Formally, the Petri net model is said to be a *timed abstraction* of the hybrid system if there exists a bijective mapping  $\Gamma : Q \rightarrow M$  and therefore, given  $m$  we can uniquely determine  $\mathbf{q}$  by  $\mathbf{q} = \Gamma^{-1}(m)$ . This requirement can be satisfied by ensuring that all the events of the hybrid system model are present in the Petri net model.

The Petri net model for the normal operation of the paper feed system is shown in Fig. 8. Control commands issued by the controller are synchronized with 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. The transition labeled by “Dr\_ac\_rl” corresponds to the autonomous event “drop\_acquisition\_roll” that for the normal operation of the system may occur within a time interval 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. This is an event that is used for fault detection. The timed Petri net model of the paper feed system was generated manually and it can be shown by enumeration of the modes that it is a timed abstraction of the hybrid system model (for the nominal system operation).

Next, we describe how fault detection and monitoring is performed using the Petri net model. For a transition  $t_0$ , we define a *consequence* as  $(t, I)$  where  $t \in T_A$  is an autonomous transition that is enabled by the firing of  $t_0$  and  $I = [\tau - \delta, \tau + \delta] \in \mathcal{I}$  is the firing time domain. A *consequence sequence* of  $t_0$  is defined as

$$CS(t_0) = \{(t_j, I_j)_{j=1, \dots, k}, \quad t_j \in T_A : \mu_0[t_0] \mu_1[t_1] \\ \mu_2 \dots [t_j] \mu_j \text{ and } t_i \in T_A, 0 < i < j\}.$$

Note that a transition may have more than one consequence sequences and also a transition may be contained in more than one consequence sequence. An *expectation* is defined as  $(\tau_0, t_0, C)$  where  $\tau_0 \in \mathbb{R}$  is the firing time of the transition  $t_0$ , and  $C$  is the set of consequence sequences of  $t_0$ . The objective of the Petri net is to maintain online the current set of expectations  $\mathcal{E}$ .

If no fault has been detected, the timed Petri net model monitors the system by synchronously firing the controlled transitions and the controlled commands, and simulating the firings

of the autonomous transitions asynchronously. It is assumed that the control commands are nonfaulty. Each autonomous transition  $t$  with consequence  $(t, I)$ ,  $I = [\tau - \delta, \tau + \delta]$  is assumed to fire at time instant  $\tau$ . The new state is computed by updating the marking and the global clock of the Petri net.

In the printer, fault detection is based on the arrival time of the paper, which is detected by the built-in sensors. In more detail, fault detection is performed by monitoring the set of consequences  $(t_k, I_k)$  where  $\sigma_k = \ell(t_k)$  is a sensor event  $\sigma_k \in \Sigma_B$ , that is comparing the actual occurrence of each sensor event with the expected one described by the corresponding occurrence. Consider the consequence sequence  $CS(t_0) = \{(t_j, I_j)_{j=1, \dots, k}\}$  where  $t_0$  is the latest controlled transition that affects the firing of  $t_k$ . We say that the consequence  $(t_k, I_k)$  with  $I_k = [\tau_k - \delta_k, \tau_k + \delta_k]$  is *satisfied* if the event  $\sigma = \ell(t_k)$  occurs at time  $\tau \in \tau_0 + I_1 \oplus \dots \oplus I_k$  where  $\oplus$  indicates interval addition<sup>1</sup>. If a consequence is not satisfied, i.e., it is *violated*, the monitoring algorithm signals a fault, computes the qualitative value of the deviation  $\tau - \tau_k$ , and invokes the decision tree diagnoser to isolate the fault.

Since fault detection is based on interval addition, the time interval for detecting a sensor event increases and therefore, the discriminatory power of the corresponding consequence decreases as the number of autonomous transitions in a consequence sequence increases. However, in this work the growth of the interval is curtailed by synchronizing the controlled transitions with the commands issues by the controller, which are assumed to be nonfaulty.

After a fault has been detected, the current set of expectations  $\mathcal{E}$  provides the prior probabilities to the mode estimation algorithm for all the autonomous events that may occur between the control commands and the fault. All autonomous transitions that are affected by the fault have to be estimated in order to determine if the transitions fired and if so, compute the most likely firing times. These transitions are contained in the set of consequence sequences. The task of the monitoring algorithm after a fault occurrence is to update the consequence sequences with the estimated firing times of the autonomous transitions and maintain a valid set of expectations. The transitions are estimated in chronological order. Suppose that the first autonomous transition in the consequence sequences of  $C$  is  $t_1 \in CS(t_0)$ . The mode estimation algorithm (presented in Section VI-B) estimates the most likely firing time of  $t_1$ , say  $\tau_1$ . Then the monitoring algorithm updates the set of expectations  $E$  by replacing  $CS(t_0)$  by  $CS(t_1)$ . This process continues until all autonomous transitions are estimated. The Petri-net-based monitoring algorithm is summarized in the following.

#### Timed Petri Net Monitoring Algorithm

```

Initialize  $\mathcal{E} = \emptyset$ 
if no Fault
  for each new event  $(\sigma, \tau)$ 
    if  $\sigma \in \Sigma_C$  and  $\sigma = \ell(t_0)$ 
       $C := \bigcup CS(t_0)$ 
       $\mathcal{E} := \mathcal{E} \cup (\tau_0, t_0, C)$ 
      Simulate  $(t_0, \tau_0)$  and every  $(t_i, I_i) \in C$ 

```

<sup>1</sup>Given two intervals  $I_1, I_2$ ,  $I_1 \oplus I_2 = \{\tau_1 + \tau_2 | \tau_1 \in I_1, \tau_2 \in I_2\}$ .

```

  if  $\sigma \in \Sigma_B$  and  $\sigma = \ell(t_k)$ 
    if  $(t_k, I_k)$  is satisfied and  $t_k \in CS(t_0)$ 
       $\mathcal{E} := \mathcal{E} \setminus (\tau_0, t_0, CS(t_0))$ 
    if  $(t_k, I_k)$  is violated
      declare Fault
      invoke DecisionTree(deviation)
end
if Fault
  for each new event  $(\sigma, \tau)$ 
    if  $\sigma \in \Sigma_A$  and  $\sigma = \ell(t_1)$ 
       $C''' := (C \setminus CS(t_0)) \cup CS(t_1)$ 
       $\mathcal{E} := (\mathcal{E} \setminus (\tau_0, t_0, C)) \cup (\tau_1, t_1, C''')$ 
end

```

#### B. Mode Estimation

After a fault has been detected, the decision-tree diagnoser requests additional sensor tests to be performed in order to isolate the fault. These sensor tests include algorithms for estimating the occurrence of autonomous events that are not measured directly but must be estimated using the data streams of the audio and current sensors. Such autonomous events are associated with mode transitions in the Petri net and therefore, the hybrid system model. Hence, the problem of estimating such events is equivalent to estimating the mode transitions times of the hybrid system model. The mode estimation algorithm is formulated with respect to the mode  $m \in M$  of the timed Petri net. By construction of the Petri net model, this is equivalent to estimating the mode  $\mathbf{q} \in Q$  of the hybrid system model.

We associate with the system the following mode transition sequence  $(m_0, \tau_0), (m_1, \tau_1), \dots, (m_k, \tau_k)$ , where  $(m_k, \tau_k)$  means that  $m_k$  is the system mode for  $\tau_k \leq \tau < \tau_{k+1}$  assuming that  $m_k \neq m_{k+1}$  and  $\tau_0 < \tau_1 < \dots < \tau_k$ . The objective of the algorithm is to estimate the mode transitions from the observed data. For an  $l$ -sensor system, the sensor output vector is the discrete-time signal  $\mathbf{y}(n)$ ,  $n \in \mathbb{N}$  with sampling period  $T_s$  where  $\mathbf{y}(n) = [y_1(n), \dots, y_l(n)]^T \in Y$  and  $y_i(n)$  is the output of sensor  $i$  at time  $\tau = nT_s$ . The data stream of observations up to time  $nT_s$  is denoted as  $\mathbf{y}[n]$ .

The mode estimation can be viewed as the mapping  $\mathcal{O} : M \times \mathbb{R} \times Y \rightarrow M \times \mathbb{R}$ , meaning that the mode estimation problem is to compute  $(m_k, \tau_k) = \mathcal{O}((m_{k-1}, \tau_{k-1}), \mathbf{y}[n])$ , that is find the time  $\tau_k$  of the mode transition and the next mode  $m_k$  given the previous mode  $m_{k-1}$  transitioned at  $\tau_{k-1}$  and the observed data stream  $\mathbf{y}[n]$ . The main advantage of using the timed Petri net model is to focus the mode estimation algorithm to look for signatures of interest. This is achieved by considering observations only for a time interval that contains all the events that are expected to affect the mode transition.

For multicomponent systems, mode estimation is particularly challenging due to the *data association* problem. Consider a hybrid system consisting of  $N$  components  $COMPS = \{c_1, c_2, \dots, c_N\}$  with mode vector  $\mathbf{q} = [q_1, q_2, \dots, q_N]^T$ . Note that the mode space  $M$  of the Petri net can be also understood as the product of individual component modes of an  $N$ -component system. Further, let  $z_j$  be the signal contribution of component  $c_j$  to the sensor output. Each  $y_i$  could be a measure of a signal  $z_j$  from component  $c_j$  alone or a composite

signal of multiple components. Estimating  $(m_k, \tau_k)$  requires solving the data association problem where each sensor output  $y_i$  must be associated with the component  $c_j$ . We illustrate the *computational difficulties* of data association for the hybrid system mode estimation problem for two cases.

1) Assume there is no signal mixing and each  $y_i$  measures a signal  $z_j$  from system component  $c_j$  only. The number of possible associations of  $y_i$ 's with the corresponding  $c_j$ 's is  $N^l$ , that is, it is exponential in the number of sensors at each time step.

2) More generally, each sensor signal  $y_i$  measures a composite of  $z_j$ 's through a mixing function  $\mathcal{H}$ . Without prior knowledge about  $\mathcal{H}$ , any combination of  $z_j$ 's could be present in  $y_i$ 's. Pairing each  $y_i$  with  $z_j$ 's creates  $N!$  associations. The total number of associations of  $\mathbf{y}$  with  $\mathbf{q}$  (and therefore  $m \in M$ ) is  $(N!)^l \propto 2^{Nl}$ , that is 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 that a component fault occurred in an earlier mode. Each pairing of observations with components in each time step creates a hypothesis for the 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. We address this problem using the temporal prior provided by the Petri net model in order to reduce the number of sensors and measurements required for estimating the mode transitions.

The mode estimation algorithm is invoked by a query from the diagnoser with a request for estimating the occurrence of a particular autonomous event  $(\sigma_k, \tau_k)$  in order to reduce the computational complexity of the estimation. The algorithm also receives as input the current set of expectations  $\mathcal{E}$  maintained by the Petri net model. The event  $(\sigma_k, \tau_k)$  is associated with the mode transition  $m_{k-1} \rightarrow m_k$  by using  $\sigma_k$  to label a Petri net transition  $t_k$ . Since  $t_k$  is autonomous the corresponding consequence  $(t_k, I_k)$  is contained in  $\mathcal{E}$ . The algorithm estimates the actual firing time  $\tau_k \in I_k$  of this transition. To achieve this task, the algorithm must estimate all the predecessor transitions of  $t_k$ . The task is further complicated by the fact that the current set of expectations  $\mathcal{E}$  may contain multiple consequence sequences. These challenges are addressed by recursively estimating all the transitions in  $\mathcal{E}$  while taking into account overlapping transitions from multiple consequence sequences.

Estimation of mode transitions is based on the data streams received by the sensor array. Each transition is associated with a data stream, denoted by  $\mathbf{z}[n]$ , of duration  $\Delta$ . In the paper feed system, for example, based on training data it can be shown that the transition described by the event "drop\_acquisition\_roll" is associated with a data stream  $\mathbf{z}_{\text{drop\_acquisition\_roll}}[n]$  with duration  $\Delta = 25$  ms consisting of current signal components and acoustic signal components). This association means that the contribution of the event occurrence to the sensor measurements is described by  $\mathbf{z}_{\text{drop\_acquisition\_roll}}[n]$  and it has duration 25 ms.

Since the firings of transitions may overlap, we have to consider the so-called cocktail party phenomenon, where each sensor output is a result of mixing of the individual signal components. Suppose that in the current set of expectations,  $\mathcal{E}$ , there exist totally (after renumbering)  $j = 1, \dots, \nu$  transitions

to be estimated. The data stream of each transition is denoted by  $\mathbf{z}_j[n] = [z_{j1}[n], \dots, z_{jl}[n]]^T$  where  $z_{ji}[n]$  is the signal component contributed to the sensor  $i$  by the firing of the transition  $t_j$ . Assuming each sensor output can be written as a linear superposition<sup>2</sup> of possibly time-shifted  $z_{ji}$ 's

$$y_i(n) = \sum_{j=1}^{\nu} \lambda_{ij} z_{ji}(n - s_{ij}) = \sum_{j=1}^{\nu} \lambda_{ij} \delta_{ji}(n - s_{ij}) * z_{ji}(n), \quad i = 1, \dots, l \quad (2)$$

where  $\lambda_{ij} \in \{0, 1\}$  represents whether the transition  $t_j$  contributes to the composite sensor output  $y_i$  (whether  $t_j$  fires or not),  $s_{ij}$  is the onset of the signal component  $z_{ji}$  that represents the signal arrival time at the  $i$ th sensor,  $\delta(n)$  is the sampling function (with sampling period  $T_s$ ), and  $*$  denotes the convolution operator. For small distances between the signal sources and the sensors as in the printer test-bed, the travel time of the signal  $z_{ji}$  is assumed to very small and therefore,  $s_{ij}$  represents accurately the firing time of the transition  $t_j$ . Equation (2) can be written more compactly as

$$\mathbf{y}(n) = H_n(\Lambda, S) * \tilde{\mathbf{z}}(n). \quad (3)$$

$H_n(\Lambda, S)$  is an  $l \times (\nu l)$  mixing matrix that is a function of the sensor gains  $\Lambda = [\lambda_{ij}], i = 1, \dots, l, j = 1, \dots, \nu$  and the signal onsets  $S = [s_{ij}], i = 1, \dots, l, j = 1, \dots, \nu$  of the form

$$H_n(\Lambda, S) = \begin{bmatrix} h_n^1(\Lambda, S) & 0 & \dots & 0 \\ 0 & h_n^2(\Lambda, S) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h_n^l(\Lambda, S) \end{bmatrix}$$

where  $h_n^i(\Lambda, S)$  is an  $\nu$ -dimensional row vector with elements  $\lambda_{ij} \delta(n - s_{ij}), j = 1, \dots, \nu$ . The signal components  $z_{ji}$  are contained in the  $\nu l$ -dimensional column vector  $\tilde{\mathbf{z}}(n) = [z_{11}(n), \dots, z_{\nu 1}(n), \dots, z_{1l}(n), \dots, z_{\nu l}(n)]^T$  and  $*$  denotes matrix-vector convolution similarly to matrix-vector multiplication.

The mode estimation algorithm is based on a Bayesian approach where the objective is to compute the parameters  $\Lambda$  and  $S$  that maximize the posterior probability distribution  $P(\Lambda, S | \mathbf{y}[n])$ . This is a nonlinear estimation problem since the matrix  $H_n(\Lambda, S)$  is a nonlinear function of  $S$ .

**Prediction:** The first step in the algorithm is to compute the prediction for the parameters  $\Lambda$  and  $S$ . This prediction describes what combinations of signal components are expected to be present ( $\lambda_{ij}$ ) and how they are shifted ( $s_{ij}$ ) within the time window of interest. The prior probability distributions for individual faults are computed using training data from reliability studies and testing. During the run-time phase of the algorithm these distributions are combined based on the information provided by the Petri net model. Suppose that at the  $j$ th step of the algorithm we want to estimate the occurrence of transition  $t_j$  with firing time domain  $I_j$ . From the set of expectations provided by the Petri net model, we consider every autonomous

<sup>2</sup>When the signals are nonlinearly superposed, then a nonlinear source separation method must be used.

transition  $t_\kappa$  with firing time domain  $I_\kappa$  that overlaps with  $I_j$ . The transitions are ordered chronologically and indexed by  $\kappa = 1, \dots, j$ . Since each signal component  $z_{ji}[n]$  has a nonzero finite length, it is necessary to account for adjacent events spilling into the current time window. Therefore we set the time window of interest to be  $W_j = \bigcup_{\kappa=1}^j I_\kappa$ . The prediction contains the probability that a transition  $t_\kappa$  does not fire within  $W_j$  represented by  $\lambda_{i\kappa} = 0$ . Also the probability that  $t_\kappa$  fires at time  $s_{i\kappa}T_s$  in  $W_j$  and therefore contributes to the sensor output  $y_i$ . We denote this prediction by the joint probability distribution  $P(\Lambda, S)$ .

**Likelihood generation:** The second step is the computation of the likelihood function. To perform this step, we use training data to generate the set of signal event templates  $\hat{z}_j[n]$  that characterize the firing of each transition. Practically, we activate each system component individually multiple times. After filtering and de-noising, we average the sensor outputs to create a template  $\hat{z}_{ji}[n]$  of duration  $\Delta_j$  for each sensor output  $i$  and autonomous transition  $t_j$ . Given parameters  $\Lambda$  and  $S$ , the predicted sensor vector output is  $\hat{\mathbf{y}}(n) = H_n(\Lambda, S) * \hat{\mathbf{z}}(n)$ . The likelihood functions for the sensors are assumed to be independent of each other. For simplicity, we assume the Gaussian likelihood function

$$P(\mathbf{y}[n]|\Lambda, S) = (2\pi)^{-(l/2)} |R[n]|^{-(1/2)} \exp\left(-\frac{1}{2} \mathbf{r}^T[n] R^{-1}[n] \mathbf{r}[n]\right)$$

where  $\mathbf{r}[n] = \mathbf{y}[n] - \hat{\mathbf{y}}[n]$  is the residual signal with covariance matrix  $R[n]$  and all signals are assumed to be zero outside the time window  $W_j$ . For non-Gaussian multimodal priors and likelihood functions, techniques such as mixture models or particle filter could be used.

**Posterior update and parameter estimation:** In the final step of the algorithm, we update the posterior distribution applying the Bayes rule

$$P(\Lambda, S|\mathbf{y}[n]) \propto P(\mathbf{y}[n]|\Lambda, S)P(\Lambda, S)$$

and we compute the parameters  $\hat{\Lambda}$  and  $\hat{S}$  that maximize the posterior by solving the following optimization problem:

$$\max_{\Lambda, S} P(\Lambda, S|\mathbf{y}[n]).$$

In general, this is a multidimensional optimization problem. A brute-force search of the space is complete but has exponential cost in the number of predicted active component sources since it all combinations of active sources may be present in the signal. Estimating the actual firing times employs a search for the maximum peak in the posterior. A gradient-descent 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 since the posterior is not necessarily convex (see Fig. 12). For the estimation algorithm to run online, we assume that only the firing of a single autonomous transition can vary and we solve multiple one-dimensional optimization problems. It should be noted that this assumption limits our approach to diagnosis of single faults.

The parameters  $\hat{\Lambda}$  and  $\hat{S}$  describe the most likely firing of the transitions and are used to compute the next mode of the system  $(m_k, \tau_k)$ . The information about the occurrence of the transition is then send back to the Petri net model that revises the set of expectations by updating the firing time of the estimated transitions (as discussed in the Petri net monitoring algorithm).

The mode estimation algorithm starts its execution upon receiving a request for estimating a particular event from the diagnoser. First, it recursively estimates all the transitions leading to this event as provided by the Petri net model. After it estimates the event of interest, it returns its firing time (or no firing time) to the Petri net and terminates waiting for another query. The algorithm is summarized in the following.

#### Mode Estimation Algorithm

```

Initialize Transition  $(t_k, \tau_k)$  to be estimated;
Current set of expectations  $\mathcal{E}_0 = (\tau_0, t_0, C)$ ;
Mode  $m_0$ ;
Compute the set  $\{t_j\}$ ,  $j = 1, \dots, k$  of transitions from  $C$  that may fire in  $[\tau_0, \tau_k]$ 
for  $j = 1, 2, \dots, k$ 
(1) Prediction
    Compute the time window of interest  $W_j$ 
     $P(\Lambda, S) = \text{modelPrediction}(m_{j-1}, W_j)$ 
(2) Likelihood generation
     $P(\mathbf{y}[n]|\Lambda, S) = (2\pi)^{-(l/2)} |R[n]|^{-(1/2)} \exp\left(-\frac{1}{2} \mathbf{r}^T[n] R^{-1}[n] \mathbf{r}[n]\right)$ 
    where  $\mathbf{r}[n] = \mathbf{y}[n] - \hat{\mathbf{y}}[n]$  and  $\hat{\mathbf{y}}(n) = H_n(\Lambda, S) * \hat{\mathbf{z}}(n)$ .
(3) Parameter estimation
     $P(\Lambda, S|\mathbf{y}[n]) \propto P(\mathbf{y}[n]|\Lambda, S)P(\Lambda, S)$ 
     $\{\hat{\Lambda}, \hat{S}\} = \max_{\Lambda, S} P(\Lambda, S|\mathbf{y}_{0:n})$ 
(4) Update
    Update the mode  $m_j$  in the Petri net using the estimated  $(t_j, \tau_j)$ 
    Get updated set of expectations  $\mathcal{E}_j$ 
     $j := j + 1$ 
end
    
```

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 algorithm is suited for a distributed implementation. Assume each node stores a copy of signal component templates  $\hat{z}_j[n]$ . At each step, a few global nodes broadcast the model prediction, and each node locally performs signal decomposition, likelihood function generation, and Bayesian estimation.

## VII. EXPERIMENTAL RESULTS

The diagnosis system presented in 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 pre-recorded data streams to emulate online monitoring. The

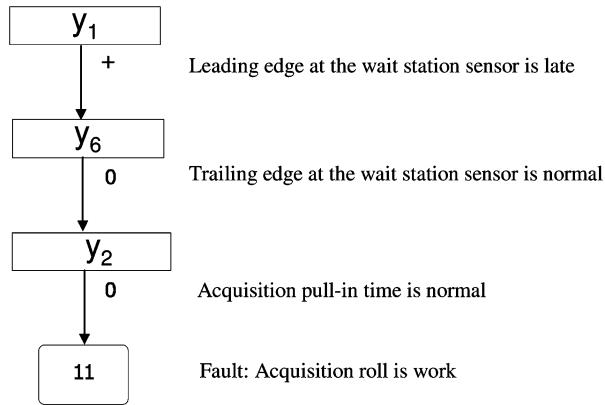


Fig. 9. Trace of the decision-tree diagnoser.

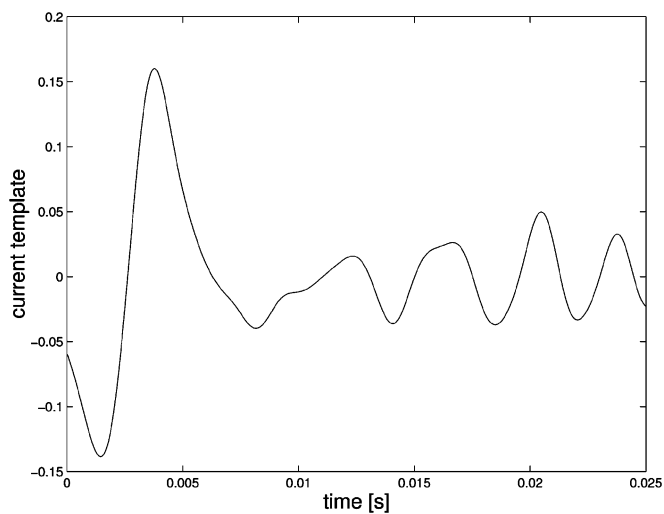


Fig. 10. High-pass filtered current signal template for “drop\_acquisition\_roll” event.

four test cases are: 1) a feed roll worn fault (labeled as “8” in the decision tree of Fig. 7); 2) a feeder motor belt broken fault (label “5”); 3) an acquisition roll worn fault (“label 11”); and 4) a motor slow ramp-up fault (label “2”). They cover an interesting subset of system-level faults of the printer. These faults may cause a delayed paper arrival or no paper arrival 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 estimated by the corresponding virtual sensor test, but with a substantial cost for the signature analysis. Instead, our algorithm uses the less expensive system built-in sensors to monitor and detect faults and only invokes virtual sensor tests on a when-needed basis.

We trace the execution of the diagnoser for one of the fault scenarios. The trace is shown in Fig. 9. The paper arrives late at the wait station sensor. The arrival time is compared with the expected time to generate a qualitative deviation “+”, which triggers the diagnosis. The qualitative value of  $y_1$  rules out faults such as broken belt. Reading off of the decision-tree, the next test for  $y_6$  (trailing edge arrival time) is then invoked and it returns normal (“0”). This rules out feed roll worn and motor slow ramp-up faults since both would cause the trailing edge

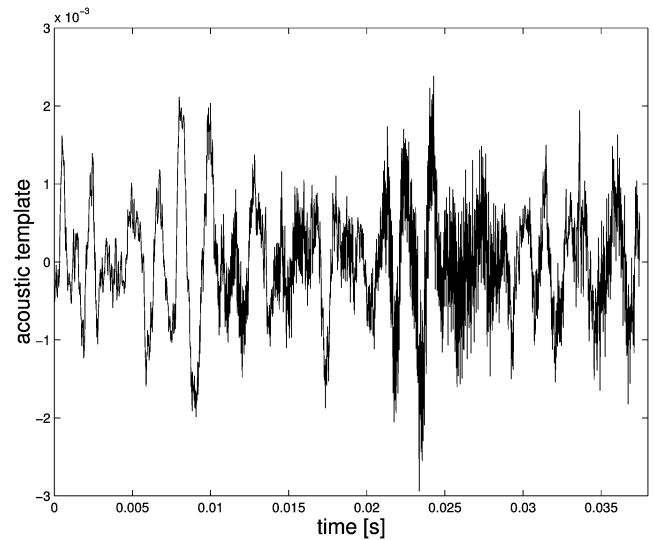


Fig. 11. Acoustic signal template for “motor\_ramp\_up” event.

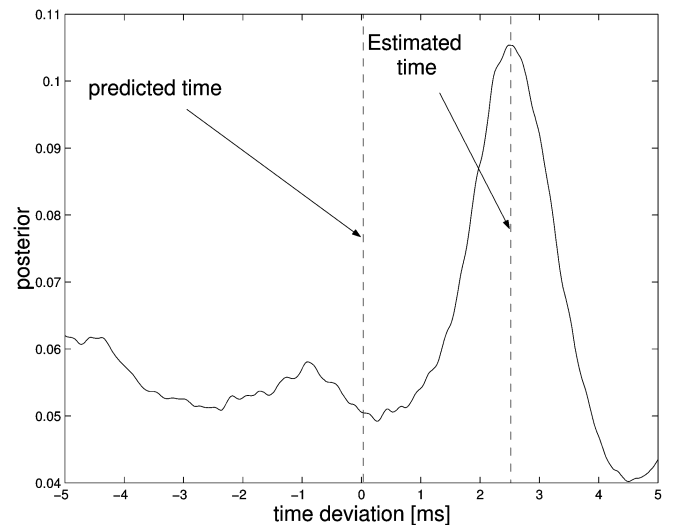


Fig. 12. Posterior distribution of “drop\_acquisition\_roll” firing time.

to be late. Next on the decision-tree, the more expensive acquisition solenoid pull-in time test ( $y_2$ ) is invoked. This calls the mode estimation algorithm to determine the transition time at which the acquisition roll contacts the paper (“drop\_acquisition\_roll”), which is 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. 10) and motor (Fig. 11) to compute a posterior probability distribution of the event. Using the Petri net model prediction [495 ms, 505 ms] 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. 12). Therefore, the test for  $y_6$  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 the wait station sensor. 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 both the leading and trailing edge to be late.

The efficiency of the diagnosis approach depends on the number of tests involving the virtual sensors that are requested by the diagnoser. From the decision-tree shown in Fig. 7, we know that faults 1, 5, and 14 require three sensor tests that invoke the mode estimation algorithm, fault 12 requires two such tests, and the remaining faults only one. The running of these tests depends on the length of the acoustic and current templates as well as the consequence interval of the corresponding autonomous transition. In our experiments, every virtual sensor test was completed in approximately 2–3 s.

### VIII. CONCLUSIONS

This paper presents a novel approach for monitoring and diagnosis of hybrid systems. Model-based and statistical methods are integrated in the diagnostic scheme using the underlying principles of model-driven signature analysis and utility-driven information querying. The method has been applied to diagnosis of faults in a complex electro-mechanical system, the Xerox DC265 printer. A fault modeling framework based on hybrid automata is presented. The model can be used for the automatic generation of a fault symptom table for single faults. The fault symptom table is then compiled to a decision tree that is used as the online diagnoser. An online monitoring approach based on a timed Petri net model that abstracts away the continuous dynamics is also described. The advantage of the model-driven signature analysis is that it significantly reduces the computational cost for the mode estimation and enables the online operation of the diagnostic system. The proposed techniques exploit the interaction between the models and signal processing for computational efficiency, and have been experimentally validated for a diagnostic problem on the Xerox DC265 networked printer test-bed.

### ACKNOWLEDGMENT

The authors would like to thank C. Picardi for her assistance in implementing the decision tree algorithms during an internship at Xerox PARC, S. Narasimhan for his assistance in implementing a Petri net simulator during an internship at Xerox PARC, B. Siegel for helping acquiring the test fixture, M. Sampath, R. Root, and L. Durfey for their help in instrumenting the testbed, J. de Kleer, and B. Siegel for insightful discussions on diagnostics, and J. Kurien for comments on drafts of the paper.

### REFERENCES

- [1] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artif. Intell.*, vol. 32, no. 1, pp. 97–130, 1987.
- [2] W. Hamscher, L. Console, and J. de Kleer, *Readings in Model-Based Diagnosis*. San Mateo, CA: Morgan Kaufmann, 1992.
- [3] C. Goodrich and J. Kurien, "Continuous measurements and quantitative constraints—Challenge problems for discrete modeling techniques," in *Proc. iSAIRAS-2001*, Montreal, QC, Canada, 2001.
- [4] A. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, pp. 600–611, 1976.
- [5] E. Isermann, "Process fault detection based on modeling and estimation methods—A survey," *Automatica*, vol. 20, pp. 387–404, 1984.

- [6] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.
- [7] J. Gertler, "Analytical redundancy methods in fault detection and isolation—A survey and synthesis," in *IFAC/IMACS Safeprocess*, 1991, pp. 9–21.
- [8] R. Patton and J. Chen, Eds., *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Norwell, MA: Kluwer, 1999.
- [9] J. Lunze, "Diagnosis of quantised systems by means of timed discrete-event representations," in *Hybrid Systems: Computation and Control*. ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds. New York: Springer, 2000, vol. 1790, pp. 258–271.
- [10] S. Narasimhan, "Model-Based Diagnosis of Hybrid Systems," Ph.D., Elec. Eng. Comput. Sci. Dept., Vanderbilt Univ., Nashville, TN, 2002.
- [11] P. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 6, pp. 554–565, Nov. 1999.
- [12] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta, "Hybrid systems diagnosis," in *Hybrid Systems: Computation and Control*. ser. Lecture Notes in Computer Science, N. Lynch and B. Krogh, Eds. New York: Springer, 2000, vol. 1790, pp. 282–295.
- [13] D. Koller and U. Lerner, "Sampling in factored dynamic systems," in *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. New York: Springer, 2001, pp. 445–464.
- [14] X. Koutsoukos, J. Kurien, and F. Zhao, "Estimation of hybrid systems using particle filtering methods," in *Proc. MTNS 2002*, Notre Dame, IN, 2002.
- [15] —, "Estimation of distributed hybrid systems using particle filtering methods," in *Hybrid Systems: Computation and Control (HSCS'03)*, ser. Lecture Notes in Computer Science. New York: Springer, 2003, vol. 2623, pp. 298–313.
- [16] X. Koutsoukos, "Estimation of hybrid systems using discrete sensors," in *Proc. 42nd IEEE Conf. Decision and Control*, Dec. 2003, pp. 155–160.
- [17] R. Dearden and D. Clancy, "Particle filter for real-time fault detection of planetary rovers," in *Proc. 13th Int. Workshop on Principles of Diagnosis (DX'2002)*, Semmering, Austria, 2002.
- [18] M. Basseville, A. Benveniste, and L. Tromp, "Diagnosing hybrid dynamical systems: Fault graphs, statistical residuals and viterbi algorithms," in *Proc. 37th IEEE Conf. Decision and Control*, 2000, pp. 3757–3762.
- [19] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proc. 7th Nat. Conf. Artificial Intelligence (AAAI'2000)*, 2000.
- [20] M. Hofbauer and B. Williams, "Mode estimation of probabilistic hybrid systems," in *Hybrid Systems: Computation and Control (HSCC'02)*. ser. Lecture Notes in Computer Science, C. Tomlin and M. Greenstreet, Eds. New York: Springer-Verlag, 2002, vol. 2289, pp. 253–266.
- [21] E. Benazera, L. Trave-Massuyes, and P. Dague, "State tracking of uncertain hybrid concurrent systems," in *Proc. 13th Int. Workshop on Principles of Diagnosis (DX'2002)*, Semmering, Austria, 2002.
- [22] G. Ferrari-Trecate, D. Mignone, and M. Morari, "Moving horizon estimation for hybrid systems," *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1663–1676, Oct. 2002.
- [23] V. Srinivasan and M. Jafari, "Fault detection/monitoring using time Petri nets," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 4, pp. 1155–1162, Jul./Aug. 1993.
- [24] D. Pandalai and L. Holloway, "Template languages for fault monitoring of timed discrete event processes," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 868–882, May 2000.
- [25] 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*, 2000, pp. 494–501.
- [26] E. Hung and F. Zhao, "Diagnostic information processing for sensor-rich distributed systems," in *Proc. 2nd Int. Conf. Information Fusion (Fusion'99)*, Sunnyvale, CA, 1999.
- [27] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P.-H. Ho, X. Nicollin, A. Oliveiro, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoret. Comput. Sci.*, vol. 138, pp. 3–34, 1995.
- [28] L. Console, C. Picardi, and M. Ribando, "Diagnosis and diagnosability analysis using process algebra," in *Proc. 11th Int. Workshop on Principles of Diagnosis (DX'2000)*, Michoacan, Mexico, 2000.
- [29] L. Trave-Massuyes, T. Escobet, and R. Milne, "Model-based diagnosability and sensor placement application to a frame 6 gas turbine sub-system," in *Proc. IJCAI'2001*, Seattle, WA, 2001, pp. 551–556.

- [30] P. Antsaklis and X. Koutsoukos, "Hybrid systems: Review and recent progress," in *Software-Enabled Control*, T. Samad and G. Balas, Eds. New York: IEEE Press, 2003, pp. 272–298.
- [31] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete abstractions of hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 971–984, Jul. 2000.
- [32] J. Quinlan, "Combining instance-based and model-based learning," in *Proc. 10th Int. Conf. Machine Learning*, Amherst, MA, 1993.
- [33] J. Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981.



**Feng Zhao** (M'92) received the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge.

He is a Senior Researcher at Microsoft Research, Redmond, WA, where he manages the Networked Embedded Computing Group. He has taught at Stanford University and Ohio State University. He was a Principal Scientist at Xerox Palo Alto Research Center (PARC), Palo Alto, CA, and directed PARC's sensor network research effort. His current interests are in programming models and robust computing

techniques for networked devices such as wireless sensor networks. He is well known for his work in networked embedded systems, distributed algorithms, and artificial intelligence. He recently co-authored a book, *Wireless Sensor Networks: An Information Processing Approach* (San Mateo, CA: Morgan Kaufmann).

Dr. Zhao is serving as the Editor-In-Chief of *ACM Transactions on Sensor Networks*.



**Xenofon Koutsoukos** (M'00) received the Ph.D. degree in electrical engineering from the University of Notre Dame.

He is an Assistant Professor in the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, and a Senior Research Scientist in the Institute for Software Integrated Systems (ISIS). His research work is in the area of embedded and hybrid systems with emphasis on formal methods, fault tolerance, and adaptive resource management. He was a Member

of Research Staff in the Xerox Palo Alto Research Center (PARC), Palo Alto, CA, during 2000–2002, working in the Embedded Collaborative Computing Area. He has published over 40 refereed journal and conference papers and is co-inventor of two patents.

Dr. Koutsoukos is a recipient of a National Science Foundation Career Award.



**Horst Haussecker** (M'03) received the M.S. and Ph.D. degrees in physics from Heidelberg University, Heidelberg, Germany.

He is a Principal Engineer in Intel's Corporate Technology Group, Santa Clara, CA, and Manager of the Computational Nano-Vision research project at Intel Research. Prior to joining Intel, he was a member of the Research Staff at the Xerox Palo Alto Research Center (PARC), Palo Alto, CA, from 1999–2001. During research visits at the Scripps Institution of Oceanography, University of California

at San Diego, La Jolla, between 1994 and 1997, he developed image sequence processing techniques for quantitative analysis of microscopic transport processes across the air-ocean interface. From 1996 to 1999, he was a Researcher at the Interdisciplinary Center for Scientific Computing, Heidelberg University. His research interests include physics-based computer vision, image sequence analysis, infrared thermography, and application of digital image processing as a quantitative instrument in science and technology. He is co-editor and main contributing author of two textbooks in Computer Vision, has authored or co-authored more than 50 peer-reviewed technical articles, and he has nine patents pending.



**Jim Reich** (M'96) received the M.S. degree in electrical and computer engineering from Carnegie Mellon, Pittsburgh, PA, in 1996 and the B.S. degree in aeronautical/astronautical engineering from Massachusetts Institute of Technology, Cambridge, in 1989.

He is a Researcher and Manager of the Embedded Collaborative Computing Area at the Xerox Palo Alto Research Center (PARC), Palo Alto, CA. His career spans 15 years of research and development, ranging from collaborative signal processing

algorithms to control systems and electromechanical design. Since 1999, his research has focused on distributed sensor systems, including mechanical system diagnosis, acoustic and video tracking, and programming methodologies.



**Patrick Cheung** (M'96) received the B.S. degree in electrical engineering from the University of Wisconsin at Madison, the M.S. degree in electrical engineering and control systems from the University of California at Berkeley, and the Ph.D. degree in MEMS from the Mechanical Engineering Department, University of California at Berkeley, in 1995. His doctoral dissertation was titled "Design, fabrication, position-sensing, and control of electrostatic, surface-micromachined, polysilicon microactuator."

He is currently a Research Scientist in the Embedded Collaborative Computing Area of the Systems and Practices Laboratory, Palo Alto Research Center (PARC), Palo Alto, CA. His research interests include engineering- and document-research disciplines, with a particular interest in collaborative sensing. He holds 13 patents.