Component-Based Modeling and Diagnosis of Process-Control Systems†

Gregory Provan and Yi-Liang Chen‡
Rockwell Science Center
1049 Camino Dos Rios
Thousand Oaks, California 91360
{gmprovan, ylchen}@rsc.rockwell.com

Abstract

This paper describes a component-based modeling approach for diagnosing process-control systems. Our approach allows users to build diagnostic models for complex process-control systems based on a library of component models. Process-control systems are modeled in the form of causal networks and diagnoses are generated using a model-based diagnostic technology. We present our software tools for component-based model building and causal networks diagnostics.

1. Introduction

Process-Control systems are an important class of systems that occur frequently in practice, ranging from chemical manufacturing systems to water systems to oil wells. Diagnosis and prognosis of such systems is typically done using model-free signal-processing techniques [BG97, Wowk91], although some model-based analyses are starting to appear [LA98]. Moreover, traditional approaches are labor intensive and machine specific, and only diagnose the local condition of a machine rather than treating it as part of a system. Typically, a set of expert rules derived from significant past case histories is used to interpret the output of one or more sensors.

We propose a novel approach to the diagnosis and prognosis of such systems, using a system-level model-based approach. Our approach combines signal processing and domain knowledge within a system-level, model-based framework for process control. The system model describes the physics of the system, and makes predictions for the sensors based on the operating mode of the system. This new approach provides the following novel capabilities:

1. Building system models by interconnecting elements from a library of component models;
2. Qualitative simulations of multiple system modes, including both nominal and faulty modes – this provides the ability to diagnose those faults that can be detected using average-value sensor readings, rather than sensor spectral data;
3. Fusion of disparate data and information types within the system-level model – this facilitates more accurate diagnostics and prognostics;
4. Ability to integrate new/updated software components easily, using a modular, open systems technology; and
5. Integrating continuous-valued components with a discrete-valued diagnostic/ prognostic executive.

In this paper we focus on presenting our Component-based Diagnostic Model Builder (CDMB), a tool that allows us to build system-level models from component modules in an efficient manner. We use this compositional, modular approach for process-control systems by using a library of primitive modules, such as a pump, motor, section of pipe, valve, sensor, etc. Given a library of such primitive components, a complex system can be constructed simply by interconnecting the components. Once this system has been constructed, we can simulate the behavior of the system, and by connecting the model to real sensor inputs, we can diagnose the health of the system based on the diagnostic capabilities inherent in our system model. We describe the user interface that we developed to allow process engineers to build models using this component-based approach.

We adopt causal networks as the underlying framework for representing systems [Dar98]. Causal networks provide an excellent platform for qualitative modeling, knowledge fusion, and diagnostics, and allow system models to be composed from primitive model fragments. Because of the rigorous framework in which the causal model is constructed, completeness of the knowledge fusion and diagnostics is guaranteed. For detailed description of causal networks, please refer to [Dar98].

This approach is currently being applied to real process-control systems. In this document we describe a real system on which we are testing this approach, a pump-loop system.

We make several advances over existing approaches. First, we have integrated several modules that are typically available only separately; i.e., analysis at the system level is typically not performed using raw sensor inputs, but instead sensors are typically analyzed one by one [BG97, Eshl98, Wowk91]. Second, our model-based approach has several advantages over traditional rule-based approaches. Third, the component-based approach allows us to model a wide variety of systems, and different configurations of systems, such that the resulting diagnostics are automatically updated once the new system is specified. Finally, the component-based approach frees engineers from coming up directly with diagnostic rules for the entire system, instead, it allows them to focus on specifying the fault behavior of individual components

†Research supported in part by The Office of Naval Research, USA, under contract number N00014-98-3-0012.
‡Corresponding author. Phone: (+1) 805 373-4108.

In the next section, we describe one an application domain of our approach: a pump loop. In Section 3, we present the overall architecture of our component-based modeling and diagnostic system and briefly explain each module in the system. We describe the modeling and diagnostic reasoning of the pump loop using our system in Section 4.

2. Application Domain: A Pump-Loop System

We now describe a particular type of process-control system, which we term a pump loop. Figure 1 shows an example of a pump loop. This pump loop consists of a fluid tank, a pump connected to a motor, and a pipe loop through which the liquid is pumped.

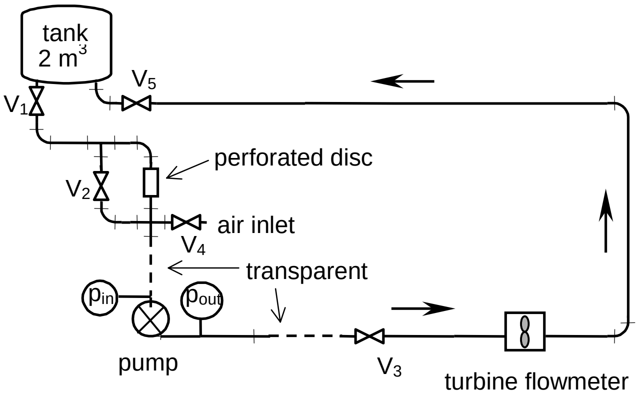

Figure 1: Pump-Loop Configuration

In this pump loop, water flows from the tank to the inlet of the pump and returns to the tank from the outlet of the pump. Several valves are installed along the piping to control the water flow. The water pressure at both the inlet and the outlet of the pump are measured using two pressure sensors. A turbine flowmeter is used to measure the water flow. We are mainly interested in two types of faults that commonly occur in pump loops: pipe blockage and pump cavitation. The pipe blockage can be simulated by the closing of Valve 3 and the pump cavitation can be induced by using different perforated discs to partially block the input flow to the pump.

3. System Architecture

This section describes the architecture for our prototype diagnostic development system, and then summarizes the function of each module in this architecture.

3.1 Overall Architecture

Our current prototype consists of five components, as shown in Figure 2.

- The component database is a repository of the primitive components used for building system-level models.
- The user interface, CDMB, allows a user to create system-level models, and to set values of variables in the model for simulation and diagnostic testing.

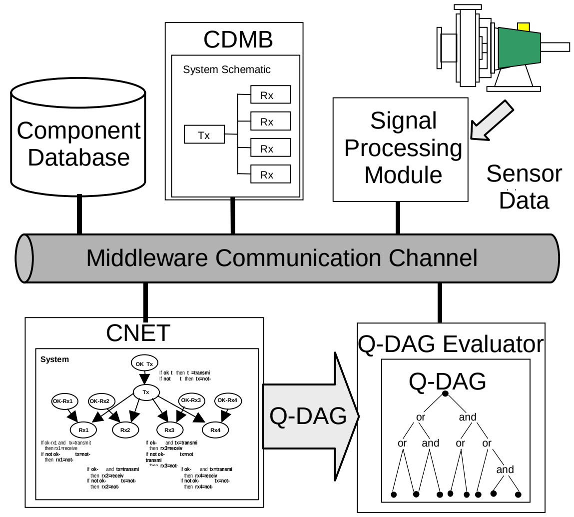

Figure 2: Architecture for Distributed Diagnostic Development System

- The Causal Network Inference System (CNETS) performs all diagnostic inference based on model settings in the user interface CDMB.
- The signal-processing module takes data from real systems and processes then into discrete diagnostic indicators that are used by the diagnostic inference system, CNETS [Dar92].
- If an embedded system is required, CNETS can compile the model into an embedded representation, called a Q-DAG (Query Directed Acyclic Graph) [Dar98], and use the Q-DAG for on-line diagnostic inference.

The distributed environment allows for multiple data collectors and/or presenters, if so desired. Separating these components using a modular, open systems technology provides a system design that allows flexible applications for the diagnostic reasoning methods, the data collection and concentration mechanisms, and the maintenance display platform. Additional benefits of this architecture are improved responsiveness, throughput, upgradeability, and extensibility. This architecture also allows us to simply integrate a continuous-valued signal processing component with a discrete-valued diagnostics/prognostics module.

Figure 3 shows the typical diagnostic processing for a pump-loop. A sensor, such as a turbine flowmeter, transmits data to the Signal Processing Module, labeled as DSP in the figure. A set of discrete diagnostic indicators as specified by the model are then computed based on different analyses (e.g., measurements of amplitude at particular frequencies in the spectra, signal-to-noise ratios, etc.). The inference system takes these indicators and the system model to compute the most likely failed components of the system, if any such failures are indicated.

3.2 Diagnostic System Modules

This section summarizes the major modules in the diagnostic system, including the CDMB interface and the CNETS inference system and compiler.

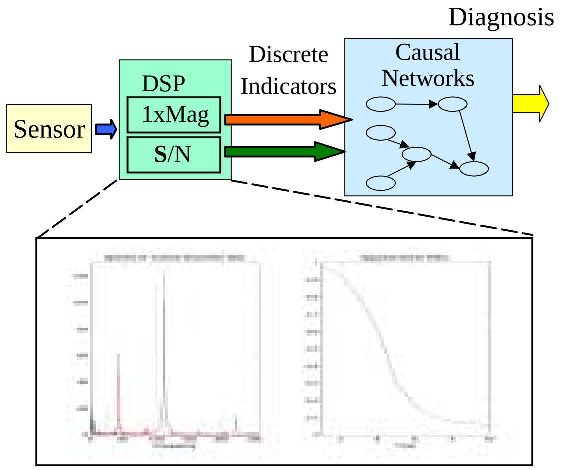

Figure 3: Stages of processing for typical sensor data

3.2.1 Component Database

Our component-based modeling approach has several important characteristics. For example, it allows users to model a wide variety of systems, and different configurations of systems, such that the resulting diagnostics can be automatically updated once the new system is specified. In addition, this approach focuses engineers on specifying fault and normal behavior of individual components rather than coming up directly with diagnostic rules for the entire system.

The component database contains causal network fragments or primitives for a particular application domain. A component diagnostic model (i.e., a primitive) describes the physical behavior (both normal and faulty) and failure characteristics of the modeled system component. Note that only system behaviors relevant to the diagnosis of potential failures are modeled. Depending on the problem specifications, a primitive could be a valve, pipe, sensor or a complex component like a motor-pump subsystem. Figure 4 shows a sample of the typical components for the process-control domain.

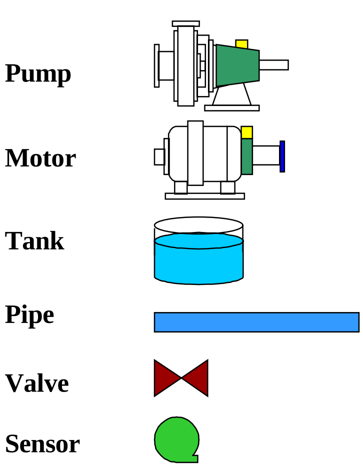

Figure 4: Primitive Components

3.2.2 CDMB User Interface

Diagnostic models developed in our approach are viewed as a collection of interconnected primitives. Once a set of primitives is identified for a diagnostic problem, causal network structures for the primitives can be built and compiled using our CDMB graphical interface. These primitive models are then stored and made available in the CDMB for constructing system models. To develop a system model, users simply place instances of these primitives and interconnecting them to define a complete system. The tool then generates a causal network system model from the specification of these primitives and their interconnections. Figure 5 shows a representation of a system-level model for the pump loop shown in Figure 1 that can be constructed from the primitives in Figure 4.

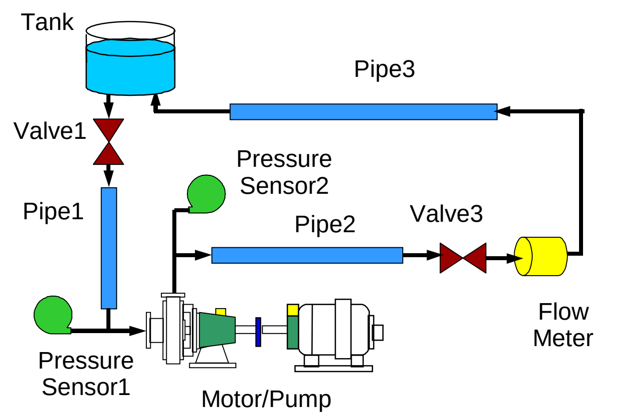

Figure 5: Component-Based Model of a Pump Loop

Diagnostic models developed in CDMB can be debugged and refined through CDMB's simulation utility. CDMB allows the user to simulate normal and abnormal conditions in the system behavior. System users can also simulate and refine the model by changing the state of individual components. In the system, the colors of the blocks indicate the state of the block, e.g., active or inactive. The fully interactive environment provides immediate feedback (e.g., highlighting of affected pathways, changing of color-coded states) to gain understanding of model behavior.

3.2.3 Causal Networks Inference System (CNETS)

CNETS is a system for representing and reasoning with causal networks, which include probabilistic, Order-of-Magnitude Probability (OMP), and symbolic causal networks. CNETS allows the user to create a model by specifying a causal structure and a quantification of this structure using "behavior equations". The behavior equations (which map into causal network properties) defined for each primitive building block in CDMB can be simple or complex. Flexible equations are used to capture this behavior: CNETS supports functional relationships based on both multi-valued propositional logic and conditional probabilities for equations; it uses in addition

OMP or probabilistic weights for modeling component reliabilities, thereby capturing failure models.

A number of inference algorithms in CNETS can be used to process queries about the behavior of a system, such as simulation, prediction and/or diagnostic queries. The fundamental class of algorithms used by CNETS transforms the input causal network into a form that facilitates the particular type of query-processing. The main algorithm used is a modified version of the clique-tree algorithm [JLO90]. CNETS also supports a symbolic form of this algorithm for answering symbolic queries [Dar98] and for Discrete-Event Systems [DP96].

For diagnostic inference, the causal network approach provides a number of guarantees. First, the causal network approach provides soundness and completeness guarantees based on the model [Dar98]. In other words, if any diagnosis is computable from the model, CNETS is guaranteed to compute it. This guarantee allows users, when developing models, to test diagnostic coverage.

Second, it can provide complexity guarantees in terms of the structural parameters of the model (e.g., one of the most significant parameters, the maximum number of parent nodes of any node in the causal structure). Because all significant complexity parameters in this approach are ultimately derivable from the system structure, one can manipulate the structure to obtain acceptable inference complexity without altering the diagnostic coverage. Approaches that are not “structure-based” do not have this capability to alter inference complexity without altering the diagnostic coverage.

Beyond the complexity guarantees, the causal network approach has been optimized in a number of ways. This approach uses a focusing mechanism to focus inference on only the most likely diagnoses [Dar98]; in addition, it makes use of observations to prune and to decompose the system (and reduce overall inference complexity) [DP96].

3.2.4 CNETS Compiler for Run-Time Inference

A run-time version of the system model can be compiled [Dar98] in conjunction with an algorithm like [JLO90]. This allows diagnostic queries to be performed on embedded platforms with limited processing power. Using this novel compilation technique, CNETS is able to compile a causal network into a Boolean expression, called a Q-DAG [Dar98] that contains all possible diagnoses for the system, given inputs for the observable system variables. The Q-DAG provides the same complete diagnostic information that is available with the original causal network, given input variables and diagnostic output variables are clearly specified. This embeddable model together with a very simple evaluator require minimal memory resources and CPU power and can be ported to other platforms and languages with relative ease.

3.2.5 Signal Processing Module

We model a signal processing module using the pair (Θ,Φ), according to the continuous input data Θ (from sensors) and the discrete output data (diagnostic indicators specified by the system model) Φ. For any sensor there is a function *f*: $\Theta \rightarrow \Phi$ that maps the input to the output. For this application, this mapping *f* consists of signal processing algorithms. For example, if the sensor input Θ is time-domain accelerometer data, then *f* will consist of an extraction of amplitude peaks at particular frequencies of the FFT of the data, yielding a discretized classification of these amplitude peaks, e.g., peak at 60Hz is >1.5, peak at 120Hz is >2.0, etc.

4. Modeling and Diagnosis of a Pump Loop

In this section, we describe how we use our CDMB/CNETS system to construct the diagnostic model for a pump loop and how to perform diagnostic reasoning based on the diagnostic model. We start with the modeling of system components and then move on to the modeling and diagnosis of the entire system. Note that we model the pump loop as a *symbolic* causal network.

4.1 Modeling of Pump Loop Elements

We construct the causal network diagnostic models for components in a pump loop (as shown in Figure 4) by using the CDMB interface.

To construct a diagnostic model for a component, we follow a four-step procedure as described below. First, we specify the input and output variables of the component model and the values that they can hold. Second, we specify the internal variables (i.e., unobservables) and potential failure modes (i.e., assumables) and their possible values. We then construct the causal structure of the component using the variables defined as nodes. The causal structure describes the causal relationship among the variables. Finally, we construct a set of the propositional equations for each node/variable (inputs and assumables excluded) that describe how the values of its parents nodes determine the value of this node.

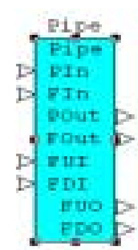

Figure 6: Component Model for a Section of Pipe

Figure 6 shows the top-level view of the component diagnostic model for a section of pipe in CDMB. This model has four inputs (P_{in}, pressure at upstream; F_{in}, flow at upstream; FU_i, flow condition from upstream; and FD_i, flow condition from downstream) and four outputs (P_{out}, pressure at downstream; F_{out}, flow at downstream; FU_o and

FD_O, flow conditions to upstream and downstream).

The internal variables and causal structure of the pipe model created in CDMB are shown in Figure 7. There are five blocks (i.e., internal/output variables) in the figure. The "$Blockage_M$" block (whose possible values include normal, partial-blockage, and blockage) represents the blockage failure mode of the pipe. The other four blocks are associated directly to the four outputs of the model. The links between nodes characterize the causal relationship between nodes. For example, from Figure 7, we know that the value of P_{out} will be determined by the values of P_{in} and $Blockage_M$.

Figure 8 shows a partial set of propositional equations for P_{out} and the interface in CDMB that allows easy entry of these equations. Note that the CDMB interface provides automatic syntax checking for the propositional equations.

Diagnostic models for other components in a pump loop can be constructed in the same manner. These component models can then be stored in a database that can be used in CDMB for constructing system level model.

4.2 Modeling of a Pump Loop

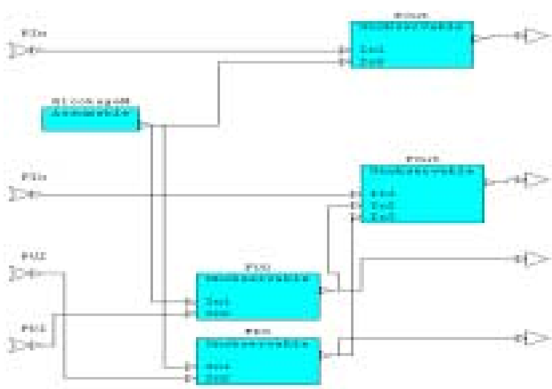

Figure 7: Causal Structure for Pipe Model

With the component diagnostic models built and stored in a database, the construction of the pump loop model in CDMB is straightforward. We first retrieve component models needed for the pump loop and then interconnect them appropriately. The CDMB provides several checking functions that could help to ensure the consistency and correctness of the model.

Figure 9 shows the complete diagnostic models for the pump loop shown in Figures 1 and 5 using the component models developed in CDMB. This model consists of a tank, a pump, two valves, three sections of pipe, a flowmeter, and two pressure sensors. Note that we use two different models for pressure sensors. The first pressure sensor (PresSensor1) reports only the "static" information about the pressure it measures. However, the data collected at the other pressure sensor (DSPPresSensor) are going through

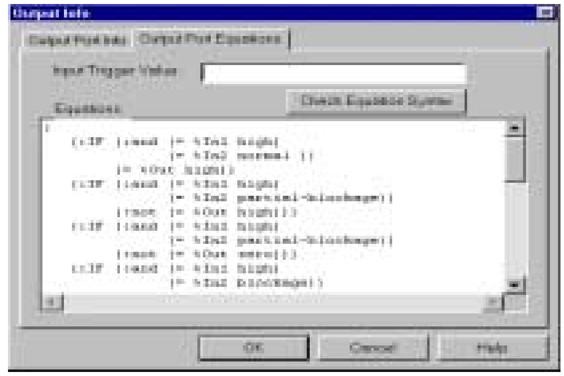

Figure 8: Entering Propositional Equations for P_{out}

further analyses that include the measurement of signal amplitudes at certain frequencies and the Signal-to-Noise ratio. The CDMB/CNETS then use this model to generate textual diagnostic code for the complete system.

4.3 Example: Diagnostic Reasoning

We now use our model to show two of the important features of our approach as mentioned in Section 1. They are, first, qualitative simulations of multiple system modes including both nominal and faulty behaviors and, second, fusion of disparate data and information types within the system-level model for more accurate diagnostics.

Using the qualitative simulation capability of our CDMB/CNETS system, we can assign values to different variables and generate possible diagnoses under these different settings. There is no need to set the values of all variables. Variables whose values are not set will be treated as unavailable or unknown and CNETS will generate possible diagnoses based only on the information available. For example, if we set the flowmeter reading to low in the pump loop and leave readings of other sensors to be unknown, then our system will return four possible diagnoses indicating there could be partial blockage in one of the pipe sections or cavitation in the pump. Figure 10 shows one of the four diagnoses indicated by CDMB. (The diagnostic message appears at the low left corner of the figure.)

The multiple diagnoses of the previous example were due to the lack of information since only the flowmeter reading was available. Those four diagnoses were the best inference that can be made with the data. If we set the three outputs of DSPPresSensor, (P_{out}, SN, Mag), to be (high, nominal, nominal) and keep the flowmeter reading to low, then our system returns with a unique diagnosis of Pipe 1 being partially blocked. This demonstrates the ease of fusing different data to produce more accurate diagnosis in our approach. Note that, if we unset the flowmeter reading and use only the pressure sensor data, our system would return with no failure identified (i.e., normal) which means that the pressure sensor data alone in this case do not indicate there is any abnormality in the system behavior.

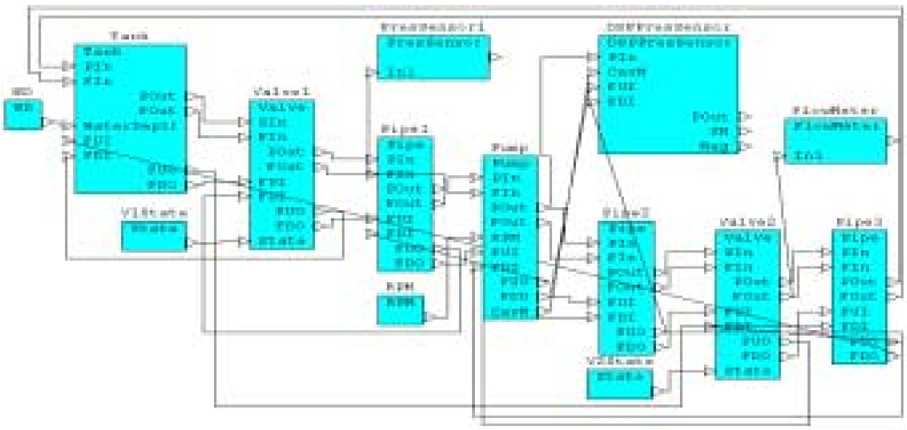

Figure 9: Diagnostic Model for a Pump Loop

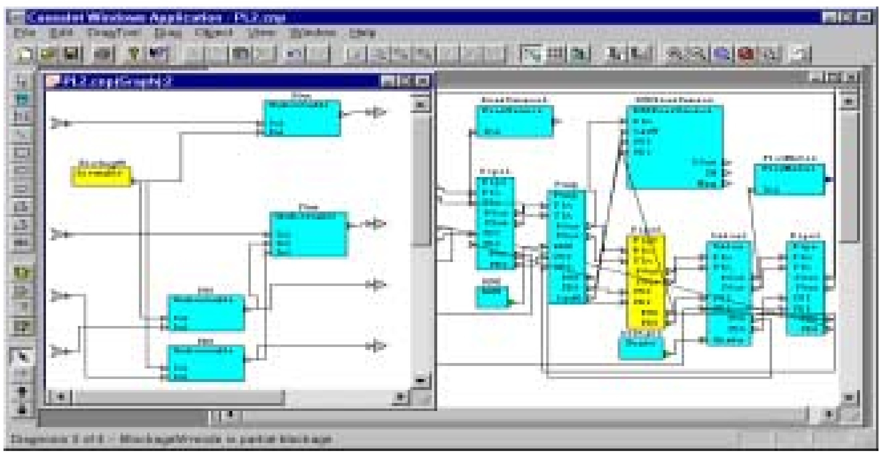

Figure 10: One of the four diagnoses – partial blockage of Pipe 2

5. Conclusions

This paper has described an implemented system for component-based modeling and diagnosis of process-control systems. This approach allows engineers to build models of complex systems using a set of pre-defined primitives, thereby isolating the engineers from the underlying diagnostics models. The underlying diagnostics models are specified using causal networks. Given a system-level model, our approach can take input from several sensors and provide a diagnostic classification of the system as output.

References

[BG97] H. P. Bloch and F. K. Geitner. *Machinery Failure Analysis and Troubleshooting*, Vol. II, 3rd ed., Gulf Publishing Co., 1997.

[Dar98] A. Darwiche. Model-based diagnosis using structured system descriptions. *J. of AI Research*, 8:165-222, June 1998.

[Dar92] A. Darwiche. *A Symbolic Generalization of Probability Theory*. Ph.D. Thesis, Department of Computer Science, Stanford University, 1992.

[DP96] A. Darwiche and G. Provan, 1996. Exploiting system structure in model-based diagnosis of discrete-event systems. *Proc. of Intl. Workshop on Principles of Diagnosis*, pp. 95-105.

[JLO90] V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in recursive graphical models by local computation. *Computational Statistics Quarterly*, 4:269--282, 1990.

[LA98] K. Loparo and M. Adams. Development of machinery monitoring and diagnostic methods based on nonlinear vibration characteristics, In *Proc*. of 52nd MFPT, pp. 239-249, 1998.

[Wowk91] V. Wowk. *Machinery Vibration: Measurement and Analysis*. McGraw-Hill, 1991.