DISTRIBUTED DIAGNOSIS OF CONTINUOUS SYSTEMS: GLOBAL DIAGNOSIS THROUGH LOCAL ANALYSIS

By

Indranil Roychoudhury

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

August, 2009

Nashville, Tennessee

Approved:

Professor Gautam Biswas

Professor Xenofon Koutsoukos

Professor Gabor Karsai

Professor Sankaran Mahadevan

Professor Nilanjan Sarkar

To *Ma*, *Baba*, and Minu *didi*.

PREFACE

Early detection and isolation of faults is crucial for ensuring system safety and efficiency. Online diagnosis schemes are usually integrated with fault adaptive control schemes to mitigate these fault effects, and avoid catastrophic consequences. These diagnosis approaches must be robust to uncertainties, such as sensor and process noise, to be effective in real world applications. Also, diagnosis schemes must address the drawbacks of centralized diagnosis schemes, such as large memory and computational requirements, single points of failure, and poor scalability. Finally, to be effective, fault diagnosis schemes must be capable of diagnosing different fault types, such as incipient (slow) and abrupt (fast) faults in system parameters.

This dissertation addresses the above problems by developing: (*i*) a unified qualitative diagnosis framework for incipient and abrupt faults in system parameters; (*ii*) a distributed, qualitative diagnosis approach, where each diagnoser generates globally correct diagnosis results without a centralized coordinator and communicates minimal measurement information and no partial diagnosis results with other diagnosers; (*iii*) a centralized Bayesian diagnosis scheme that combines our qualitative diagnosis approach with a Dynamic Bayesian network (DBN)-based diagnosis scheme; and (*iv*) a distributed DBN-based diagnosis scheme, where the global DBN is systematically factored into structurally observable independent DBN factors that are decoupled across time, so that the random variables in each DBN factor are conditionally independent of those in all other factors, given a subset of communicated measurements that are converted into system inputs. This allows the implementation of the combined qualitative and DBN-based diagnosis scheme on each DBN factor, which operate independently with a minimal number of shared measurements to generate globally correct diagnosis results locally without a centralized coordinator, and without communicating any partial diagnosis results with other diagnosers. The correctness and effectiveness of these diagnosis approaches is demonstrated by applying the qualitative diagnosis approaches to the Advanced Water Recovery System developed at NASA Johnson Space Center; and the DBN-based diagnosis schemes to a complex, twelfth-order electrical system.

ACKNOWLEDGMENTS

I am indebted to my advisors, Professors Gautam Biswas and Xenofon Koutsoukos, for their invaluable comments, critiques, and guidance that ensured the completion of my research and this dissertation. I would also like to thank the remaining members of the committee, Professors Gabor Karsai, Sankaran Mahadevan, and Nilanjan Sarkar, for their insight on the research presented in this dissertation.

I owe thanks to Matthew Daigle, Nagabhushan Mahadevan, Manish Kushwaha, Joe Porter, Ashraf Tantawy, Chetan Kulkarni, Jian Wu and Eric Manders for the numerous fruitful discussions and brain-storming sessions that helped improve the quality of this work. I would also take this opportunity to express my gratitude to other past and present members of our research group, and friends outside this group, who made my stay at Vanderbilt enjoyable and fun. Of course, this dissertation would have never materialized without the undying support of my family, and I dedicate this work to them.

The two summers I have spent at NASA Ames Research Center working on the Advanced Diagnostics and Prognostics Testbed (ADAPT) have been very rewarding, and I would like thank the members of the ADAPT Team, Ann Patterson-Hine, Scott Poll, Adam Sweet, Joe Camisa, David Nishikawa, David Hall, Serge Yentus, Charles Lee, Christian Neukom, John Ossenfort, and David Garcia for the same.

This work was supported in part by the National Science Foundation under Grants CNS-0347440, CNS-0452067, CNS-0615214, and NSF ITR grant CCR-0225610, and National Aeronautics and Space Administration under grants NRA NNX07AD12A, NASA USRA 08020-013, NASA SGER 0208799, and NASA-ALS NCC 9-159.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Motivation

Modern day engineered systems are a product of careful design and manufacturing, and undergo rigorous testing and validation before deployment. This reduces the likelihood of system failures, but degradation and faults in system components still occur due to wear and tear from sustained operations. Unlikely events and unanticipated situations can also create faults. Early detection and isolation of faults is the key to maintaining system performance, ensuring system safety, and increasing system life. Traditionally, the *fault diagnosis* task has been performed offline during maintenance operations, using test results and alarm signals to isolate faults in system components. For present-day, safety-and-mission critical systems, it is imperative to monitor system behavior and performance online, i.e., during operation, so that system control and operation can adapt to changes and avoid catastrophic failures.

The process of fault diagnosis consists of *detection*, *isolation*, and *identification* of faults [2]. Fault detection typically produces a binary decision that determines whether the observed system behavior has deviated from the expected nominal behavior. Fault isolation involves determination of the cause of the fault, and is sometimes called root cause analysis. Fault identification is the task of determining the extent or magnitude of the fault. We consider a *fault* to be a change in the system which causes the system's behavior to deviate from the expected nominal behavior. Faults manifest at different locations, e.g., in the sensors, actuators, or plants. These faults may manifest at very fast rates (called abrupt faults) or they may be gradual (called incipient faults). In some cases, they may cause unwanted changes in the system structure. Hence, fault diagnosis schemes must be generally applicable and apply to different kinds of faults.

Fault diagnosis approaches can be broadly classified as *model-free* and *model-based* methods [2]. Model-based approaches are considered to be more general than model-free approaches that are mostly based on expert knowledge and data-driven methods. Model-based approaches posses provable properties, such as detectability and isolability of sets of faults. Typically there is a clear separation between the particular model and the reasoning algorithm used in model-based diagnosis approaches. Again, in contrast to model-free approaches, this contributes to more general and scalable solutions. In model-free approaches, the diagnosis algorithm parameters are functions of

the system operating ranges and modes of operation, and any change in the system components or operating ranges usually require the recalibration of the parameters and thresholds used in the detection and isolation scheme. The generation of appropriate system models to support diagnosis, however, can be a challenge, since, real-world systems encompass multiple domains, such as the hydraulic, electrical, and mechanical domains. Their behaviors can contain sharp nonlinearities, and it may be difficult to capture all possible interactions between components of the system, and between the system and the environment. Moreover, the system models have to be built at the correct level of abstraction, balancing the details needed in the model to make the system diagnosable, while keeping the model complexity low, so as to not affect the performance of online diagnosis.

In the real world, uncertainties caused by measurement and process noise, and modeling abstractions and errors are unavoidable. Therefore, effective fault diagnosis schemes must be robust to these uncertainties, and generate correct diagnosis of faults in their presence. Probabilistic reasoning techniques are well suited for this purpose and are based on an intuitive and theoretically sound mathematical foundation which generates consistent diagnostic results under uncertainties, and usually require the generation of probabilistic system models, such as Dynamic Bayesian Networks (DBNs), to capture the uncertainties in the systems to be diagnosed [3]. Once generated, standard Bayesian inference approaches are applied on these probabilistic models to diagnose faults correctly in the presence of uncertainties. However, exact computation of probabilities for systems, barring a few restricted cases, is computationally exponential, and hence, approximate methods for computing these probabilities have to be applied. But, these approximate Bayesian inference schemes can be computationally very expensive for large systems, and may suffer from convergence issues [3, 4].

High costs, such as memory and computational requirements, plague most centralized model-based diagnosis schemes (probabilistic, or otherwise), since these schemes involve one monolithic diagnoser that operates on a global system model and requires all available system measurements for diagnosis [2,5]. The computational expense can be reduced by distributing the diagnosis task into subtasks that can be executed on separate processors. Therefore, distributed diagnosis approaches fit well with present day embedded systems architectures, where each subsystem has associated local processors, memory, and sensors for monitoring and control of that subsystem, e.g., electronic control units in aircrafts. In addition to improving computational efficiency, distributed diagnosis schemes also reduce the high *costs* of shielding and protection of the cables usually incurred to transmit measurements to a centralized computer while maintaining signal quality, especially in harsh environment. Furthermore, distribution of the diagnosis task addresses other issues of

centralized diagnosis schemes, such as *single points of failure*, and *poor scalability*. In centralized diagnosis schemes, a glitch or failure in the diagnosis unit may disable the entire diagnosis system. Distributed diagnosis approaches do not have any such single point of failure. Further, centralized diagnosis schemes scale poorly, since changes in the system configuration and components may cause significant changes in the system's dynamic behavior, requiring the diagnoser to be redesigned. Again this drawback can be addressed by distributing the diagnosis task.

Research Contributions

To address the aforementioned challenges in model-based diagnosis of real-world systems, this dissertation develops a distributed, probabilistic, model-based approach for the accurate diagnosis of *incipient* and *abrupt* faults in a unified framework, in the presence of uncertainties, such as sensor noise and process disturbances. By distributing the diagnosis task into smaller subtasks, we improve the computational efficiency of our diagnosis approach. We further improve the computational efficiency and scalability of our diagnosis approach by combining a qualitative diagnosis approach with a quantitative Bayesian state estimation scheme.

Incipient and abrupt faults are classifications of *parametric* faults, which are characterized by unwanted changes in system parameters. Incipient faults are modeled as slow drifts in system parameter values caused by wear and tear in system components, such as degradation in the stator windings or bearings in induction motors [6] and gradual blockage in pipes in hydraulic or chemical systems due to the accumulation of sediments. Abrupt faults represent faults that are caused by unwanted changes, where the rate of change in system parameter is much faster than the dynamics of the system, or rate of sampling of system measurements. Hence, abrupt faults are modeled as step changes in the parameter values. Examples of abrupt faults include a sudden (partial or complete) blockage in a pipe carrying fluid, or a bias that develops in a sensor. Since both abrupt and incipient faults are common in real-world engineering systems, our comprehensive diagnosis scheme applies to both these types of faults in a unified framework.

The specific research contributions of this thesis are listed below.

1. **Qualitative diagnosis of incipient and abrupt faults in a unified framework:** We extend the TRANSCEND qualitative diagnosis scheme [7, 8] to allow for qualitative diagnosis of both incipient and abrupt faults in a unified framework. The qualitative TRANSCEND diagnosis approach was originally designed for the diagnosis of abrupt faults. The isolation of abrupt faults in TRANSCEND is based on the analysis of fault transients, where observed deviations

in measurements from nominal behavior are matched against qualitative predictions of faulty behavior, known as fault signatures, to isolate faults. We extend the diagnosis scheme to include the detection and isolation of incipient faults.

2. **Distributed qualitative diagnosis of incipient and abrupt faults:** We develop a distributed TRANSCEND-based qualitative diagnosis scheme for continuous dynamic systems. Most of the previous work in distributed diagnosis has been developed for discrete event system models [9,10], but these methods do not scale up for complex continuous systems [11]. Our distributed diagnosis approach designs a multiple diagnoser solution that generates globally correct diagnosis results by local analysis without a centralized coordinator, with no exchange of partial diagnosis results amongst the diagnosers, and with minimal sharing of measurements. We propose two approaches for designing the distributed qualitative diagnosers. The first algorithm assumes the subsystem structure is known and constructs a local diagnoser for each subsystem. The second algorithm creates a partition structure and local diagnosers simultaneously. The absence of a centralized coordinator ensures that our distributed diagnosis scheme does not have a single point of failure. Moreover, because a distributed diagnoser does not depend on the partial diagnosis results of other diagnosers for its own diagnosis, the failure of individual diagnosers do not affect the performance of the other diagnosers. Hence, our distributed diagnosis scheme degrades gracefully as one or more distributed diagnosers fail. Also, in our distributed diagnosis scheme, the diagnosis task is distributed amongst the different distributed diagnosers, and hence, this distributed scheme is computationally less expensive than its centralized counterpart.

3. **Efficient Bayesian diagnosis of incipient and abrupt faults:** We combine the TRANSCEND qualitative fault isolation with a DBN-based state and parameter estimation scheme to develop an efficient probabilistic approach for diagnosis of *both* incipient and abrupt faults in continuous dynamic systems using DBNs to explicitly model the system dynamics and uncertainties. To accommodate nonlinearities, and non-Gaussian distributions, we employ a particle filtering-based state estimation scheme for diagnosis [12]. We use particle filters to ensure that our Bayesian diagnosis scheme is generally applicable to complex nonlinear systems, with non-Gaussian probability distributions. However, particle filtering-based fault diagnosis schemes suffer from the *sample impoverishment* problem [4,13]. We develop a solution to this problem, and describe three different fault identification approaches to estimate the value of the faulty parameter based on the observed measurements, and isolate the true fault. The use of

Bayesian estimation ensures robustness to measurement and process noise, and provides more precise diagnosis results than our qualitative TRANSCEND-based diagnosis scheme. However, the Bayesian state estimation scheme can be computationally expensive for large systems. We improve the efficiency of this Bayesian diagnosis scheme by integrating it with our extended qualitative TRANSCEND diagnosis scheme. The efficiency gain is obtained by first using the qualitative diagnosis scheme to refine the possible fault hypotheses to a tractable number, and then invoking multiple DBN-based filtering schemes for the reduced fault hypothesis set.

4. **Distributed Bayesian diagnosis of incipient and abrupt faults:** We develop a distributed combined qualitative and Bayesian diagnosis scheme that further improves the efficiency of our centralized diagnosis scheme. The basis of our distributed Bayesian diagnosis scheme is the factoring of the system DBN model into multiple, non-overlapping DBN factors, such that each random variable in a DBN factor is conditionally independent of random variables in all other DBN factors given the measurements communicated between these factors. Our DBN factoring scheme is based on computing some state variables in the system as algebraic functions of measurements (considered to be system inputs), which allows the replacement of the across-time links between these state variables with new intra-time links from the measurements to the state variables. If sufficient number of across-time links are removed, we can factor the system DBN into DBN factors such that the random variables in each generated factor is conditionally independent from those in any other factor, given the measurements that were used to compute some state variables. It is well-known that the state variables of a system can be estimated from the system measurements only if the system is *observable*. We ensure that each factor is observable based on the analysis of *structural observability* properties of the system's bond graph model and its component parts [14, 15]. We analyze the structural observability properties of the system to ensure that each DBN factor represents a structurally observable subsystem, and together all the DBN factors retain the structural observability properties of the global system. Once the global DBN is factored, the conditional independence of the random variables in each DBN factor allow the implementation of Bayesian estimation schemes on each DBN factor independently. For our distributed Bayesian diagnosis scheme, we apply our combined qualitative-quantitative diagnosis scheme on each DBN factor instead of the global system DBN. Previous work in factored estimation schemes, such as the Boyen-Koller algorithm, presented in [16], creates the individual factors by eliminating causal links between weakly interacting subsystems. Therefore, the belief state

derived from the individual factors is an approximation of the true belief state. The error in this approximation is bounded, but these bounds may not be sufficiently precise for online monitoring of mission-critical systems. The novelty of our factoring scheme lies in the fact that the DBN factors together preserve the overall system dynamics in the factored form, and there is *no* approximation involved in the factored belief state.

5. **Experimental studies.** We apply our distributed qualitative diagnosis approaches to a complex, real-world system, the Advanced Water Recovery System, which was designed and built at the NASA Johnson Space Center to convert wastewater into potable water for long duration manned missions [17]. Our centralized and distributed Bayesian diagnosis schemes are applied to a complex, twelfth-order electrical circuit, with highly oscillatory behavior. The use of Bayesian diagnosis schemes result in correct and precise diagnosis results in the presence of noisy sensors, while our distributed diagnosis schemes address the drawbacks of centralized diagnosis approaches, especially by improving the computational efficiency when compared to the centralized schemes.

Our distributed approach assumes faults are persistent, abrupt or incipient, and parametric. We assume that the faults are non-catastrophic, i.e., the system still can operated, albeit in a degraded state, after fault occurrence. We make the single fault assumption since simultaneous multiple fault occurrences are much less likely.

Organization of Dissertation

This dissertation is organized as follows. Chapter II presents related work in model-based diagnosis of continuous systems. We start with a taxonomy of faults and fault diagnosis approaches. Then we present and compare different model-based diagnosis schemes for dynamic systems, and how these different schemes handle uncertainties. Specifically, we describe different diagnosis approaches, such as discrete-event systems approaches, qualitative diagnosis schemes, analytical redundancy relations-based diagnosis approaches, and probabilistic model-based diagnosis schemes. We conclude this chapter by presenting our diagnosis approaches in context of the related work.

Chapter III starts with the necessary background on the TRANSCEND qualitative diagnosis scheme, and extends it to the combined diagnosis of incipient and abrupt faults. We begin by presenting the modeling paradigms used in TRANSCEND, i.e., bond graphs [18] and temporal causal graphs (TCGs) [7], and then describe, in detail, the different steps of the TRANSCEND diagnosis scheme. Then, we present our extensions to TRANSCEND for the diagnosis of incipient faults.

Specifically, we describe how we have extended the TRANSCEND fault signature generation scheme to generate fault signatures for incipient faults. This allows for seamless integration of incipient and abrupt fault diagnosis in TRANSCEND. Finally, we present an analysis of the diagnosability properties of the TRANSCEND qualitative diagnosis scheme for the unified qualitative diagnosis of incipient and abrupt faults.

Chapter IV describes our distributed qualitative scheme for diagnosis abrupt and incipient faults. First, we present our distributed diagnosis architecture, where each distributed diagnosis is essentially a TRANSCEND diagnoser that uses a subset of observations to diagnose a subset of faults. Through the careful design of these distributed diagnosers, we guarantee that each distributed diagnoser will generate globally correct diagnosis results through local analysis, without a centralized coordinator, and no exchange of partial diagnosis results, but through the communication of minimal number of measurement information. Two approaches for designing the distributed qualitative diagnosers are presented. In the first diagnoser design approach, we assume knowledge of subsystem structure, especially the measurements and faults that belong to each subsystem, and based on this information, we design a local diagnoser for each subsystem such that it required minimal number of additional external measurements to globally diagnose all the faults assigned to that subsystem. In the second approach, we assume no prior partitioning information. Instead, we generate the maximal number of distributed diagnosers, such that, each local diagnoser can operate independently *without* sharing *any* measurements to generate globally correct diagnosis results. The formulation of the diagnoser design problems and the algorithms for designing these distributed diagnosers are described in the next two sections. This is followed by a set of studies that demonstrate the usefulness of this distributed diagnosis approach. We then present a case study for a real-world engineering system. We verify the correctness and efficacy of the different diagnosis approaches and the DBN factoring scheme by applying them to the Advanced Water Recovery System, developed at the NASA Johnson Space Center, a real-world large engineering system.

Chapter V presents our centralized Bayesian scheme for diagnosing abrupt and incipient faults in continuous systems, where we combine the qualitative TRANSCEND fault isolation scheme with a DBN-based quantitative fault hypothesis refinement and identification approach. First, we present the computational architecture of our diagnosis approach, and then describe the procedure for systematically deriving the DBNs for nominal and faulty system behavior from the system bond graph. The following section presents our centralized diagnosis approach, which includes detection, isolation, and identification of the fault hypotheses using a particle filtering-based state estimation scheme. We then present three schemes for accurate estimation of faults using the particle filter approach.

Use of this approach requires addressing the sample impoverishment problem, as discussed in the following section. This is followed by a set of experimental results that demonstrate the efficacy of our diagnosis scheme. We conclude this chapter with a discussion of the contributions in this work.

Chapter VI presents our distributed Bayesian approach for diagnosing incipient and abrupt faults. We start by presenting our distributed diagnosis architecture. Then we formulate the diagnoser design problem. The main idea is to factor the system's global DBN into conditionally independent DBN factor, such that each DBN factor is structurally observable, and apply the combined qualitative-Bayesian diagnosis scheme on each DBN factor independently. We present our diagnoser design approach based on factoring the system DBN into structurally observable DBN factors in the following section. The next section provides proof that the design of our distributed diagnosers ensures that our distributed diagnosis properties of generating globally correct diagnosis through local analysis is satisfied. We then present some experimental results to demonstrate the effectiveness of our factored estimation and distributed diagnosis scheme.

Chapter VII summarizes the contributions of this dissertation, and presents some conclusions. We also describe the current limitations of our approaches, and identify future directions of work to improve the current research.

CHAPTER II

RELATED WORK IN MODEL-BASED DIAGNOSIS OF DYNAMIC SYSTEMS

Timely detection and isolation of faults is very important for efficient and safe performance of engineering systems. For safety critical systems, such as aircraft, a fault in its component, if it goes undetected, may have serious consequences in terms of the system's operation, and cause loss of life and property. A "potentially harmless" fault in a computer network can hinder the productivity of office staff, and result in monetary losses and unexpected delays. Given the varied nature of faults, and the adverse effects they can have on system operation, the task of accurate and timely diagnosis of system faults in complex dynamic systems presents a number of important and interesting research challenges. In this chapter, we provide a taxonomy of faults and present several classical model-based diagnosis approaches that have been developed for the detection and isolation of faults in dynamic systems.

A Taxonomy of Faults

Faults are undesired changes that cause deviations in expected system behavior, which then affect system performance [2, 7]. In this research, we differentiate faults from complete failures of the system. We assume that faults cause degradation in system performance, but may not result in complete loss of system functionality. As an example, a short-circuit in a battery that causes the battery to explode is considered a failure, and hence, beyond the scope of this dissertation. However, the gradual or an abrupt decrease in the battery's charge storage capacity is considered a fault, since the battery can still operate in a degraded manner. We adopt the terminology used in the domain of fault detection and isolation [2, 19, 20] to present the different concepts in the remainder of this chapter.

Definition 1 (Fault)**.** A *fault* is an unexpected change in the plant or its instrumentation that causes the system to deviate from its nominal behavior.

Faults can be classified based on how they are modeled, their temporal profile, and their location, as shown below.

1. **Fault Model:** Based on how they are modeled, faults can be classified as *additive*, *parametric*, or *discrete*. Additive faults are modeled as unknown inputs to the system, which are nominally

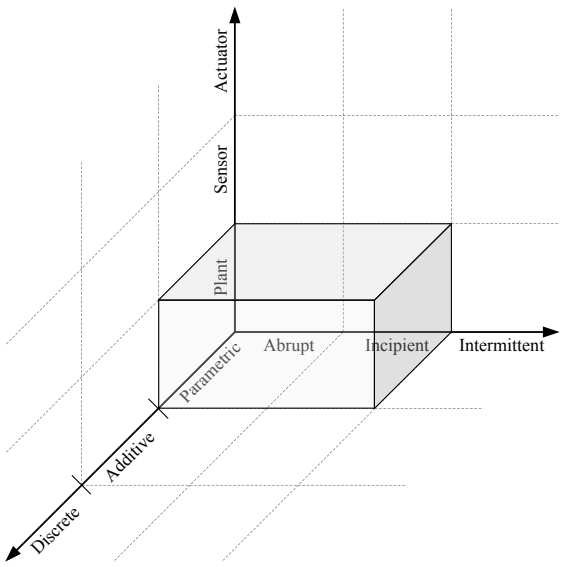


Figure 1: Scope of this dissertation with respect to the types of faults we diagnose.

zero. Additive fault effects are decoupled from the system dynamics, therefore, they can be studied by analyzing changes in the system input-output relations without changing the basic system dynamics model. Examples of additive faults include sensor faults (if the sensor measurements are not part of the control loop), small leaks in a system, and changes to the plant loads. A discrete fault causes a change in the system structure or topology. Examples of discrete faults include broken wires and unexpected changes in a system's configuration. *Parametric* faults result in changes to the system parameters, and hence, these faults directly impact the system dynamics. Therefore, fault effects cannot be analyzed by decoupling them from the nominal system dynamics. In other words, parametric faults directly affect the system's dynamic behavior and one has to analyze the nominal system and fault dynamics simultaneously to isolate these faults. Examples of parametric faults include changes in the parameter values of physical processes in the system, such as, the energy storage elements or the dissipative elements.

2. **Temporal profile:** The temporal profile of a fault is linked to the persistence of a fault's effects. *Persistent* faults do not "disappear" once they have occurred. On the other hand, *intermittent* faults manifest for some time, and then their effects cease to perturb the system dynamics. Intermittent faults typically appear and disappear at random intervals. Persistent faults can be further categorized as *abrupt* or *incipient*. *Abrupt faults* cause changes in parameter values that occur at a rate much faster than the nominal system dynamics. Abrupt

parametric faults are usually modeled as step-changes in component parameter values. In contrast, incipient faults develop slowly over time, and the change in fault parameter value is defined by a slow temporal function.

3. **Fault location:** Based on where faults are located in the system, they can be *sensor*, *actuator* and *plant* faults. While sensor faults are attributed to measurement devices, and characterized by discrepancies between the actual values of plant variables and the values reported by the instrumentation system, actuator faults are located at system inputs, and plant faults are characterized by faults in the system parameters.

In summary, a fault is completely defined by its fault model, temporal profile, and location. For example, a sensor bias fault is characterized as an additive, (persistent) abrupt, sensor fault. Similarly, some of the common failures in electric induction motors are defined as parametric, (persistent) incipient, plant faults.

As shown in Fig. 1, this dissertation research aims at diagnosing abrupt and incipient parametric plant faults in continuous dynamic systems. Faults in sensors and actuators, as well as discrete faults are beyond the scope of this dissertation.

A Taxonomy of Model-Based Fault Diagnosis Approaches

The task of fault *diagnosis* includes fault *detection*, fault *isolation*, and fault *identification*, as described below:

1. **Fault Detection:** Fault detection comprises of methods that produce binary decisions as to whether the deviation in system behavior from nominal is attributed to a fault in the system or not.

2. **Fault Isolation:** Fault isolation refers to schemes that determine the component or subsystem malfunction that explains the observed discrepancies in system behavior.

3. **Fault Identification:** Fault identification is the task of determining the magnitude or extent of the fault. For parametric faults, fault identification involves estimating the amount of deviation in an abrupt fault, or the rate at which an incipient fault parameter changes over time.

A number of fault diagnosis approaches have been developed by researchers and practitioners [7, 21–25]. The primary form of prior knowledge required for diagnosis is the set of faults, and the

relationship between the observed symptoms and the faults. This prior domain knowledge may be derived from first principles, domain-theoretic understanding of the system, such as physical system models derived using physical laws. Model-based diagnosis schemes base their reasoning on such knowledge. In contrast, model-free diagnosis approaches use an implicit or associational knowledge of the system behavior, based on past experience of faulty and nominal system behavior. As mentioned in the previous chapter, our focus is on model-based diagnosis approaches for nonlinear dynamic systems [7,23,24,26]. In dynamic systems, the system dynamics vary with time and the system has state, i.e., its behavior depends on both present and past inputs. In non-linear systems, the system behavior is best represented as a non-linear function of the system parameters, control inputs, and other system variables. Non-linear systems subsume linear systems.

Fig. 2 shows the architecture of a generic model-based diagnosis approach. In this scheme, the model is used to represent the expected behavior of a system under nominal and faulty conditions. A mathematical model capturing the relation between the input signals and the output measurements is used to *track* or estimate the system outputs based on the observed measurements. The resulting differences between estimated and measured outputs, or *residuals*, are processed to detect, isolate, and identify the true fault(s). With a perfect model, and no measurement noise, a zero residual implies nominal operating conditions, while a non-zero residual implies the presence of faults. In practical applications, however, a residual is seldom zero under nominal operating conditions due to the presence of noise in the measurements and imperfections in the models employed. Hence, statistical mechanisms have been developed to mitigate the effects of modeling abstractions and measurement noise. These statistical methods increase the robustness of model-based detection and isolation approaches and reduce the generation of false alarms [26]. While early detection and isolation of faults is imperative in safety-critical systems, fault identification is important for fault-adaptive control [27] and prognosis [28]. The fault detection and isolation steps can be aggregated into one decision making scheme (e.g., see [26,29]), or solved as sequential problems (e.g., [7]), where the fault isolation module is invoked once the fault detection mechanism indicates the occurrence of a fault. Once the true fault is isolated, the magnitude or slope of this fault is estimated through fault identification.

Based on the reasoning strategy employed, model-based approaches can be broadly classified as *abductive* and *consistency-based*. Abductive approaches reason from effects to causes, while consistency-based approaches reason from causes to effect. In abductive approaches, the observed measurements are compared to the expected nominal behavior of the system, and any discrepancy between the two is explained by the diagnosis, which is defined as "a set of abnormality assumptions

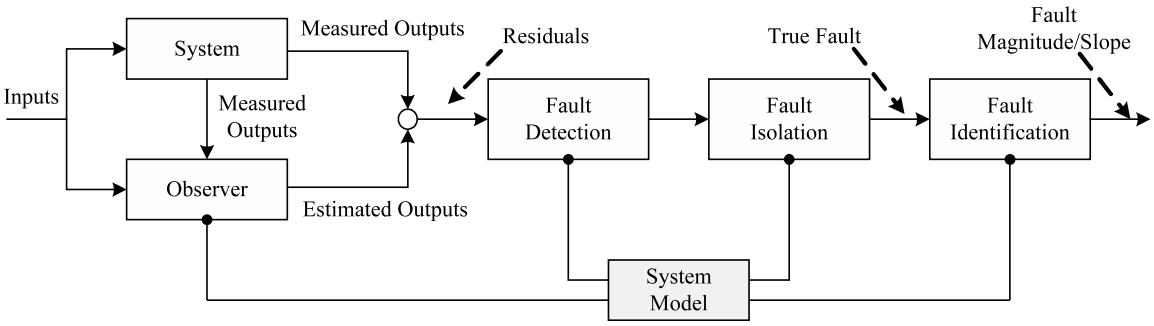


Figure 2: The architecture of a generic model-based diagnosis approach.

that covers (or, in terms of logic, implies) the observations" [30]. On the other hand, in *consistency-based* diagnosis approaches, diagnosis involves identifying the minimal set of faulty components, which along with the assumption that all other components are not faulty, makes the system model consistent with the observed sensor measurements [21, 30, 31].

Dynamic physical processes are, in reality, *continuous-time* processes, where the system behavior is defined at every instance of *dense* time. However, all automated diagnostic tools implemented on computers use sampled data. Hence, to facilitate the simulation and analysis of these models using digital computers, *discrete-time* models are developed where the system behavior is defined at discrete time points. In our work, we will focus on discrete-time models of dynamic systems. At the highest level of abstraction, a dynamic system can be represented as *discrete-event-system* (DES) representations, where time is not explicit in the system behavior. Instead, system behavior is defined by transitions between pre-defined symbolic states, and these transitions are governed by pre-defined events. At a lower level of abstraction, the system behavior can be represented using *qualitative* models, wherein the relationships between faults and symptoms, as well as evolution of system dynamics, are represented by qualitative expressions. At the next level of abstraction, the system behavior is defined using *state-space*, or *input-output* equations [2]. The diagnosis approaches that use the above mentioned models must be effective in diagnosing faults in real-world scenarios, where uncertainties created by sensor noise and modeling abstractions cannot be avoided. Uncertainty present in noise sensor readings, and inaccurate system models are captured by probabilistic and fuzzy-set driven schemes [32], or by interval methods [33]. In our research, we use the probabilistic framework for uncertainty, and review reasoning schemes based on *Bayesian* methods for robust diagnosis [3, 34]. Hence, at the lowest level of abstraction, we use models of dynamic systems which explicitly capture these uncertainties. Example of such models include Bayesian networks, hidden Markov models, and Dynamic Bayesian networks [3].

Robust Model-Based Fault Diagnosis Approaches

Uncertainties are unavoidable in the real world. There are several possible causes of uncertainty, such as *modeling inaccuracies* that can be attributed to modeling abstractions and parameter uncertainties, *sensor noise*, *disturbances*, and *process noise*. These topics are discussed in greater detail below:

1. **Modeling inaccuracies:** The main causes of modeling inaccuracies are (*i*) *modeling abstractions*, and (*ii*) *parametric uncertainties*. Models rarely capture the exact dynamics of a real system, mainly because it is seldom possible to know *everything* about a system. Furthermore, modeling complex nonlinearities present in the system in a sufficiently precise form is difficult. Hence, a system model is usually an abstraction of the actual system behavior. For example, complex nonlinear models are simplified by linearizing the parameters, or reducing the order of the nonlinearity. During abstraction or simplification, the modeler focuses on the important behaviors of the system while avoiding computational intractability, and reducing the modeling effort that would be required to capture every small detail of system behavior. As a result, the predicted model behaviors will invariably have certain differences from the actual underlying system behavior. Moreover, building a complete system model requires detailed knowledge of the system configuration and component behaviors, as well as component parameters. This knowledge is typically obtained by consulting system designers and experts, extracting information from device manuals and research papers, and using experimental data collected during system operations. When experimental data is used, unknown parameters and function relations associated with the models are estimated using system identification techniques. Methods for estimation of system parameters are seldom exact, and estimated parameter values and their functional forms are generally approximations of the actual parameter values. Inaccurate parameter estimates also result in uncertainties in system behavior. Modeling errors usually have a multiplicative effect on the system behavior.

2. **Sensor noise:** Most real world sensors are noisy, and the noise may or may not conform to known probability distributions. However, for practical purposes, noise is typically modeled as a random Gaussian white noise with known parametric or non-parametric stochastic distributions.

3. **Disturbances and Process noise:** Disturbances are usually modeled as unknown extra inputs acting on the plant. For the purpose of fault diagnosis, disturbances are considered

as "nuisance variables", the presence of which the diagnosis approach must ignore and be unaffected by. Process noise captures the difference between the actual observed evolution of state variables based on the values of the states in the previous time step as compared to the state evolution modeled by our system models.

Other causes of uncertainty may also be present. For example, in large systems, information is often carried to different parts of the system, as well as, to the reasoners, via information networks which introduce additional uncertainty, because the transmission may result in dropping of information packets and transposing of observation sequences. However, modeling these forms of uncertainty is beyond the scope of this dissertation.

1. **Robustness through modeling abstractions:** Different model-based diagnosis schemes handle these uncertainties in different ways. Discrete-event systems [5, 35, 36] and the qualitative simulation-based diagnosis schemes [37–40] address the uncertainties through modeling abstractions. The DES schemes abstract the system dynamics into a set of discrete modes and events, while the qualitative model-based diagnosis schemes abstract the system evolution in terms of qualitative differential equations and abstracted qualitative behaviors.

2. **Handling disturbances through decoupling:** The analytical redundancy relation-based approaches [2] handle uncertainties modeled as additive disturbances by decoupling their effect on the outputs through algebraic matrix manipulation.

3. **Handling parameter uncertainties by accommodating parameter variations:** Another approach to handle uncertainty is presented in [41], where the system parameters are modeled as intervals, rather than constants, and sensitivity analysis of the generated residuals is used to correctly evaluate the fault residuals for diagnosis.

4. **Handling process and sensor noise through probabilistic approaches:** Probabilistic model-based diagnosis schemes handle sensor and process noise within the same framework using probability theory that provides a mathematically sound reasoning mechanism for diagnosis under uncertainty. These approaches uses probabilistic models, such as Bayesian networks and Dynamic Bayesian networks [3] to explicitly model measurement noise and modeling inaccuracies, and use Bayesian reasoning techniques to generate correct diagnosis results in the presence of uncertainties.

In the remainder of this chapter, we present and compare the different model-based diagnosis schemes for continuous dynamic systems introduced above. We start with diagnosis approaches

that use discrete-event system models, followed by qualitative diagnosis of dynamic systems. Then we present analytical redundancy-based diagnosis schemes, followed by probabilistic model-based diagnosis schemes.

Discrete-Event Systems Diagnosis

Wonham, in [42], defines a discrete event system (DES) representation of a dynamic system to be "equipped with a state space and state-transition structure. In particular, a DES is discrete in time and (usually) in state space; it is asynchronous or event-driven: that is, driven by events other than, or in addition to, the tick of a clock". *Finite state machines* (FSMs) have been widely used for modeling DES [43]. FSMs are graphs where nodes represent states, and edges represent transitions that can be taken from one state to another if the event guarding the transition occurs [43]. Formally FSMs can be represented as a tuple $(\Sigma, S, S_0, \delta, F)$, where Σ is the input alphabet, S is a finite non empty set of states, $S_0 \subseteq S$ is the set of initial states, $\delta : S \times \Sigma \rightarrow S$ is the state transition function, and $F \subseteq S$ is the set of final states. FSMs allow for intuitive modeling of systems and match the mental models many people use to analyze complex systems [36]. Also, capturing the ordering of events is straightforward using FSMs. However, as we will show in the remainder of this section, generation of FSMs usually involves quantization of the continuous, and depending on how fine this quantization is, FSMs can suffer from high space complexity, and result in the FSM-based DES diagnosis schemes to be computationally very expensive [5, 35, 36].

Example. Consider the two-tank example shown in Fig. 3. The fluid level in the tanks represent the system state. A discrete, quantized state space representation for the tank system involves representing the liquid levels into *High*, *Medium*, *Low*, and *Empty*, as shown in the figure. Here, the state of the two tank system can be defined as an ordered pair (s_{tank_1}, s_{tank_2}), where s_{tank_1} is the quantized state of tank 1 and s_{tank_2} is the quantized state of tank 2. In total, there are sixteen states possible quantized states in the two tank system, as shown in Fig. 4, with state 1 corresponding to both tanks being *Empty*, i.e., (*Empty*, *Empty*), state 2 corresponding to the (*Low*, *Empty*) state for two tanks, and so on. One possible FSM for the two tank system is shown in Fig. 5. Assume the fluid inflow F_{in} to be constant. Under normal operation, the system starts in (*Empty*, *Empty*). Then as water flows into tank 1, the system moves to (*Low*, *Empty*) and stays in this state till the water level in tank 1 reaches pipe R_{12}. After this, the state changes to (*Medium*, *Empty*), water starts to flow into tank 2, and the system-state changes to (*Medium*, *Low*). Then, the system moves into (*Medium*, *Medium*), and finally to (*High*, *High*). The states mentioned above are the

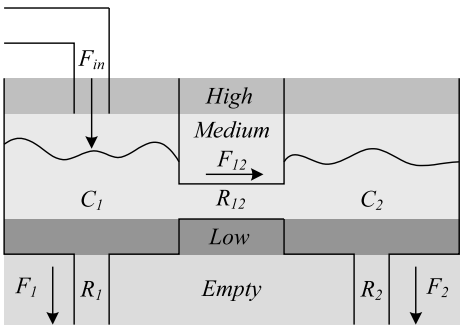


Figure 3: Schematic of a two tank system with quantized states.

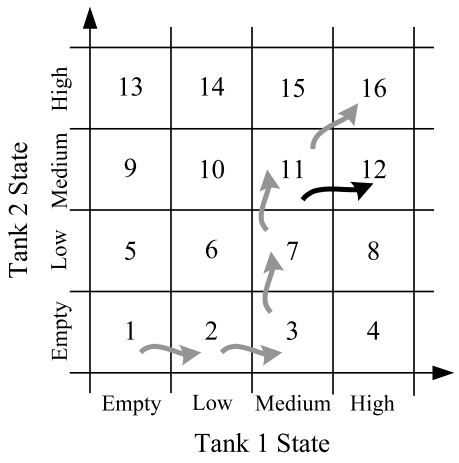


Figure 4: Quantized state-space of the two-tank system. Grey arrows depict normal transitions, the black arrow shows an unobservable faulty transition.

nominal states. States other than these can be considered faulty states and can be reached through transitions caused by faults, such as the transition from (*Medium*, *Medium*) to (*High*, *Medium*) (shown by a black arrow in Fig. 4) could be a result of an abrupt blockage in pipe R_{12}. Similarly, a leak in pipe R_1 could result in the system going to (*Low*, *Medium*) from (*Medium*, *Medium*). In the DES framework, faults are usually modeled as unobservable transition events.

The diagnosis approaches available in literature for discrete event approaches can be classified as one of two types: *event-based* and *state-based* [36]. In event-based diagnosis approaches, faults are modeled as unobservable events, otherwise, they are trivially diagnosable. Fault diagnosis in event-based frameworks typically involves inference to be made about the occurrence of unobservable failure events based on observed events. In state-based approaches, the state-space is partitioned according to the failure status of the system, into nominal and faulty modes, and the problem of fault diagnosis involves the determination of whether the system is in nominal or faulty mode based on the most recent observed measurements.

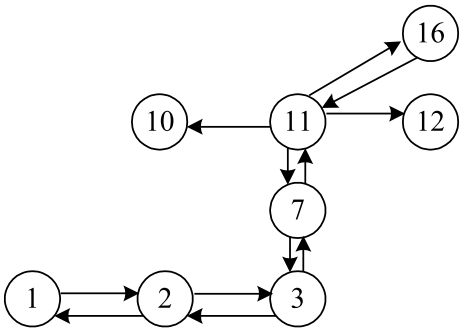


Figure 5: A FSM for the two-tank system.

An event-based DES diagnosis framework is presented in [5, 44]. The system is modeled as a FSM, and has both observable and unobservable events. Observable events can include controller commands and sensor readings. Unobservable events consist of fault events and other events which cause state transitions that are not observable by the sensors. At the core of this diagnostic methodology is a *diagnoser* that is modeled as a deterministic FSM and systematically generated offline from the system model. This diagnoser serves as an extended observer, and gives estimates of the current state of the system after the occurrence of every observable event. The transitions of diagnoser FSM are only based on observable events. The state of a diagnoser consists of estimated current system states, and a failure label which indicates whether a faulty transition of that specific type necessarily had to be taken for the system to reach the estimated state. A fault is unequivocally diagnosed when a state in the diagnoser is reached, wherein each system state estimate has the fault label corresponding to this fault.

A state-based DES diagnosis scheme is described in [36], where the system is modeled as a Moore FSM, so that each state of the system represents a system condition (or, failure mode). Being state-based, the goal of this diagnosis scheme is to identify the failure mode the system is in, rather than explicit failure events that caused the system to be in this mode, as is the case for event-based DES diagnosis schemes. State-based diagnosis schemes are useful because for most practical settings, system models are constructed by composing several smaller component models, each usually having a single nominal mode and a few failure modes. Hence, a direct relation exists between system state and the failure mode. This approach assumes that a system has a single nominal mode and several failure modes. Like [5, 44], this approach also generates a *diagnoser*, which takes as input the output sequence of the system and estimates the failure mode of the system. The states in the diagnoser contain an output, possible system states consistent with the output sequence, and possible system condition estimate associated with these states. If in a diagnoser state, an output is possible from

its set of system states, a transition is created from this diagnoser state to the next diagnoser state corresponding to that output, and the new system states that could have been reached from the possible system states. The possible system conditions for this new state is also marked accordingly. A fault state is diagnosed when it is estimated with certainty that the system is in that particular fault state.

The diagnosability property of DES-based schemes is studied in [5, 44] and [36]. A fault event or mode is considered to be *diagnosable* if it can be detected and isolated after the occurrence of a finite number of events following the failure event. In addition, [5, 44] also define the notion of *I-Diagnosability*. A system is *I-Diagnosable* if it can be detected and isolated within a finite number of events following the occurrence of an indicator event corresponding to that fault.

Even though there are similarities between the approaches presented in [5] and [36], there are some inherent differences between the two approaches. In [36], the authors assume that the states of the system can be partitioned according to the condition (failure mode) of the system. Therefore, their focus is on determining the system condition rather than detecting failure events. This approach, therefore, allows diagnosis of faults in situations where the fault event has occurred *before* the start of diagnosis. Moreover, the assumption simplifies the transition function of the state-based diagnoser in [36] because the system condition is assumed to be a function of the system state, and avoids the need for propagating fault labels, as is done in the event-based diagnoser in [5, 44].

Another state-based, consistency-based DES diagnosis scheme is presented in [35], wherein the continuous state-space is partitioned and the system is represented as a *timed* discrete-event model. A *quantizer* partitions the quantitative state-space of the system into a finite set of qualitatively similar states, each represented by a *qualitative value*. The quantizer also generates an event every time the system moves from one quantized state to another. The diagnosis approach uses the timed event sequences, along with timed input sequences, to diagnose the system. Quantization results in nondeterminism in the model, since for a known quantized state and known input, it may be possible for the system to enter more than one new quantized state, as the exact point in the quantized space is not known. To represent the non-determinism in its behavior, the author models the system as a *semi-Markov process* in a compact manner. Based on the sequence of inputs and possible events, the diagnoser computes the probabilities of faults, and as more events occur, the probability of the true fault will increase in value, while for other faults, the probability should eventually become negligible. If at the end of diagnosis, no fault can be uniquely diagnosed, the relative ranking of the probabilities gives an indication of the likelihood of different fault hypotheses.

Discussion

DES models handle robustness to uncertainties by abstracting away details from continuous systems, and representing the system dynamics in terms of discrete states and events. Once abstracted, the DES diagnosis schemes can be applied to these DES models without additional mechanisms for handling uncertainty. Note, however, that decisions about whether the system is in a particular state or where a particular event has occurred must still be taken in the presence of uncertainties in system dynamics at the lowest level of abstraction. As a result, the task of handling such uncertainties is delegated to these decision tasks, so that once it is determined, taking into account uncertainties, that a system is in a particular state and a certain event has occurred, the DES diagnosis approaches can generate correct diagnosis results in the presence of uncertainties without explicitly having to handle such uncertainties.

The abstraction of details in a DES has a trade-off. If the level of abstraction is very high compared to the actual system behavior, information crucial to fast and accurate diagnosis of faults is lost. A detailed DES model at a lower level of abstraction can overcome this drawback, but would increase the size of the model and the computation time for the DES diagnosis schemes. Moreover, the model may be difficult to develop. Generation of DES models require quantization of the continuous system state-space. Quantization can be leveraged to build both untimed and timed models. Timed DES models capture information about system dynamics beyond that obtained from a simple ordering of events. Quantization seems to be appropriate for systems with discrete inputs, sensors and discrete faults. If the sensors are not discrete, quantization loses information. Moreover, quantization based approaches suffer from state explosion depending on the resolution of quantization of its state-space. In addition, to use these approaches, faults have to be quantized as well. Also, if the faults are possible in any state of the system, then the DES model becomes very large. In [1], the author presents an approach for constructing a DES model for continuous systems by systematically abstracting the dynamics of the observed measurements in the presence of different faults to relatively avoid the exponential blow-up of states and state-transitions. The DES diagnosis schemes provide a well-developed framework for event-based and state-based fault diagnosis. But, they lack sophisticated mechanisms to handle measurement noise and unknown disturbances that cannot be avoided in practical scenarios. In addition, the performance of DES diagnosis schemes depend on the order in which the input and output events are observed, and if the order in which events are generated by the system is not the same as that observed by the diagnoser, or if some

observed events are missing, the DES diagnosis approaches may fail to generate correct diagnosis results.

Diagnosis Using Qualitative Models

Qualitative models express the relationships between observed symptoms and the faults, as well as the system dynamics in terms of qualitative functions and equations [45]. The qualitative diagnosis schemes leverage the cause-effect relation behavior for different faults captured in the qualitative models to correctly isolate faults. Rather than focusing on expert systems which use $IF-THEN-ELSE$ rules [34], we will focus our discussion on those qualitative model-based diagnosis schemes that are derived from first-principles and a sound understanding of the physics of the system. In the following, we present a few qualitative model-based diagnosis approaches.

Several qualitative diagnosis schemes, such as [37, 38, 46, 47], use signed digraphs (SDGs) for diagnosis. An SDG is a directed graph whose nodes represent deviation from the steady state of a variable, and signed arcs represent the relationships between these nodes [38]. SDGs are much more compact than truth tables, decision tables, or finite state models.

Example. For example, if we denote the fluid-level of a tank as H, its inlet flow as F_{in}, its outlet flow as F_{out}, and the resistance of its outlet pipe as R_{out}, then the equations to represent this system are [45]:

$$\begin{aligned} F_{in} - F_{out} &= \frac{dH}{dt} \\ F_{out} &= \frac{H}{R_{out}}. \end{aligned}$$

The corresponding SDG for this system is given in Fig. 6. An external change causing the flow rate F_{in} to change, would cause dH and H to change in the same direction, which will in turn cause F_{out} to also change similarly. However, this change in the F_{out} would cause the dH to change in the other direction, implying a feedback. The SDG can be obtained by abstracting the mathematical model of the underlying process.

In [38], the authors derive a cause-effect (CE) graph from a system's SDG. The CE graph consists of only valid nodes (i.e., nodes which are abnormal) and consistent arcs (i.e., arcs which explain the local propagation of the fault and hence, the observed symptom). The sign of the nodes in a SDG can be considered as a pattern which may match a particular fault condition, and a fault is isolated if the system's SDG shows the corresponding pattern. If the signs of some of the nodes are not

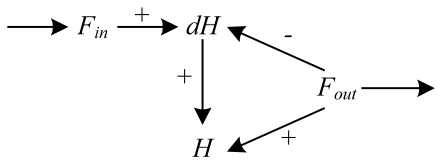


Figure 6: Signed digraph for a simple tank system.

known, a partial pattern is formed, and this indicates the possibility of all possible fault conditions the partial pattern corresponds to being present. In [37], the arcs gains of SDGs are allowed to vary dynamically, thereby allowing modeling of nonlinearities. In [46], the authors address the issue of conditional arcs in SDGs. SDGs can get quite complicated for complex systems, and in [47], the authors present techniques to simplify the SDGs for fault diagnosis. A rule-based approach to diagnosis using SDGs is presented in [48], where logical statements, or *rules*, such as, *IF−(SDG rule premise)−THEN−(possible fault)*, are automatically derived from SDGs and these statements are evaluated using online data to generate the diagnosis result. These automatically derived rules can also be integrated with other rules using an expert system framework, such as forward chaining [34]. A method for qualitative analysis of causal feedback in SDGs is proposed in [49] to resolve feedback ambiguities, and this method is implemented in a simulator for qualitative ordinary differential equations, called QUAF.

A qualitative simulation (QSIM)-based diagnosis approach, MIMIC, is presented in [39,40], where the system is modeled using qualitative differential algebraic equations (QDAEs). QDAEs are a general, implicit form of qualitative differential equations (QDEs) [50]. Given a set of ordinary differential equations (ODEs), QDEs can be considered as an abstraction of these ODEs, and represented by a set of qualitative constraint equations which are satisfied by any behavior that satisfies the given ODEs [50]. The constraint equations consist of a set of symbols representing the process variables, and a set of constraints on how these variables may be related to each other. The constraints allow the expression of simple mathematical relationships between the variables such as addition, multiplication, and differentiation. The evolution of the system is qualitatively simulated (see [50]), where essentially, the infinite number of numeric behaviors of a system is discretized into a smaller set of qualitatively distinct behaviors or states, and qualitative transitions exist between states, depending on the constraints.

In MIMIC, given the nominal model of the system, QSIM starts at the initial state, and as and when observations change, the qualitative model is simulated based on the observations, predicting the immediate successor qualitative states the system might possibly be in. In addition, QSIM also

generates the quantitative ranges of values for each model parameter. Then, the model constraints are utilized to identify the permissible next states. Further reduction of the possible next states is obtained by employing tests to detect when steady-state conditions are reached, or when a cyclic behavior is observed. If multiple successor states are present, the simulation branches. Finally, based on both qualitative and quantitative values, a similarity function computes the similarity between the observations and behavior predicted by the nominal system model. For the next iteration of the hypothesize-and-match loop, only those models are retained whose similarity to the observations exceeds a threshold. A fault is detected when the normal model of the system is discarded. An earlier version of MIMIC used a decision tree method to generate the new hypothesis. The corresponding fault models was then selected from a set of pre-enumerated fault models and initialized as per the current observations. Then the QSIM continued as described above. A new version of the MIMIC framework does not generate new hypotheses using decision trees. This implies that pre-enumeration of faults is not required. As a result, MIMIC is no more restricted to the diagnosis of only those faults that were pre-enumerated and MIMIC was trained for. In the updated version, MIMIC tries to modify the system model, before discarding it, in an attempt to make its predictions agree with the observations. MIMIC's algorithm identifies all components and parameters that could have contributed to the discrepancy, the only valid suspects being those that account for all discrepancies. In addition, MIMIC further tests the suspects' global consistency through constraint-suspension. A suspect is exonerated if no assignment of values consistent with all symptoms is found. For each of the remaining suspects, each of its operating modes is tested for compatibility with the observations, and discarded if necessary. The remaining modes are now tracked by qualitative simulation, till only the true fault model survives and tracks the observations of the (faulty) system.

Discussion

Qualitative models are abstractions of the analytic models that rely on quantization of the continuous state-space. The modeling abstractions also help these approaches to be robust to uncertainties, by not requiring the actual qualitative model-based diagnosis schemes to explicitly handle uncertainties to generate correct diagnosis results.

Moreover, these methods suffer from nondeterminism and state explosion, just like DES-based diagnosis approaches do. Similar to the DES diagnosis schemes, depending on the level of quantization, qualitative diagnosis methods may have to deal with intense quantitative analysis at run-time. However, the lack of precise quantitative information introduces ambiguities in the solutions, such

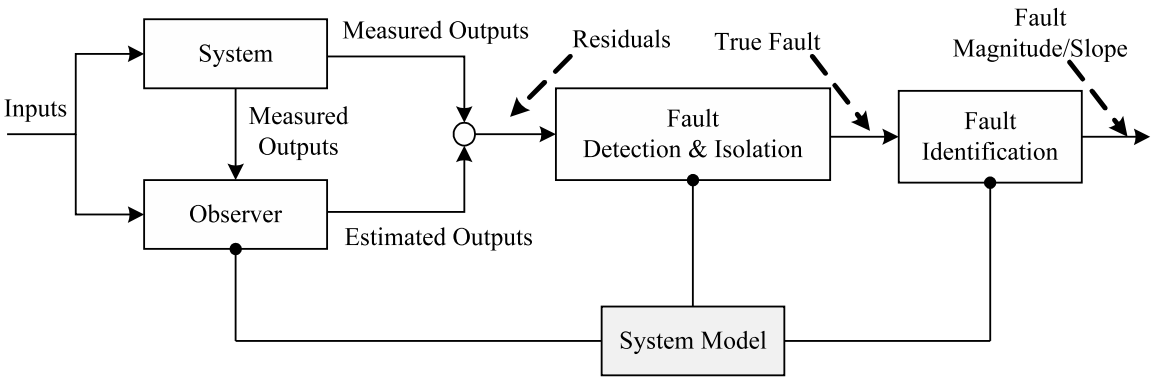


Figure 7: Architecture of a generic model-based diagnosis approach where detection and isolation of faults are combined.

as two different faults having the same qualitative effect. Quantitative analysis of the observed measurements can help resolve ambiguities that may be present in qualitative reasoning methods. Another drawback of qualitative diagnosis approaches are spurious solutions, i.e., solutions which are not physically realizable. Spurious solutions can be avoided, to a reasonable extent, by imposing strict constraints, and modeling the system from different perspectives. Moreover, qualitative diagnosis schemes may produce multiple possible behavior predictions because of the nondeterminism that may be introduced due to quantization. Also, based on the degree of quantization, a fault may not cause the system to move to a different qualitative state, thereby resulting in delayed, or, missed detection and isolation. The qualitative diagnosis schemes do not have any specific techniques to handle uncertainties, such as measurement noise and modeling abstractions.

Analytical Redundancy Relations-Based Diagnosis Scheme

Analytical redundancy relations (ARR)-based diagnosis scheme is a quantitative diagnosis approach that involves the generation of inconsistencies between actual and expected behavior, also known as *residuals*. Checking for inconsistency requires some form of redundancy, and in ARR-based approaches, the redundancy is *analytical*, and achieved from a fundamental understanding of the dependence among the process variables. The residuals are expected to be close to zero in the absence of faults, and show significant non-zero values otherwise. Hence this cause-effect relationship between faults and residuals can be analyzed to generate diagnostic conclusions. Several diagnosis approaches for continuous systems are based on the notion of ARRs. As shown in Fig. 7, ARR-based approaches perform the detection and isolation tasks together.

In observer-based ARR schemes, observers track the system behavior closely, based on the system model and observed measurements, and the residual $r(t) = y_i(t) - \hat{y}_i(t)$ is defined as the difference between the estimated and observed measurements. While Unknown Input Observers (UIO) [51] allow for generation of residuals in deterministic settings, in stochastic settings, Bayesian observers, such as Kalman filters and Extended Kalman filters [34] can be used to estimate system states correctly in the presence of uncertainties, to generate residuals. Although UIOs are typically applicable to linear systems, a similar design and analysis approach can result in unknown input observers for nonlinear systems [52]. To detect and isolate faults, a set of observers is developed such that an observer is sensitive to a subset of faults, while insensitive to the remaining faults and unknown inputs. The presence of measurement and model redundancy results in extra degrees of freedom to build such observers. The main idea is that for fault-free conditions, the observer tracks the system behavior closely, generating residual values close to zero. Once a fault occurs, however, the system outputs estimated using observers that are sensitive to this fault will show significant non-zero residuals, while the system outputs estimated using other observers will produce statistically insignificant residual values. Depending on the design of the set of observers, distinct residual patterns are developed for each fault, making fault isolation possible.

Parity relations are another form of ARRs, which involves rearranged and transformed variants of the I/O or state-space model of the system that allow the checking of parity (or consistency) using sensor outputs and known process inputs only. The idea of this approach is to rearrange the model structure so as to get the best fault isolation. In parity relations, the residual $r(t) = \mathbf{w}^T\mathbf{Y}(t) - \mathbf{w}^T\mathbf{R}\mathbf{U}(t)$, where $\mathbf{U}(t)$ is system input, $\mathbf{Y}(t)$ is measurements, and $\mathbf{w}$ and $\mathbf{R}$ are matrices, can be structured such that one measured variable has no impact on a specific residual by choosing the $\mathbf{w}^T$ matrix appropriately. It has been shown, in [53], that residual generators developed based on parity relations and observers are identical or equivalent. The residuals, once generated, need to be selectively responsive to particular faults, while unaffected by other faults, to obtain the desired fault detection and isolation properties. Therefore, residuals are *enhanced* to support fault isolation by providing well-defined responses to particular faults. There are three main schemes of enhancing the residuals, namely *diagonal residuals*, *directional residuals*, and *structured residuals* [2]. Diagonal residuals are designed such that each element of the residual vector responds to one and only one fault. Directional residuals are generated such that the residuals are confined to a fault specific direction in the multidimensional residual space. As a result, the fault isolation step requires the determination of a predefined direction to which the residual is the closest. In structured residuals, each residual element responds selectively to a subset of faults. The residual structures are

characterized by incidence matrices, whose columns correspond to individual faults, rows correspond to residuals, and whose elements determine whether a residual is affected by a fault.

Example. For example, given three possible faults $\{F_1, F_2, F_3\}$, a possible incidence matrix [54]:

$$\begin{pmatrix} & F_1 & F_2 & F_3 \\ r_1 & I & I & 0 \\ r_2 & 0 & I & I \\ r_3 & I & 0 & I \end{pmatrix}$$

where the I element indicates that the residual responds to the fault while, a 0 indicates otherwise. Each column of the incidence matrix can be termed as the *fault signature*. Hence, a fault is not detectable by a residual structure if the corresponding column in the incidence matrix contains all zeros. Also, two faults are not distinguishable from each other by a residual structure if their fault signatures in the incident matrix are identical, i.e., the columns in the incident matrix corresponding to these two faults are identical.

Parameter estimation methods can be used to diagnose parameter drifts which are not directly measurable. In practical systems, it is very unlikely that the parameters of the system model is known completely. Either the parameters are not known at all, or at best, known partially. However, if the basic model structure of the system is known, these parameters can be estimated by measuring the input and output signals. The process model used for parameter estimation is obtained using only the measured inputs and outputs, and is of the form

$$\mathbf{y}(t) = f(\mathbf{u}(t), \theta).$$

The model parameters θ can be estimated as $\mathbf{y}(t)$ and $\mathbf{u}(t)$ are measured using techniques such as least squares, instrumental variables, and estimation via discrete-time models [19, 55]. Changes in the parameters can be related to process faults.

Discussion

Depending on the residual generation scheme, ARRs apply different approaches to handle uncertainties. If the residuals are generated using UIO observers, ARRs handle uncertainties in a deterministic setting by using matrix manipulation to make the generated estimates to be insensitive to unknown disturbances [2]. A similar matrix manipulation is used in parity relations-based residual generation

schemes [56]. If, however, ARRs use stochastic observers for generating residuals, these observers usually handle uncertainties due to sensor and process noise using probabilistic schemes, as we explain in the next section.

The main issue with ARRs is that one has to wait till all residuals have fired in order to correctly isolate the true fault. For example, consider the incidence matrix below:

$$\begin{pmatrix} & F_1 & F_2 & F_3 \\ r_1 & I & I & I \\ r_2 & 0 & I & I \\ r_3 & I & 0 & I \end{pmatrix},$$

where, if the true fault is F_3, we must wait for all three residuals, r_1, r_2, and r_3 to fire before this fault can be correctly isolated. However, consider a scenario where the true fault is F_2. In this case, when residuals r_1 and r_2 fire, we can refine the set of possible faults to F_2 and F_3. However, as indicated in the incidence matrix, r_3 will never fire if F_2 is the true fault. Hence, to diagnose F_2, we would have to impose additional heuristics, such as assuming that if a residual r_j is sensitive to fault F_i, then residual r_j will fire within $t_{r_j}^{F_i}$ time steps from the time of occurrence of fault F_i. Hence, once the first residual fires, we would wait for $t_{r_j}^{F_i}$ for each fault F_i and residual r_j before making a decision on what the true fault is. Also, typically a binary test is performed on the residual, and hence, ARRs do not provide sufficient discriminatory power. One way to address this drawback is to use the sign of the residual, as well as its transient behavior for fault isolation, as done in [7]. From the above discussion, it is clear that the ARR-based approaches concentrate mainly on the tasks of fault detection and isolation, and parameter estimation techniques need to be implemented for identification of faults. The models used for ARR-based diagnosis are usually limited to linear models, though they have been adapted for nonlinear systems as well [57]. Linear approximations may prove to be poor for highly nonlinear systems, and hence these approaches may not be effective for such nonlinear systems. Also, ARR-based diagnosis methods cannot guarantee that the residuals will be able to detect a fault that has not been specifically modeled.

Probabilistic Diagnosis Schemes

Probabilistic reasoning methods have been used in several diagnosis algorithms [3,58,59]. Probability theory provides mathematically sound reasoning mechanisms based on a numerical *degree of belief*

(between 0 and 1) associated with hypotheses and measurements (i.e., evidences) in a diagnostic scheme.

The fundamental problem we seek to solve in a probabilistic diagnosis is to determine the chance of a particular fault occurring given the observed systems. This question, however, is counterintuitive, since our knowledge about the real world is causal. In other words, domain experts usually have a fairly good intuition about the chances of seeing a particular symptom given a fault in the system, e.g., the chances of having a headache if someone has fever. However, trying to ascertain the chances of the fault happening given a particular effect, e.g., the chances of someone having a fever given he/she has a headache, is somewhat counter intuitive, and the precise question we ask in a diagnosis problem. In general, Bayes' theorem provides the fundamental mechanism for diagnosing faults in the presence of uncertainty, by relating symptoms to faults [34]. For example, assuming $Symptom$ and $Fault$ are two random variables, the posterior probability of $Fault$ given $Symptom$, $P(Fault|Symptom)$ can be ascertained from "intuitive", causal information such as $P(Symptom|Fault)$, and prior probabilities $P(Fault)$ and $P(Symptom)$ as follows:

$$P(Fault|Symptom) = \frac{P(Symptom|Fault)P(Fault)}{P(Symptom)}.$$

Example. As a simple example, consider a single-tank system as shown in Fig. 8. This tank has an input fluid flow source, F_{in}, and an output pipe, R_1, at the bottom. A blockage in the output pipe could cause the fluid level in the tank to rise abnormally. Let the opening in the output pipe be represented by the random variable, $Pipe$, and the height of fluid in the tank is represented by the random variable, $Height$. Let each random variable assume two discrete values, e.g., $Pipe = \{nominal, blocked\}$, and $Height = \{nominal, increase\}$. The value of $Height$ depends on $Pipe$. It is common knowledge that the fluid level in the tank is highly likely to increase in the event of a blockage in output pipe, i.e., say $P(Height = increase|Pipe = blocked) = 0.9$. We also know that, to start with, chances of the pipe getting blocked, or fluid level abruptly increasing is fairly slim, i.e., say $P(Pipe = blocked) = 0.1$ and $P(Height = increase) = 0.3$, respectively. Notice however, inferring the chances of the pipe being blocked given the fluid level in the tank has increased does not follow easily from common "causal" domain knowledge. However, using Bayes' theorem, we can calculate this probability as follows:

$$\begin{aligned} P(Pipe = blocked|Height = increase) &= \frac{P(Height = increase|Pipe = blocked)P(Pipe = blocked)}{P(Height = increase)} \\ &= \frac{0.9 \times 0.1}{0.3} = 0.3 \end{aligned}$$

Thus, observing an increase in fluid level increases the chances of a blockage in the pipe threefold compared to when the knowledge of pipe blockage was not available.

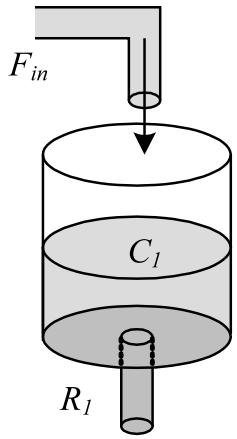


Figure 8: Schematic of a one tank system.

If additional symptoms are observed, the likelihood values of the fault can be further updated as shown below:

$$P(Fault|Symptom_1, \ldots, Symptom_n) = \frac{P(Symptom_1, \ldots Symptom_n|Fault)P(Fault)}{P(Symptom_1, \ldots, Symptom_n)}.$$

If each variable has two possible values, and there are n *Symptom* variables, calculating probability $P(Symptom_1, \ldots, Symptom_n|Fault)$ requires the knowledge of conditional probabilities for 2^n possible combinations of observed symptoms. Hence, there is no considerable savings over using the full *joint probability distribution* for inference instead. In general, the full joint distribution can be written as

$$\begin{aligned} P(Fault, Symptom_1, \ldots, Symptom_n) &= \prod_{i=1}^{n-1} P(Symptom_i|Symptom_{i+1}, \ldots, Symptom_n, Fault) \\ &\times \quad P(Fault)P(Symptom_n|Fault). \end{aligned}$$

Assuming that the single hypothesis directly influences the evidences, all of which are conditionally independent given the cause, the joint probability distribution can be decomposed into

$$P(Fault, Symptom_1, \ldots, Symptom_n) = P(Fault)\prod_{i=1}^{n} P(Symptom_i|Fault).$$

Such a probability distribution is called the *Naive Bayes* model [34] as it is often used as a simplifying assumption in cases where the $Symptom_i$ variables are not conditionally independent given the $Fault$ variable.

The above example illustrates the usefulness of probabilistic techniques to reason under uncertainty. We also saw how the probabilities of different hypotheses are updated as more evidence is obtained. The storage of joint probability distributions and marginalization of unobserved variables,

however, are exponential. In the naive approach, assuming each evidence variable is conditionally independent from other evidence variables given the hypothesis reduces the computational complexity and the need for a large number of probability values. However, this assumption is very strict and may not always be correct. The lack of causal dependencies between variables, is a good indication of independence, and can be exploited in several graphical models for correct and efficient inference, as explained below.

Graphical models such as Bayesian networks (BNs) and Dynamic Bayesian networks (DBNs) explicitly model uncertainty and graphically represent efficient factorizations of the joint probability distributions over a set of variables. This is possible because these models capture the multiple causal dependencies, as well as, the independence between different random variables. The notion of independence is used to perform diagnosis in a tractable manner. In this paper, we focus on the model-based diagnosis of dynamic systems. BNs assume the state of the system to be static, and do not explicitly model the dynamic states of a system, and the transition between these states. DBNs extend BNs by including this temporal information. In this chapter, we present different model-based diagnosis schemes that use stochastic models of dynamic systems, and show how these stochastic diagnosis approaches extend the deterministic diagnostic techniques presented earlier in this chapter. We will, however, first discuss BNs briefly for the sake of completeness.

Model-based Diagnosis using Bayesian Networks

BNs are directed acyclic graphs (DAGs) where the nodes are random variables, and the arcs specify the direct probabilistic influences between the random variables [60]. BNs are static causal models, with an arc between two nodes representing a causal relationship between the source node, the "cause", and the destination node, the "effect". There are two types of nodes in a BN: *observable* and *hidden*. A node in a BN is observable if it can be measured. The hidden nodes either represent variables that are not measured, and can include fault hypotheses. The diagnostic reasoning procedure involves inferring the likelihood of the unobservable fault hypotheses variables based on the observed evidence. The random variables in a BN can be discrete, with a finite range of possible values, or continuous, and can be specified by a finite number of parameters that correspond to standard families of probability density functions.

The construction of a BN must ensure that it is a complete, yet compact, and correct representation of the full joint distribution of all random variables. In [34], the authors suggest that the correct order in which to add the nodes in a BN is to add the *root causes* first, then the variables they

influence, and so on, until we reach the "leaves" which have no direct causal influence on the other variables. This construction of a locally structured BN is not a trivial problem, and usually requires guidance from a domain expert as well as knowledge engineers who are well versed in Bayesian models. Once the BN is constructed, the next step is to assign appropriate probability distribution tables to the nodes again using expert knowledge or empirical data collected in the domain, e.g., empirical knowledge of frequency of observed symptoms given particular faults.

Example. Fig. 9 shows a highly simplified BN for the two tank system shown in Fig. 3. Each node labeled $R_i Increase$ is unobservable, and represents a possible fault, i.e., a blockage in pipe R_i, where $i \in \{1, 2, 12\}$. For simplicity, we assume these random variables take on two discrete values, $True$ and $False$. The flow status through the pipe F_i is dependent on $R_i Increase$. In addition, flows F_1 and F_2 affects F_{12}. Finally, the height in tank 1, $Tank_1 Height$, depends on the inflow F_{in}, and flows F_1 and F_{12}. Similarly, $Tank_2 Height$ is determined by F_{12} and F_2.

If $Tank_1 Height$ and $Tank_2 Height$ are the only observed (measured) discrete nodes, the diagnosis problem can be formulated as to determining $P(R_i Increase = True | Tank_1 Height = t_1, Tank_2 Height = t_2)$ for each $i = \{1, 2, 12\}$ and identifying that $R_i Increase$ as the true fault, for which this probability is the highest. t_1, t_2 could be discrete $High$, $Nominal$ or Low values.

The above probability can be computed by marginalization from the joint probability distribution of all variables, with the values for $R_i Increase$, $Tank1 Height$ and $Tank_2 Height$ instantiated. In BNs, this joint probability distribution can be computed efficiently by utilizing the conditional independence between nodes. Any two variables A and B are conditionally independent given C if $P(A, B|C) = P(A|C)P(B|C)$. In a BN, there are two specifications of conditional independence [34]: (i) a node is conditionally independent of its non-descendants, given its parents, and (ii) a node is conditionally independent of all other nodes in a network, given its parents, children, and children's parents, i.e., its *Markov Blanket*.

Hence, once a BN with nodes $\{X_1, X_2, \ldots, X_i\}$ are constructed, the aforementioned notions of conditional independence, along with the necessary probability information, is used to represent and compute the joint probability distribution in an efficient manner:

$$P(X_1, \ldots, X_{i-1}, X_i) = \prod_{i=1}^{n} P(X_i | X_{i-1}, \ldots, X_1) = \prod_{i=1}^{n} P(X_i | Parents(X_i)).$$

This factorization of the joint probability distribution results in considerable reduction in the number of probability values or distributions needed to perform the marginalization calculations, as well as the saving of space and computational complexity when computing the conditional probabilities.

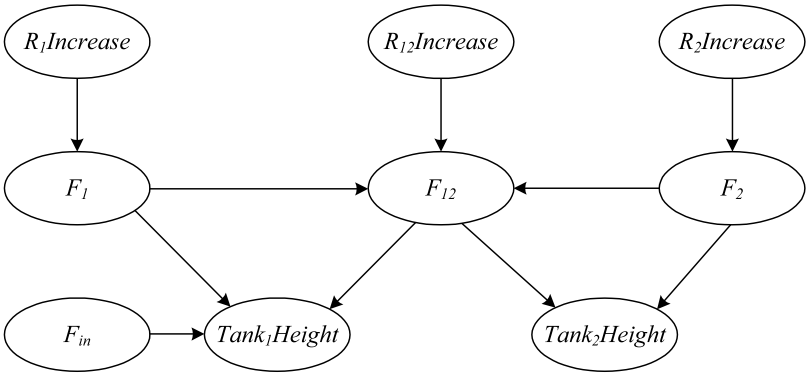


Figure 9: A Simplified Bayesian Network for the two-tank system.

BNs have been used for diagnosis in a number of application domains such as, medical diagnosis [61, 62], and communication networks [63, 64], among others. Formally, the diagnosis procedure using BNs involves evidence gathering and *belief updating*, i.e., the computation of the *belief functions*, the conditional probability of a node taking a specific value given the available evidence. Belief functions of the nodes reflect the overall belief accorded to that node value by all evidence that has been collected so far. Exact inference in BNs is NP-hard, except for the class of singly connected Bayesian networks. For other classes of Bayesian networks, three approximate approaches of belief updating have been developed, namely *clustering*, *simulation*, and *conditioning* [60, 65]. In clustering, compound nodes are formed in such a way that the resulting networks of clusters is singly connected. Every BN can be structured as singly connected if the size of the clusters is not limited. Simulation techniques involve an approximate solution to the evaluation of belief functions by using Monte Carlo techniques to estimate probabilities by counting how frequently events occur over a series of simulation runs [66, 67]. Conditioning involves breaking the loops in a BN by instantiating a selected set of nodes to reduce the network to being singly connected, so that a polynomial belief updating algorithm can be applied, and then, properly aggregated with the different value instantiations.

As mentioned above, diagnosis using BNs involves belief updating in uncertain causal networks as more observations become available. BNs are based on a well founded mathematical model, and the reasoning is consistent. The main disadvantage of a BN is that it does not directly accommodate relations between variables that evolve over time. As a result, dynamic systems cannot be modeled easily using BNs. DBNs explicitly model temporal evolution of system state in a Bayesian framework and allow for Bayesian reasoning with dynamic systems.

In the following, we define DBNs, and describe how diagnosis problems are formulated in terms of DBNs. Typically, the diagnosis problem in the DBN-framework requires estimation of unobserved state variables in the system based on observed measurements. Therefore, we present two different state estimation approaches, and describe some diagnosis schemes based on each estimation approach. The main state estimation schemes we describe are Kalman filtering [34], which is an optimal state estimation approach, but restricted to a linear Gaussian systems; and particle filtering, which is a more general approach applicable to any general DBN [12].

Model-Based Diagnosis Using Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBNs) [3] provide a systematic method for modeling the dynamics of complex systems in the presence of uncertainties. A DBN is a compact, directed acyclic graph structure that represents a probabilistic discrete-time model of a dynamic system. Nodes in the graph represent random variables. Links in a DBN are of two types: (i) those that denote causal dependencies between nodes in a particular time step, and (ii) those that capture causal relations across time steps. The absence of a link between two nodes imply that these nodes do not causally affect each other directly.

Definition 2 (Dynamic Bayesian Network)**.** Formally, we define a Dynamic Bayesian Network (DBN) as $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, where $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{Y}$ are sets of stochastic random variables that denote (hidden) state variables, system input variables, and measured variables of the dynamic system, respectively[1]. Graphically, a DBN is a two-slice Bayesian network, representing a snapshot of system behavior in two consecutive time slices, t and $t+1$. Each DBN time-slice represents the observation model, $P(\mathbf{Y}_t|\mathbf{X}_t, \mathbf{U}_t)$ derived from causal links $X_t \rightarrow Y_t$ and $U_t \rightarrow Y_t$, where $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, $U \in \mathbf{U}$, and subscript t represents time. Across-time causal links $X_t \rightarrow X_{t+1}$, $X_t \rightarrow X'_{t+1}$, and $U_t \rightarrow X_{t+1}$, where $X' \in \mathbf{X}$, represent the first order Markov state-transition model, $P(\mathbf{X}_{t+1}|\mathbf{X}_t, \mathbf{U}_t)$.

Our graphical representation of DBNs in this dissertation include thick-lined circles that denote state variables, thin-lined circles that denote measured variables, and squares that denote input variables. The Markov assumption significantly reduces the number of time slices required for representing the complete system evolution, and allows for a compact description of system evolution using two time slices. However, across multiple time steps, the relations propagate, and eventually, within a finite number of time steps, all state variables causally connected to each other. In other words, DBN reasoning is exponential, in general.

[1] We chose to use different notation for our DBNs than that presented in [3] for simplicity, but essentially, both imply the same DBN structure, as presented in [3].

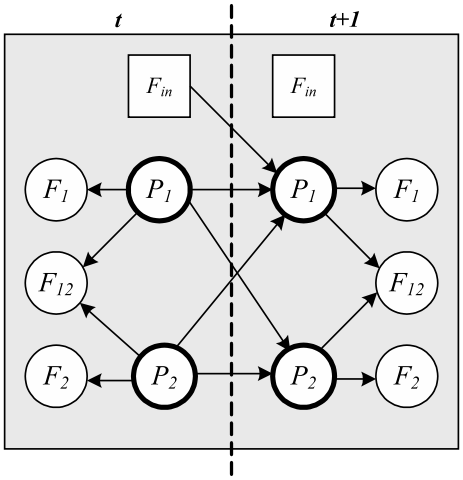


Figure 10: Dynamic Bayesian network of a two-tank system.

Example. Fig. 10 shows the DBN for the two-tank system shown in Fig. 3. The state variables in the two-tank system are the pressures at the bottom of tanks 1 and 2, and represented by the variables P_1 and P_2, respectively, i.e., $\mathbf{X} = \{P_1, P_2\}$. Flows F_1, F_{12}, and F_2 are the measured variables, i.e., $\mathbf{Y} = \{F_1, F_{12}, F_2\}$, and the flow in, F_{in}, is the input variable, i.e., $\mathbf{U} = \{F_{in}\}$. Therefore, the DBN in Fig. 10 can be represented as $D = (\{P_1, P_2\}, \{F_{in}\}, \{F_1, F_{12}, F_2\})$. In the two tank system, the flow F_{12} at the current time instance depends on the pressure difference in tanks C_1 and C_2 at the current time instance. Hence, the DBN contains the intra-time-slice causal links $P_{1_t} \rightarrow F_{12_t}$ and $P_{2_t} \rightarrow F_{12_t}$. The inter-time-slice causal link $P_{1_t} \rightarrow P_{2_{t+1}}$ and $P_{2_t} \rightarrow P_{2_{t+1}}$ represents the dependence of the value of P_2 at the current time step on the values of P_1 and P_2 at the previous time step. Similarly, links $P_{1_t} \rightarrow F_{1_t}$ and $P_{2_t} \rightarrow F_{2_t}$ denote the dependence of flows F_1 and F_2 on P_1 and P_2, respectively. The absence of a causal link from F_{in} to P_2 implies that volume P_2 at a time step does not depend on the value of the inflow F_{in} at the previous time step.

In DBNs with exclusively discrete random variables, the prior and conditional probabilities are expressed using probability distribution tables. For continuous systems, the prior and conditional probabilities are expressed using probability density functions. DBNs are the most general of graphical probabilistic models. DBNs exploit the conditional independence among variables to provide a compact and factored representation of a dynamic system. DBNs do not impose any restrictions on using arbitrary probability distributions to describe the dynamic evolution of system behavior, or sensor noise [3].

DBNs have been used for fault diagnosis in hydraulic systems [58], planetary rovers [13], and robots [4], amongst others. In the DBN framework, the diagnosis problem can be mainly formulated

as a probabilistic inference problem, which involves the estimation of the value of the unobservable state variables, $\mathbf{X}_t$ at time t, based on the observed measurements, $\mathbf{Y}_{1:\tau}$, from time 1 to τ, i.e., computing the marginal $P(\mathbf{X}_t|\mathbf{Y}_{1:\tau})$. If $\tau = t$, the inference problem is termed *filtering* or state estimation, if $\tau > t$, the inference problem is termed *smoothing*, and if $\tau < t$, the inference problem is termed *prediction* [3].

One formulation of the diagnosis problem is to determine the most likely sequence of states that best explains the observed evidences. In other words, the diagnosis problem is posed as the classic *decoding* problem. Since DBN nodes represent more than one state, this problem reduces to the determination of the sequence of hidden states which maximizes the probability of the observed evidence, i.e., $\hat{\mathbf{X}}_{1:t} = \arg\max_{\mathbf{X}_{1:t}} P(\mathbf{X}_{1:t}|\mathbf{Y}_{1:t})$. This task can be achieved using the Viterbi algorithm [68].

In this dissertation, we focus mainly on the diagnosis problems that are posed as a Bayesian state estimation or filtering problems, and involve determining the value of $P(\mathbf{X}_t|\mathbf{Y}_{1:t})$, where the fault is included as one or more unknown state variables. For example, in [58], the authors present a DBN-based diagnosis scheme, where every parameter that can be faulty is included as an unobservable continuous random variables in the system DBN model. The DBN model also includes discrete random variables that capture whether or not a parameter is faulty. Given this hybrid DBN model, the diagnosis scheme involves solving the state estimation problem, i.e., tracking the observed measurements to estimate the values of the system parameters correctly.

The first order Markov assumption allows a iterative solution of the state estimation problem, using a two-step *predict* and *update* procedure [12]. Suppose, the probability distribution at time $t-1$ is available, i.e., $P(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$ is known. Then we first "predict" the state variable values at the next time-step, i.e.,

$$P(\mathbf{X}_t|\mathbf{Y}_{1:t-1}) = \int P(\mathbf{X}_t|\mathbf{X}_{t-1})P(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})d\mathbf{X}_{t-1}. \tag{1}$$

Then, based on the measurement values observed at time t, we "update" or modify the prior density obtained in the prediction stage using Bayes' theorem to obtain the posterior density as follows:

$$P(\mathbf{X}_t|\mathbf{Y}_{1:t}) = \frac{P(\mathbf{Y}_t|\mathbf{X}_t)P(\mathbf{X}_t|\mathbf{Y}_{1:t-1})}{P(\mathbf{Y}_t|\mathbf{Y}_{1:t-1})}, \tag{2}$$

where the normalizing constant is

$$P(\mathbf{Y}_t|\mathbf{Y}_{1:t-1}) = \int P(\mathbf{Y}_t|\mathbf{X}_t)P(\mathbf{X}_t|\mathbf{Y}_{1:t-1})d\mathbf{X}_t. \tag{3}$$

Equations 1 and 2 form the basis of many Bayesian diagnosis schemes. However barring a few restrictive cases, these two equations cannot be solved exactly, in an analytical manner. Linear time invariant (LTI) systems with only Gaussian random variables allow for an optimal solution to equations 1 and 2. Kalman filters assume that the posterior density at every time step is Gaussian, i.e., $N(\mu, \sigma^2)$, where μ is the mean and σ^2 is the variance of this distribution. and hence provides an optimal and exact solution for $P(\mathbf{X}_t|\mathbf{Y}_{1:t})$, parametrized in terms of the mean and variance [12]. However, for nonlinear systems, and systems with arbitrary probability distributions, the Kalman filtering approach does not apply, and a closed form solution for propagating the distributions across causal links is not possible. For such systems, particle filtering [12], a sequential Monte Carlo simulation-based scheme that approximates the optimal solution, form the state of the art. Extended Kalman filtering can be applied for nonlinear systems, but, with only Gaussian random variables. In the following, we first present Kalman filtering-based diagnosis schemes, followed by particle filtering-based diagnosis approaches.

Diagnosis Using Kalman and Extended Kalman Filters

If every random variable in a DBN is sampled from a Gaussian normal distribution, and the system is linear, Kalman filtering allows for closed form exact solution to the state estimation equations (see equations 1 and 2). A detailed tutorial on Kalman filters can be found in [12].

Kalman filter assumes that $P(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ is Guassian, and the system is defined as

$$\begin{aligned} \mathbf{X}_t &= A_t\mathbf{X}_{t-1} + B_t\mathbf{U}_{t-1} + Q_{t-1} \\ \mathbf{Y}_t &= C_t\mathbf{X}_t + R_t, \end{aligned} \tag{4}$$

where A_t, B_t and C_t are known matrices defining linear functions, and Q_{t-1} and R_t represent the process noise and sensor noise covariance matrices, respectively. All these matrices can be time varying.

Then, equations 1 and 2 can be viewed as:

$$P(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}) = N(m_{t-1|t-1}, P_{t-1|t-1}) \quad (5)$$

$$P(\mathbf{X}_{t}|\mathbf{Y}_{1:t-1}) = N(m_{t|t-1}, P_{t|t-1}) \quad (6)$$

$$P(\mathbf{X}_{t}|\mathbf{Y}_{1:t}) = N(m_{t|t}, P_{t|t}), \quad (7)$$

where,

$$m_{t|t-1} = A_t m_{t-1|t-1} + B_t \mathbf{U}_{t-1} \quad (8)$$

$$P_{t|t-1} = Q_{t-1} + A_t P_{t-1|t-1} A_t^T \quad (9)$$

$$m_{t|t} = m_{t|t-1} + K_t(\mathbf{Y}_t - C_t m_{t|t-1}) \quad (10)$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \quad (11)$$

and where, the covariance of the innovation term, $\mathbf{Y}_t - C_t m_{t|t-1}$, is defined as $S_t = C_t P_{t|t-1} C_t^T + R_t$, and the Kalman gain, $K_t = P_{t|t-1} C_t^T S_t^{-1}$. In the above equations, M^T denotes the transpose of a matrix, M.

If the system is nonlinear, but still Gaussian, *extended Kalman filtering* (EKF) can be applied to generate approximate solutions to the state estimation problem [69]. The details of EKF implementation can be found in [12]. The basic idea in EKFs is that since, the system cannot be represented using the linear matrices shown in equations 4, the EKF linearizes the nonlinear functions using the first term in the Taylor series expansion of the nonlinear functions representing the state transition and observation models, and then applies standard Kalman filter predict and update steps on this linearized system.

Parametric faults can be diagnosed using Kalman filter-based parameter-estimation techniques. In this method, the fault parameter is introduced as an extra "state-variable" in the above Kalman filter formulation, and the fault parameter in the other equations is replaced by this new "state-variable". This extension introduces nonlinearity, and transforms the above formulation into an extended Kalman filter. If there are $k > 1$ possible faults of interest, we design $k + 1$ separate Kalman filters, one for the normal operation, and each of the remaining k filters for modeling one of the k possible faults. A fault is detected when the observed measurements deviate from the estimated values of the system output indicating off-nominal behavior. Recall that a residual is the difference in the measured values of the system outputs and their estimated values obtained from the

Kalman filter. The fault isolation task, then, involves the estimation of the outputs of the system by each filter, and whichever filter generates the minimum residual is considered to represent the true state of the system. In this scheme, the magnitude of the true fault can also be estimated. This approach is also referred to *multiple-model* diagnosis scheme.

Sometimes, a *bank of filters* is used for diagnosis, where each Kalman filter exclusively represents either the faulty or nominal system behavior. Different A, B, C, and D matrices are used to model the different sensor or actuator faults. A sensor fault can be characterized as "hard" (i.e., the sensor reading is assumed to be stuck at a certain value and the measurements available from it are ignored), or "soft" (i.e., the sensor is degraded in quality, but not completely useless). Given n state variables, a hard failure in the second sensor can be modeled by an C matrix with a null second row, i.e.,

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{pmatrix},$$

where c_{ij} represents an element of C. Soft failures, on the other hand, are modeled by scaling the sensor noise covariance matrix R to represent the power increment of the measurement noise [70]. While modeling soft failures, the C matrix remains unchanged from the one used in the nominal system model. Likewise, given p actuator inputs, and n state-variables, a hard failure of the second actuator can be modeled by a B matrix with a null second row, i.e.,

$$B = \begin{pmatrix} b_{11} & 0 & \dots & b_{1p} \\ g_{21} & 0 & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{n1} & 0 & \dots & b_{np} \end{pmatrix}.$$

Again, b_{ij} represents an element of B. Once the different fault are modeled by different C and B matrices, residuals can be generated for each Kalman filter, with the model which generates the minimum residual being identified as the true faulty model. However, real data is always noisy and the naive approach might not be of much help as as it does not contain any information about the validity of the residual information. One way to avoid this problem is to invoke statistical tests, such as the standard hypothesis testing procedures, to isolate the true fault [70] . Another improvement

over the above approach is the use of a back propagation neural network which gets to process these residuals and identify the true fault [71].

In [71], the state-variables $\mathbf{X}$ represent a vector of fault deltas. The A matrix is an identity matrix, indicating the persistence of faults. The measurement estimates are given by the equations $\mathbf{Y} = C\mathbf{X} + \theta$, where C is the matrix of fault *influence* coefficients, and the random vector θ denotes the uncertainties inherent in the measurement process. The C matrix can be computed as shown in [72]. In this approach, for each fault, the corresponding Kalman filter is invoked with a different set of parameters chosen to accentuate that particular fault's root cause. In this way, a measurement estimate is obtained for each fault under consideration. Once all the estimates are obtained, they are ranked based on a normalized measurement error norm for each measurement estimate. The single fault admitting the minimum error is deemed the most likely root cause.

Another technique for diagnosing faults, especially additive sensor/actuator faults include the isolation of faults using a *bank of estimators* [73]. In this scheme, under the hypothesis that the input sensors are fault-free, a classical Kalman estimator is designed for each output sensor fault, which is driven by *only* that output sensor and all inputs of the system. Hence if there are m output sensor faults to be diagnosed, m different Kalman estimators are set up. If there is a fault on the ith output sensor, then only that Kalman estimator driven by this sensor reading will generate incorrect estimates, thus affecting the residual function of that particular output observer. For example, if there are n state-variables in the system and p inputs, the Kalman estimator for the residual corresponding to failure of the second sensor (of m sensors in total) will have

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \; B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{np} \end{pmatrix}, \text{ and } C = \begin{pmatrix} 0 & 0 & \dots & 0 \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

In order to isolate input sensor (or actuator) faults, however, the design of the bank of Kalman filters is slightly different. Under the assumption that the output sensors are fault-free, a bank of unknown input Kalman filters is used, where in the ith observer is driven by *all but* the ith input sensor and all outputs of the system. Each of these observers generates a residual function which is sensitive to all but the ith input sensor fault. An input sensor fault is easily isolated since a fault in the ith sensor will affect all but the ith residual function. For example, if there are n state-variables in the system and m output sensors, the Kalman estimator for the residual corresponding to failure

of the second actuator (of p actuators in total) will have

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \; B = \begin{pmatrix} b_{11} & 0 & \dots & b_{1p} \\ b_{21} & 0 & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{n1} & 0 & \dots & b_{np} \end{pmatrix}, \text{ and } C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{nn} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{pmatrix}.$$

Diagnosis Using Particle Filters

Particle filtering is the most general, sequential Monte Carlo scheme for state estimation in systems that are either nonlinear, or non-Gaussian, or both. A particle filter assumes that the state and measurement vectors, $\mathbf{X}$ and $\mathbf{Y}$, respectively, can be modeled as:

$$\begin{aligned} \mathbf{X}_t &= f(\mathbf{X}_{t-1}, \mathbf{U}_{t-1}, \mathbf{v}_{t-1}) \\ \mathbf{Y}_t &= h(\mathbf{X}_t, \mathbf{n}_t), \end{aligned}$$

where $f(\cdot)$ and $h(\cdot)$ are nonlinear functions of the state and input vectors, $\mathbf{v}_{t-1}$ denotes an i.i.d. process noise sequence vector, and $\mathbf{n}_t$ is an i.i.d. measurement noise sequence vector.

The details of particle filtering schemes can be found in [12]. The main idea in particle filtering is to represent the posterior density approximates the belief state of a system using a weighted set of N samples, or *particles*, $\{\mathbf{X}_t^i, w_t^i\}_{i=1}^N$, where $\mathbf{X}_t^i$ is a set of *support* points with weights w_t^i, $i = 1, 2, \dots, N$, and t represents the simulation time step. The weights are normalized such that $\sum_i w_t^i = 1$. The particle filtering approach computes the estimates of the state variables based on these support points and weights. Given the weighted particles, under the first order Markov assumption, the posterior density at time t can be approximated as

$$P(\mathbf{X}_t|\mathbf{Y}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(\mathbf{X}_t - \mathbf{X}_t^i), \tag{12}$$

where $\delta(\cdot)$ is the Dirac delta function.

The principle of *importance sampling* [12] is used to choose the weights of the particles. Let $p(x)$ denotes a probability density such that direct sampling from $p(x)$ is intractable. In such a scenario, samples can be drawn from an *importance density* function, $q(x)$, instead, i.e., $x^i \sim q(x)$, $i = 1, \dots, N$, and the weighted approximation of density $p(x)$ can be obtained by

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i),$$

where

$$w^i \propto \frac{p(x^i)}{q(x^i)}$$

is the normalized weight of the ith particle.

Assuming that we start at time t with $P(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ known, if we choose an importance density function that factorizes as follows:

$$Q(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t}) = Q(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{1:t})Q(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1}),$$

then, we can obtain samples $\mathbf{X}_t^i \sim Q(\mathbf{X}_t|\mathbf{Y}_{1:t})$ by augmenting each of the existing samples $\mathbf{X}_{t-1}^i \sim Q(\mathbf{X}_{t-1}|\mathbf{Y}_{1:t-1})$ with the new state $\mathbf{X}_t^i \sim Q(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{1:t})$. The weight of the particles at time t is defined as:

$$w_t^i \propto w_{t-1}^i \frac{P(\mathbf{Y}_t|\mathbf{X}_t^i)P(\mathbf{X}_t^i|\mathbf{X}_{t-1}^i)}{Q(\mathbf{X}_t^i|\mathbf{X}_{t-1}^i, \mathbf{Y}_t)}.$$

As time progresses, each particle is moved stochastically to a new state, and the observations are used to re-adjust the weights on each particle to reflect the likelihood of the observation given the particle's new state. Highly weighted particles indicate likely states of the system. However, an issue of particle filters is the *degeneracy* problem [12], where, after a few iterations, all but a very few particles are assigned negligible weights. As a result, a large amount of computation is used in updating particles with negligible weights. Choosing a good importance sampling function, and *resampling* are possible solutions to the degeneracy problem.

Resampling eliminates particles with small weights and replicates particles with large weights, by generating new set of particles $\{\mathbf{X}_t^{i*}\}_{i=1}^N$ by "resampling" with replacement N particles from the approximate discrete representation of $P(\mathbf{X}_t|\mathbf{Y}_{1:t-1})$ (shown in equation 12) with probability of choosing a particle being equal to the weight of the particle, i.e., $Pr(\mathbf{X}_t^{i*} = \mathbf{X}_t^j) = w_t^j$ [12].

Different particle filtering-based diagnosis algorithms are presented in [4, 13, 74–77]. In [77], an approach for applying PF to DBNs is presented. In PF-based diagnosis frameworks, faults can be modeled as explicit states. The fault diagnosis problem can then be formulated as the recursive state estimation of a system based on the sequence of measurements. A particle filter based diagnosis algorithm is presented in [13], where the authors use particle filtering for real-time fault detection in planetary rovers. In [76], the authors present another PF-based algorithm for hybrid state estimation.

Discussion

Probabilistic diagnosis schemes handle measurement and process noise by explicitly modeling and reasoning with them. For dynamic systems, both Kalman and particle filters have explicit variables representing process and measurement noise variance. However, while Kalman filters apply to linear Gaussian systems, particle filters can be applied to nonlinear non-Gaussian systems as well, and hence, most general. Moreover, in these schemes, we consider each system variable to be a random variable, assume a distribution about each parameter and system variable, and use Bayesian reasoning approaches to infer correct and accurate diagnosis results in terms of probability distributions, in the presence of uncertainties.

As mentioned above, diagnosis using BNs involves belief updating in uncertain causal networks as more observations become available. BNs are based on a well founded mathematical model, and the reasoning is consistent. The main disadvantage of a BN is that it does not directly accommodate relations between variables that evolve over time. As a result, dynamic systems cannot be modeled easily using BNs. DBNs explicitly model temporal evolution of system state in a Bayesian framework and allow for Bayesian reasoning with dynamic systems.

In Kalman filter-based diagnosis schemes, the *multiple-model* approaches require the pre-numeration of faults of interest, which can be difficult. In addition, those approaches, which do not use parameter estimation, are not suitable for diagnosing different magnitudes of the same fault. On the other hand, the parameter estimation based implementation can handle the isolation and identification of faults of different magnitudes. Some multiple-model approaches also allow isolation of more than one simultaneous sensor failures, and they permit handling sensors with different data rates using a sequential update scheme [78].

The *bank of estimators* methods of Kalman filters allow for diagnosis of multiple faults, albeit only for faults in the output sensors, wherein the presence of multiple faults will be indicated with multiple residual functions tending to zero. However, for input sensor faults, these approaches cannot perform multiple fault diagnosis because in the presence of multiple faults, *no* residual function will approach zero. Also, the success of the bank of estimators scheme for output sensors depends on the *observability* property of the system. A system is termed *observable* if the unobserved state variables in the system can be accurately estimated based on the measurements that can be observed (see Chapter V for details). The bank of estimators approach will fail unless the measurements are chosen such that the system is observable using each single measurement. On the other hand, the design of the bank of estimators for input sensor diagnosis is more robust. Yet, in both these schemes,

it is required to ensure that the removal of the dependencies on particular sensor readings do not compromise the observability of the system.

As shown above, once the residuals are generated, the efficient and correct analysis of these residuals is important. If the sensor to noise ratio is large, then one can simply identify the minimum residual and isolate the sensor which corresponds to this residual function as the true fault. However, this method can give incorrect results when the sensor measurements are noisy, or when the filter with the least residual is not the correct fault, as is shown in [79]. In that case, hypothesis testing, or other classification algorithms, such as artificial neural networks, can be invoked to intelligently isolate the true fault. However, the problem with these classification methods is that they are not scalable.

Finally, it is interesting to see how faults are represented in these approaches. While mostly the equations of a Kalman filter model the dynamics of the physical process, some approaches model abstract states with the Kalman filter equations [71]. In such cases, the bulk of computation is in finding the *influence factor* matrix that relates the measurements to these abstract states.

Several reasons contribute to the popularity of particle filters for fault diagnosis using DBNs. We have already mentioned that particle filters can be applied to nonlinear models with arbitrary prior belief distributions. Moreover, particle filters are *contract anytime* algorithms, i.e., if available computation time is specified in advance, the PF algorithm can estimate a belief distribution in the available time, by changing the number of particles [13]. In fact, an important property of particle filters is that the computational requirement of a particle filter depends only on the number of samples, and not on the complexity of the model.

However, while PF has proven very successful in tasks such as visual tracking [80] and robot navigation [81], they are less suited for diagnosis tasks. This is because, for diagnosis, we are most interested in tracking fault states, which initially have very low probability of occurring. As a result, during the resampling step, there is always the risk of losing these particles with low weights that might represent the fault state, when the fault occurs. This results in *sample impoverishment* [4,13] and the system may never diagnose the system to be faulty since there will be negligible particles, if at all, representing these fault states. In [13], importance sampling has been suggested in as possible solution to the above issue. Two other solutions to the sample impoverishment problem are presented in [4]. These solutions include risk-sensitive particle filters (RSPF) and variable-resolution particle filter (VRPF). In RSPF, a model of cost is factored in when generating particles. Since fault states have high cost, the RSPF ensures that particles are generated to represent them, even if they have low probability. In VRPF, multiple similar states are tracked by a single abstract particle,

allowing a limited number of particles to sufficiently represent large portions of the state space when the likelihood of occupying that part of the state-space is low. When the likelihood of the grouped state increases, the abstract particles are refined to represent individual states.

Summary

In this chapter, we presented four types of model-based diagnosis schemes, namely, DES-based approaches, qualitative schemes, ARR-based approaches, and probabilistic schemes. Each of these approaches handle uncertainties in different ways. DES and qualitative diagnosis schemes handle robustness to uncertainties by abstracting away details from continuous systems, and representing the system dynamics in terms of discrete states and events; and a set of qualitative differential equations, respectively. Once abstracted, the respective diagnosis schemes can be applied to these abstracted models without additional mechanisms for handling uncertainty. Note, however, that decisions about whether the system is in a particular state, or, where a particular event has occurred, or, which qualitative state the system is in, must still be taken in the presence of uncertainties in system dynamics at the lowest level of abstraction. As a result, the task of handling such uncertainties is delegated to these decision tasks, so that once it is determined, taking into account uncertainties, that a system is in a particular state and a certain event has occurred, the diagnosis approaches can generate correct diagnosis results in the presence of uncertainties without explicitly having to handle such uncertainties.

Depending on the residual generation scheme, ARRs apply different approaches to handle uncertainties. If the residuals are generated using UIO observers, ARRs handle uncertainties in a deterministic setting by using matrix manipulation to make the generated estimates to be insensitive to unknown disturbances. A similar matrix manipulation is used in parity relations-based residual generation schemes. If however, ARRs use stochastic observers for generating residuals, these observers usually handle uncertainties due to sensor and process noise using probabilistic schemes.

Probabilistic diagnosis schemes handle measurement and process noise by explicitly modeling and reasoning with them. For dynamic systems, both Kalman and particle filters have explicit variables representing process and measurement noise variance. However, while Kalman filters apply to linear Gaussian systems, particle filters can be applied to nonlinear non-Gaussian systems, and hence, most general. Moreover, in these schemes, we consider each system variable to be a random variable, assume a distribution about each parameter and system variable, and use Bayesian reason-

ing approaches to infer correct and accurate diagnosis results in terms of probability distributions, in the presence of uncertainties.

Modeling for diagnosis is a crucial aspect to all model-based diagnosis approaches. ARR-based approaches use standard state-space or input-output formulations of the system which quantitatively represent the continuous dynamic system behavior. These models are complete, and reasoning using such models involve intense quantitative analysis. Qualitative diagnosis models, on the other hand, are abstractions of the underlying analytic models and do not have to deal with intense quantitative analysis at runtime. However, inherent nondeterminism present in qualitative models generally results in maintaining a large number of possible qualitative evolution traces, or *interpretations* [48], which can be computationally very expensive. DES models for representing large continuous system can suffer from state-explosion. It is interesting to note that as we abstract away from the underlying mathematical models of the system, the diagnosis approaches become computationally less complex, but suffer from ambiguities. However, there is a cost involved in this abstraction, and these tasks, such as quantization of the continuous state-space or events is not trivial. In other words, to get information in the required qualitative form, a lot of analysis needs to be performed on the sensed quantitative measurements, and hence the complexity of the approach is transfered from online analysis to offline design phase.

Furthermore, while qualitative and DES diagnosis approaches mainly perform detection and isolation tasks, ARR and DBN-based approaches perform online estimation, detection, isolation and identification of faults. Table 1 summarizes these features of the different approaches.

DBNs are the most general framework for modeling dynamic systems to facilitate under uncertainty. Particle filtering approaches using DBNs permit inference using arbitrary distributions, and since their complexity is determined by the number of particles used, rather than the size of the system, these particle filtering approaches are the state of the art in online diagnosis using DBNs of large real-world systems. However, it must be noted that the standard particle filtering scheme with resampling are not well suited for diagnosis, and causes sample impoverishment issues. Kalman filters are a special case of particle filters, and provide exact, optimal solution to the posterior density function. However, Kalman filters are restricted to linear Gaussian systems.

As mentioned earlier, the diagnosis approaches mentioned in this chapter account for uncertainties through different approaches. Our approach makes use of probabilistic modeling and Bayesian inference techniques to handle uncertainties due to modeling imperfections and sensor noise. Moreover, our qualitative fault isolation scheme, which we present in Chapter III is a form of ARR approach, but instead of simply determining if a residual is affected by a fault to isolate the true

Table 1: Summary of Related Work

Approaches	Diagnostic Task			
	Estimation	Detection	Isolation	Identification
Discrete-Event		✓	✓	
Qualitative		✓	✓	
ARR	✓	✓	✓	✓
DBN	✓	✓	✓	✓

fault, our qualitative isolation scheme makes use of how a fault affects the dynamics of the measurement residuals for fault isolation.

CHAPTER III

THE TRANSCEND DIAGNOSIS APPROACH

TRANSCEND is a model-based approach for centralized diagnosis of abrupt faults in continuous systems[1] [7, 8]. This chapter provides a brief overview of the TRANSCEND continuous diagnosis scheme, and extends the existing scheme to analyze abrupt and incipient faults in a common framework [84, 85].

The observer-based TRANSCEND diagnosis approach combines a quantitative fault detection scheme with a novel qualitative fault isolation approach to isolate sensor and process faults in continuous dynamic systems. TRANSCEND also includes a quantitative fault identification scheme that estimates the extent of fault in the system, and in the process, helps refine the diagnosis results when there are ambiguities in the qualitative fault isolation scheme. Fig. 11 illustrates the computational architecture of the TRANSCEND fault diagnosis approach. The observer, formulated as an extended Kalman filter [69], takes as input the control signals and sensor measurements to track nominal system behavior. The differences between the observed and estimated values of measurements form residuals. Ideally, non-zero residuals should indicate the presence of faults, but the presence of uncertainties, such as sensor noise, makes the fault detection task more complicated. For fault detection, TRANSCEND employs a standard statistical hypothesis-testing scheme to determine if the non-zero residuals are statistically significant. The detection of a fault triggers the qualitative fault isolation process, which performs the tasks of *symbol generation* and *hypothesis generation and refinement*. The symbol generation module converts the magnitude and slope of the residual into qualitative symbols that are expressed as increases and decreases from nominal. The symbols are used to *generate* one or more single fault hypotheses, i.e., possible faults that could explain the generated symbols. Fault signatures capture the effect each generated fault hypothesis has on the individual measurements at the time of fault occurrence. *Hypothesis refinement* then is performed using a progressive monitoring scheme that compares the fault signatures to the symbols generated by the symbol generator for each measurement, removing any fault hypothesis whose fault signature for a particular measurement is inconsistent with the observed symbols. When the fault hypotheses are reduced to a small number, a quantitative fault identification step is invoked for each hypothesis, which computes the fault magnitude and the corresponding mean square error between predicted

[1] Extensions of TRANSCEND for the diagnosis of hybrid systems have been presented in [82, 83], but a discussion of these hybrid diagnosis algorithms is beyond the scope of this dissertation.

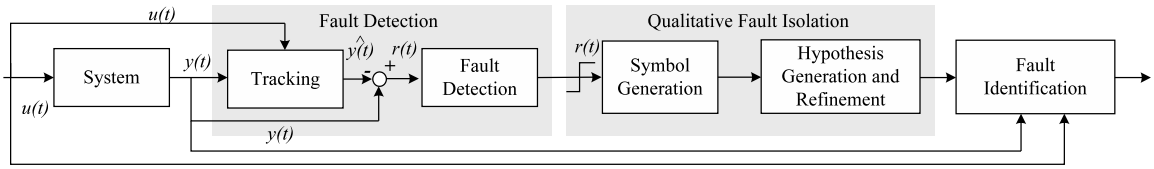


Figure 11: Computational architecture of the TRANSCEND fault diagnosis approach.

and observed behaviors. The fault parameter that produces the least mean square error in the measurements is listed as the true single fault in the system.

In the next section, we present the modeling required for diagnosis using TRANSCEND. *Bond graphs* [18] form the core of our modeling framework. Both the state-space equations required by the observers, as well as the *temporal causal graphs* (TCGs) for qualitative fault isolation [7], are automatically derived from these bond graphs. Bond graphs are domain-independent, energy-based, topological models that capture energy exchange pathways in physical processes [18]. They allow for physical systems modeling from first principles, and encode causal and temporal information that are helpful in fault isolation [18]. TCGs are used to implicate possible causes for observed deviations (from nominal) in measurements at the point of fault detection, and also to predict the effects of different faults on the measurements. These predictions are used in our hypothesis refinement scheme for fault isolation. We present in detail the different steps of the TRANSCEND diagnosis scheme. Then, we present our extensions to TRANSCEND for the diagnosis of incipient faults. Specifically, we present how we have extended the hypothesis generation and fault signature generation schemes to allow generation of incipient fault hypotheses and derivation of fault signatures for these incipient fault hypotheses. The incipient fault signatures can be used seamlessly with TRANSCEND's existing qualitative fault isolation scheme for the qualitative diagnosis of incipient and abrupt faults in a unified framework. Finally we present an analysis of the diagnosability properties of the TRANSCEND qualitative diagnosis scheme.

Modeling for Diagnosis

Bond Graphs

Bong graphs form the core of the modeling scheme used in TRANSCEND. Bond graphs allow for modular, multi-domain, physics-based, parameterized, component-based modeling of physical processes that accommodate nonlinear behaviors [18]. We have also developed extensions to the component-oriented modeling scheme in the Fault Adaptive Control Technology (FACT) paradigm [86] that

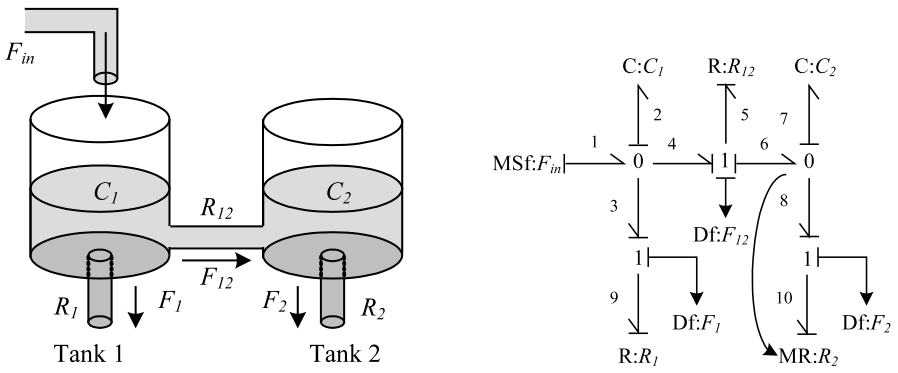
allow for explicit parametrized representation of sensors and actuators in the system. Extensions of bond graphs to hybrid modeling [87] are beyond the scope of this thesis work.

A bond graph (BG) is a directed graph, whose edges are termed *bonds*, and whose vertices are termed *bond graph elements*. Bonds represent energy pathways between the BG elements, and are drawn as half arrows ($\rightharpoonup$). Each bond, specified by a bond number i, i.e., $\overset{i}{\rightharpoonup}$, has an associated "across" effort variable, e_i, and a "through" flow variable, f_i, such that $e_i \cdot f_i$ defines the power, i.e., the rate of energy flow through the bond. The BG modeling language allows for multi-domain modeling in a common framework. Hence, the effort and flow variables map to different physical variables in the different physical domains. For example, effort and flow variables map to voltage difference and current, respectively, in the electrical domain; force and velocity, respectively, in the mechanical domain; pressure difference and volumetric (or mass) flow rate, respectively, in the hydraulic domain; and temperature difference and rate of flow of heat, respectively, in the thermal domain[2].

Bonds connect to the different BG elements through ports. Hence, BG elements can be classified as one-port, two-port, and multi-port elements. The one-port elements include energy dissipative resistor elements (denoted by R:R, such that $e_i = Rf_i$ if $\overset{i}{\rightharpoonup}$R:$R$), energy storage elements, such as capacitors (denoted by C:C, such that $\dot{e_i} = \frac{1}{C} f_i$ if $\overset{i}{\rightharpoonup}$C:$C$) and inductors (denoted by I:I, such that $\dot{f_i} = \frac{1}{I} e_i$ if $\overset{i}{\rightharpoonup}$I:$I$), and energy source elements, such as sources of effort (denoted by Se:u, such that $e_i = u$ if $\overset{i}{\leftharpoondown}$Se:$u$), and sources of flow (denoted by Sf:u, such that $f_i = u$ if $\overset{i}{\leftharpoondown}$Sf:$u$). Two-port elements perform transformation of energy and include transformers (denoted by TF:n, such that $e_i = ne_j$ and $f_j = nf_i$ if $\overset{i}{\rightharpoonup}$TF:$n$ $\overset{j}{\rightharpoonup}$) that transform energy in the same domain, and gyrators (denoted by GY:r, such that $e_i = rf_j$ and $e_j = rf_i$ if $\overset{i}{\rightharpoonup}$GY:$r$ $\overset{j}{\rightharpoonup}$) that transform energy in different domains. If the component parameters take on constant values, the bond graph models a linear time invariant system [18]. For nonlinear systems, parameter values can also be functions of time, other variables in the system, or external control variables. In our work, we assume that our systems can be nonlinear, time-varying systems. Nonlinearities are typically modeled by *modulated* elements that we define below.

Two idealized multi-port junction elements, the 0-junction and the 1-junction, connect the one- and two-port bond graph elements and satisfy the principles of conservation of energy and continuity of power by allowing for lossless energy transfer between two or more BG elements, such that at a 0-junction (respectively, 1-junction), the efforts (respectively, flows) of all incident bonds are equal,

[2]These variables define the pseudo bond graph paradigm in the thermal domain. The true bond graph paradigm defines entropy as the effort variable.



(a) Schematic. (b) Bond graph.

Figure 12: The example nonlinear two-tank system.

and the sum of flows (respectively, efforts) is zero. The direction of the bonds determine the signs of the efforts (respectively, flows) in the summation at a 1-junction (respectively, 0-junctions). The state space equations of a system can be derived from the system bond graph.

Other than bonds and BG elements, signal links drawn as full arrows (i.e., $\rightarrow$) can also be present in a BG. Each signal link allows for the exchange of information, and transfers the value of an effort or flow variable between different parts of the BGs. *Modulated elements*, i.e., BG elements whose parameters are algebraic functions of other system variables, or external signals, can be used to model nonlinearities in a BG as described above, and also capture time-varying input to the physical system. Signal links pointing from the internal and external variables to a modulated component represents the modulation graphically. Modulated elements have the prefix 'M' added to their names, e.g., MR:R denotes a modulated resistor. Signal links also carry the value of effort or flow variables to the *sensors* or *energy detector* elements, such as effort detectors (denoted by De:u, such that $u = e_i$ if $\overset{i}{\rightharpoonup}$ 0$\rightarrow$De:u) and flow detectors (denoted by Df:u, such that $u = f_i$ if $\overset{i}{\rightharpoonup}$ 1$\rightarrow$Df:u) [88].

Example. Fig. 12(b) shows the BG of a nonlinear two-tank fluid system connected by a pipe, with a source of fluid flow into the first tank, and drain pipes at the bottom of each tank (see Fig. 12(a)). In the bond graph modeling paradigm, tanks are modeled as capacitors, and pipes are modeled as resistors [18]. Pipes R_1 and R_2 drain tanks C_1 and C_2, respectively, and pipe R_{12} connects tanks C_1 and C_2. In the hydraulic domain, the effort variables denote pressure, and flow variables denote volumetric fluid flow rates. The pressure at the juncture where pipe R_i, $i \in \{1, 2\}$ connects to tank C_i, is the same as that at the bottom of tank C_i. Hence, C_i and R_i are connected by the equal-effort 0-junction. The rate of flow of fluid in pipe R_{12} depends on the pressure difference between the

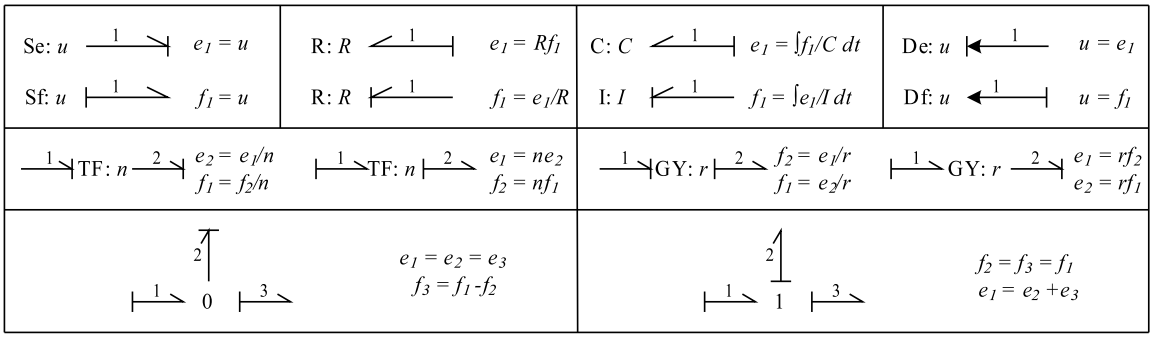


Figure 13: Possible causal assignments and corresponding constituent equations of bond graph elements.

two tanks, and hence, it is connected to the two tanks through an intermediate 1-junction. The flows through the three pipes, R_1, R_{12}, and R_3 are measured using flow detectors or sensors, Df:F_1, Df:F_{12}, and Df:F_2, respectively. The nonlinearity is introduced by the nonlinear modulated resistor, MR:R_2, which we define as a function of the pressure at the bottom of tank C_2, i.e., $R_2 = 3e_7^2$. The signal link drawn from the 0-junction connected to C_2 to this modulated resistor denotes this dependence of R_2 on e_7, the common effort at that 0-junction.

Causality

BGs allow for the systematic generation of equations to describe the dynamics of a continuous system. Basically, the constituent equations in each BG element forms part of a system of differential algebraic equations. More efficient computational forms, such as state-space equations and block diagrams can be derived from BGs through using the notion of *causality*. Causality represents the cause-effect dependencies between effort and flow variables at every BG element. In other words, causality assigns the computational directions to the effort and flow variables. Causality is denoted by a causal stroke on one end of a bond, with the BG element near the causal stroke imposing flow on the BG element away from the causal stroke.

Example. In Fig. 12(b), the causal stroke near R_{12} on bond 5 implies that the resistor R_{12} imposes flow on the adjacent 1-junction. Similarly, the causal stroke away from capacitor C_2 on bond 7 implies that C_2 imposes effort on the adjacent 0-junction.

Energy storage elements can be assigned both integral and derivative causality. For example, the constituent equation for a capacitor in integral causality is $e_i = \frac{1}{C}\int f_i dt$, i.e., the effort, e_i, is computed by integrating the flow, f_i, and the equation for a capacitor in derivative causality is $f_i = C\frac{de_i}{dt}$, i.e., the flow, f_i, is computed from the derivative of e_i. Hence, while in integral causality,

the effort is computed from the flow, in derivative causality, the flow is computed based on the effort. Derivative causality is usually not preferred because of several issues, such as, the derivative of a step input is infinity, and the requirement of future data points (this makes the model non causal) for correct computation of flow. Hence, we assume that our systems are designed to operate in integral causality. Derivative causality for energy storage elements imply degenerate models [18].

Assumption 1 (Integral Causality)**.** We assume all energy storage elements to be in the (preferred) integral causality.

Not all computational directions, or causality, can be assigned to every BG element. Each BG element imposes a set of causal constraints, as shown in Fig. 13, which also shows the constituent equations for each element, based on the causality. A source of effort (respectively, flow) always imposes effort (respectively, flow), and hence has the causal stroke away from (respectively, towards) it. Hence sources of energy have *fixed* causality. Similarly, energy sensors or detectors also have fixed causality, since for an effort (respectively, a flow) detector, its adjacent junction always impose effort (respectively, flow) on it. Resistors relate the effort and flow variables algebraically, and hence can have two possible causal assignments. Under the integral causality assumption, a capacitor (respectively, inductor) always imposes effort (respectively, flow) and have its causal stroke away from (respectively, towards) it. Transformers and gyrators also relate the effort and flow variables algebraically, and can have one of two possible causal assignments.

The causality of bonds incident on a junction is also constrained by the algebraic relations between the effort and flow variables associated with the incident bonds. For example. at a 0-junction, the efforts of all incident bonds are equal. The bond that determines the value of the efforts, and therefore, has its causal stroke towards the 0-junction, is termed the *determining* bond of this 0-junction. The value of the flow of the determining bond of a 0-junction is an algebraic sum of the flows of all other incident bonds. Therefore, the causal strokes for all of the other bonds is directed away from the 0-junction. A complementary procedure can be described for 1-junctions, based on one bond determining the flow at that junction. Note that a junction can have only one determining bond. The *Sequential Causal Assignment Procedure* (`SCAP`) [18] systematically assigns the causality in a BG based on the causal constraints.

The `SCAP` algorithm starts at elements having a fixed, unique causal assignment, i.e., the energy source elements (Se and Sf) and energy storage elements (C and I), and assigns causality. Then the constraints of this causal assignment propagate to adjacent junctions, since the possible options for determining bonds become constrained. If the determining bond for a junction can be assigned

uniquely, the junction is assigned that determining bond, and the constraints imposed by this assignment are propagated along the BG to further restrict the possible options for determining bonds at other junctions. After the causal changes have been propagated from all energy source and storage elements, if some junctions are yet to be assigned a determining bond, `SCAP` arbitrarily chooses an R element connected to one such junction, and assigns causality such that the bond connecting to the R element becomes the adjacent junction's determining bond. The constraints imposed by this assignment are then propagated along the BG, and the process continues until every junction has been assigned a determining bond.

Example. Let us consider how SCAP can be used to assign causality to the bond graph shown in Fig. 12(b). We start with the flow source, F_1, which always imposes flow on the adjacent 0-junction under integral causality. Capacitor C_1 imposes effort on this 0-junction through bond 2, which becomes the determining bond of the 0-junction. Hence, bonds 3 and 4 must impose flow on this 0-junction. Similarly, capacitor C_2 imposes effort on its adjacent 0-junction through bond 7. Thus, bond 7 is this 0-junction's determining bond. As a result, bonds 6 and 8 impose flow on this 0-junction. Next, at the 1-junction, bonds 4 and 6 both impose effort. Hence, the remaining bond 5 must impose flow on this 1-junction, and become its determining bond. The resistor R_{12} conforms to this causal assignment.

Causality has several important advantages. It not only allows us to generate computational forms of BG dynamics, such as state-space equations, signal flow graphs, or block diagrams in an efficient manner [18,89], but also helps in determining other important information about the system from its BG, such as the physical validity of the BG model, and whether the system is observable, i.e., the values of the state variables can be estimated given the measurements. A model that cannot be assigned unique causality to all its bonds according to the causal constraints laid out in Fig. 13 usually indicates a system with algebraic loops. Algebraic loops make the computation of system dynamics harder, and it may result in non-unique diagnoses [90].

Deriving State Space Equations from Bond Graphs

State space equations for the nominal system can be systematically derived from the bond graphs. The number of state variables in a system is equal to the number of C and I elements in integral causality present in the system [18]. For every C element in integral causality, the corresponding state variable is the *displacement* variable, q, such that $\dot{q} = f$. Similarly, for every I element in integral causality, the corresponding state variable is the *momentum* variable, p, such that $\dot{p} = e$.

The displacement and momentum variables of C and I elements in integral causality, respectively, make up the set of all state variables for the system. The state space equations are derived from the constituent equations of each bond graph element, and the constraints imposed by the junctions in the bond graph model. The causal structure of the bond graph model facilitates efficient derivation of the state space equations. Details of this derivation process is presented in [18]. The state space equations of the nominal system is used by the *observer* in TRANSCEND for tracking and fault detection.

Example. The state space equations of the nonlinear two-tank system shown in Fig. 12(b) are given below. The state transition equations of the two-tank system are derived from its bond graph as follows:

$$\begin{aligned}
\dot{q_2} &= f_2 = f_1 - f_3 - f_4 \\
&= F_1 - \frac{1}{C_1 R_1} q_2 - \frac{1}{R_{12}} \left(\frac{q_2}{C_1} - \frac{q_7}{C_2} \right), \\
\dot{q_7} &= f_7 = f_6 - f_8 \\
&= \frac{1}{R_{12}} \left(\frac{q_2}{C_1} - \frac{q_7}{C_2} \right) - \frac{1}{C_2 R_2} q_7, \\
&= \frac{1}{R_{12}} \left(\frac{q_2}{C_1} - \frac{q_7}{C_2} \right) - \frac{1}{C_2 (3q_7^2 / C_2^2)} q_7, \\
&= \frac{1}{R_{12}} \left(\frac{q_2}{C_1} - \frac{q_7}{C_2} \right) - \frac{C_2}{3q_7}.
\end{aligned}$$

Similarly, the observation equations can be obtained from the bond graph as follows:

$$\begin{aligned}
F_1 &= f_9 = \frac{q_2}{C_1 R_1}, \\
F_{12} &= f_5 = \frac{1}{R_{12}} \left(\frac{q_2}{C_1} - \frac{q_7}{C_2} \right), \\
F_2 &= f_{10} = \frac{C_2}{C_2 (3q_7^2 / C_2^2)} = \frac{C_2}{3q_7}.
\end{aligned}$$

In our work, we use the first-order Euler method for deriving the difference equations from these differential equations. In the Euler method, the derivative dx/dt is expressed as

$$\frac{dx}{dt} = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t},$$

where Δt represents the time difference between two consecutive time steps. Hence, for our two-tank system, the state transition equations are:

$$\begin{aligned}
\frac{q_2(t+\Delta t)-q_2(t)}{\Delta t} &= F_1(t) - \frac{1}{C_1 R_1} q_2(t) - \frac{1}{R_{12}}\left(\frac{q_2(t)}{C_1} - \frac{q_7(t)}{C_2}\right), \\
\frac{q_7(t+\Delta t)-q_7(t)}{\Delta t} &= \frac{1}{R_{12}}\left(\frac{q_2(t)}{C_1} - \frac{q_7(t)}{C_2}\right) - \frac{C_2}{3q_7}(t),
\end{aligned} \tag{13}$$

and the observation equations are,

$$\begin{aligned}
F_1(t) &= \frac{q_2(t)}{C_1 R_1} \\
F_{12}(t) &= \frac{1}{R_{12}}\left(\frac{q_2(t)}{C_1} - \frac{q_7(t)}{C_2}\right) \\
F_2(t) &= \frac{C_2}{3q_7(t)}.
\end{aligned} \tag{14}$$

Modeling Abrupt Faults

Definition 3 (Fault)**.** A *fault*, ϕ, is an unexpected change in the plant or its instrumentation that causes the system to deviate from its nominal behavior.

We define a set of faults, F, such that $F = \{\phi_1, \phi_2, \ldots, \phi_l\}$. A set is denoted by uppercase symbols, and its elements are denoted by lowercase symbols. The original version of TRANSCEND focused on the diagnosis of abrupt faults. The mathematical model for an abrupt fault is defined below, and its profile is illustrated in Fig. 14. In general, TRANSCEND can diagnose additive and parametric abrupt faults in plants, actuators, and sensors. However, in this dissertation, we focus on parametric plant faults only, which are hard to analyze because they directly affect the system dynamics. Parametric plant forms are modeled as changes in system parameters. From now on, the term "fault" will imply a "parametric plant fault". Bond graphs support component-oriented parametrized modeling of physical systems. Hence, every fault maps to a bond graph element, p, and an *abrupt* fault is denoted by $p^{\pm a}$, such that p^{+a} denotes an abrupt increase in the parameter p, while p^{-a} denotes an abrupt decrease in parameter p.

Definition 4. (Abrupt fault) An *abrupt fault* is characterized by a fast change (i.e., the rate of change is much faster than the dynamics of the system) in the system parameter, p (with nominal parameter value function, $p(t)$), and hence modeled as a constant persistent bias term, $\Delta_p^a \times p(t)$, where Δ_p^a is the percentage change in the parameter expressed as a fraction, added to the nominal

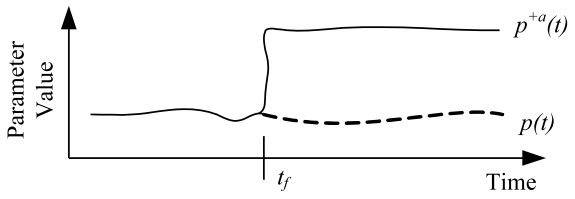


Figure 14: Abrupt fault profile.

parameter value, $p(t)$, i.e.,

$$p^{\pm a}(t) = \begin{cases} p(t) & t < t_f \\ p(t) \pm \Delta_p^a \times p(t) & t \geq t_f, \end{cases}$$

where t_f is the time of fault occurrence, and $p^{\pm a}(t)$ is the temporal profile of parameter p with an abrupt fault.

Example. Consider the two tank system shown in Fig. 12(b). In this two-tank system, the set of possible abrupt faults is $F = \{C_1^{-a}, C_2^{-a}, R_1^{+a}, R_2^{+a}, R_{12}^{+a}\}$. This includes changes in tank capacities, drain pipe resistances and connecting pipe resistances. A + (−) superscript implies that the fault occurrence causes an abrupt increase (decrease) in the corresponding parameter value. For example, C_1^{-a} indicates an abrupt decrease in tank 1 capacity, for example, if a stone is dropped into tank 1, thereby abruptly decreasing the capacity of the fault. Similarly, a block in the drain pipe of tank 1 is represented by an abrupt increase in the pipe resistance, R_1^{+a}.

Temporal Causal Graphs

The TRANSCEND fault isolation scheme uses a model that explicitly incorporates the cause and effect relationships between the abrupt faults mentioned above, and the measurements. This diagnosis model is termed a *temporal causal graph* (TCG). TCGs contain the causal information that allows the deviations of measurements from nominal to be mapped on to possible parameter deviations, and also predict qualitatively the effect each of the parameter deviations would have on the measurements.

The TCG is essentially a signal flow graph whose vertices correspond to the effort and flow variables of the bond graph, and the edges denote causal dependencies between these system variables. A TCG captures both the causal and temporal relations between its nodes. Each edge of a TCG has a qualitative label, which denotes how the source node of the edge affects the destination node. The possible edge labels include $\{=, +1, -1, \frac{1}{\theta}dt, \theta, \frac{1}{\theta}\}$, where θ denotes a system parameter, $=$ denotes equality, $+1$ denotes direct proportionality, -1 denotes inverse proportionality, and dt denotes a

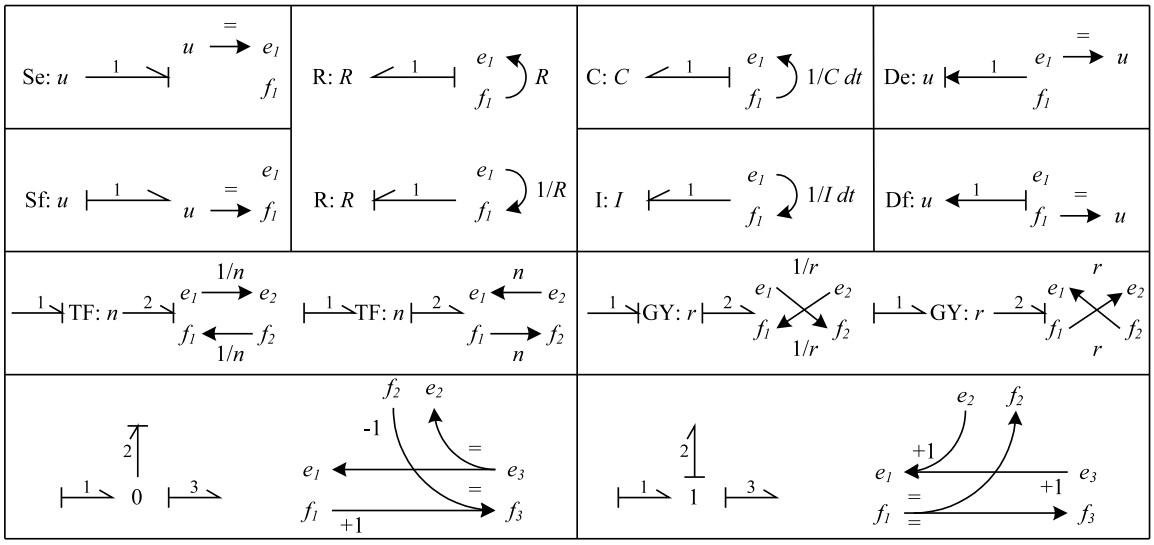


Figure 15: Generating temporal causal graphs from bond graphs.

time delay, i.e., integration. While the labels involving θ are defined by the constituent equations of each BG element based on causality, the $=$, $+1$, and -1 labels are obtained from the causal constraints and direction of bonds at the junctions, as described below.

Since the topological structure of the bond graph and the properties of its constituent elements imply inherent causal relations between system variables, TCGs can be systematically derived from the system bond graphs [7]. System components are directly mapped to bond graph element parameters, and this provides the mapping from observed behavior deviations to parameter value changes in the system components [7,8].

Deriving Temporal Causal Graphs from Bond Graphs

Given a bond graph, we first assign causality to it using `SCAP`. Then, for every bond i, we instantiate vertices corresponding to the bonds's effort, e_i and, flow f_i variables. Also, the bond graph elements are converted into directed labeled edges, where the labels include system parameters, direct proportionality, inverse proportionality, or equality relations. The direction of the edges are determined based on the assigned causality. The edge labels are based on the assigned causality, and the direction of the bonds.

For energy storage elements under the preferred integral causality assumption, e.g., a capacitor, C, having the constituent equations $\dot{e_i} = \frac{1}{C}\int f_i dt$, the TCG shows $f_i \xrightarrow{dt/C} e_i$, with dt denoting a temporal relationship, or delay, between f_i and e_i. On the other hand, a resistor R could correspond to either an $e_i \xrightarrow{1/R} f_i$ or $f_i \xrightarrow{R} e_i$ edge in a corresponding TCG. At a junction, depending on the

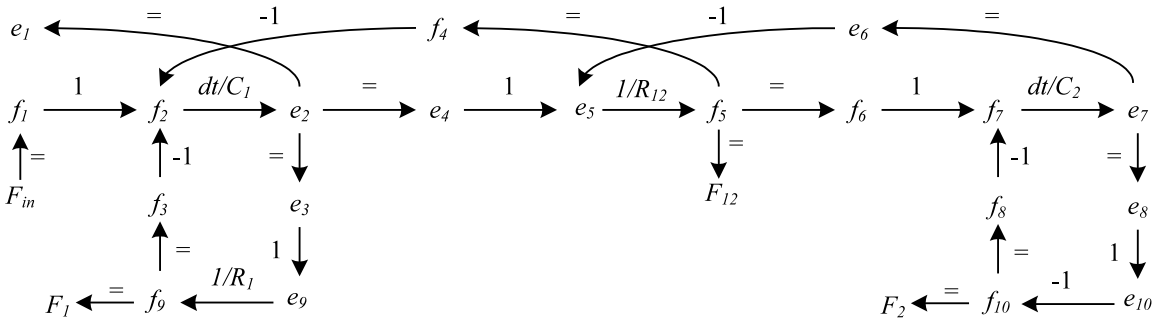


Figure 16: Temporal causal graph of the two-tank system.

junction type, one type of variables is summed, and the other type of variables are set to be equal. The determining bond of the junction sets the value. Say we have a 0-junction with three bonds i, j, and k, with bond i directed toward the junction, and the other two bonds directed away. If bond k is the determining bond for this 0-junction, we have the equations $e_i = e_j = e_k$ and $f_k = f_i - f_j$. Hence we have edges $e_k \xrightarrow{=} e_j$, $f_j \xrightarrow{-1} e_k$, and so on.

Additional sensitivity analysis is required to determine the labels on TCG edges involving modulated parameters. Consider the scenario where a parameter p relates the energy and flow variables, e_p and f_p, respectively, as $e_p = pf_p$. If the parameter p is modulated, and is a function of a set of TCG variables (or nodes), $\mathbf{N}$, i.e., if $p = g(\mathbf{N})$, then the TCG has an edge from every $N \in \mathbf{N}$ to f_p, with **sign** $\left(\frac{\partial f_p}{\partial N}\right)$ as the edge label for this link. If $\frac{\partial f_p}{\partial N} = 0$, no TCG link is drawn from N to f_p.

Example. Fig. 16 shows the TCG of the two-tank system, whose bond graph is shown in Fig. 12(b). The nodes e_i and f_i in the TCG correspond to effort and flow variables of bond i. For example, e_1 and f_1 correspond to bond 1. As explained above, under integral causality, C_1 imposes effort on its adjacent 0-junction, and hence the edge $f_2 \xrightarrow{dt/C_1} e_2$ is drawn in the TCG. The label dt represents integration. Similarly, the edge $f_7 \xrightarrow{dt/C_2} e_7$ is drawn for capacitor C_2. Resistor R_1 relates e_3 and f_3 according to the relation $f_3 = R_1 e_3$. Hence, we create edge $e_3 \xrightarrow{1/R_1} f_3$ in the TCG. Similarly we create edge $e_5 \xrightarrow{1/R_{12}} f_5$ corresponding to resistor R_{12}. To determine the label for $e_{10} \rightarrow f_{10}$ edge, related by nonlinear resistance $R_2 = 3e_7^2 = 3e_{10}^2$, we compute $\frac{\partial f_{10}}{\partial e_{10}}$, where $f_{10} = \frac{e_{10}}{R_2} = \frac{e_{10}}{3e_{10}^2}$. Hence, $\frac{\partial f_{10}}{\partial e_{10}} = -\frac{1}{3e_{10}^2}$. As the partial derivative can never be positive, it can be determined that f_{10} is inversely proportional to e_{10}, and hence the $e_{10} \rightarrow f_{10}$ edge is assigned a '-1'label. At the 1-junction, bond 5 is the determining bond. Hence, we have $f_4 = f_6 = f_5$ and, hence we have edges $f_5 \xrightarrow{=} f_4$ and $f_5 \xrightarrow{=} f_6$. At the 1-junction, we also have $e_5 = e_4 - e_6$. Hence the edges $e_4 \xrightarrow{+1} e_5$ and $e_6 \xrightarrow{-1} e_5$ have labels $+1$ and -1, respectively. In addition, we also have TCG edges relating

effort and flow variables to sources and sensors. These edge labels are usually '=', representing equality. For example, the signal links connected to the flow sensors correspond to the $f_9 \xrightarrow{=} F_1$, $f_5 \xrightarrow{=} F_{12}$, and $f_{10} \xrightarrow{=} F_2$ TCG edges. The input flow source affects the energy variables through the $F_{in} \xrightarrow{=} f_1$ TCG edge.

Tracking and Fault Detection

Tracking

In our diagnosis scheme, the first step to fault detection is the generation of measurement residuals. TRANSCEND adopts the usual definition of a residual, $r(t)$, of measurement, $y(t)$, and defines it as $r(t) = y(t) - \hat{y}(t)$, where $\hat{y}(t)$ is the estimated value of $y(t)$ given by the observer at time t. In TRANSCEND, an extended Kalman filter [69] is used as an observer to estimate the state variables for nonlinear systems, assuming that all of the state variables and measurements are stochastic and are sampled from given Gaussian Normal distributions, and the covariance matrices for the state and measurement equations are known.

Fault Detection

In an ideal scenario, a non-zero residual $r(t) \neq 0$ implies a fault is detected, while a zero residual implies nominal conditions. However, real-world scenarios have to accommodate uncertainties due to measurement noise and modeling errors. Therefore, to avoid false alarms, but retain the sensitivity of detection (i.e., to avoid missed alarms), we employ a statistical Z-test to establish if measurement residuals are statistically significant [91]. A statistically significant non-zero residual implies significant deviation of system behavior from the expected nominal behavior that can be attributed to the occurrence of faults in the system. The Z-test uses a sliding window scheme to compute the residual mean and nominal signal variance. The choice of parameters for this scheme and the confidence level chosen for the Z-test determines the properties of the fault detection filter. These parameters also determine the tradeoff between false alarms and fast detection of faults.

The signal deviation at time step t is defined in terms of an average residual for the last N_2 samples, i.e.,

$$\hat{\mu}_{N_2}(t) = \frac{1}{N_2} \sum_{i=t-N_2+1}^{t} r(i).$$

A hypothesis testing scheme based on the Z-test is employed to establish the significance of the deviation. To perform the Z-test, the variance of the measurement residual must be known. For

unknown variance the t-test may be performed. To approximate the conditions necessary for the Z-test, the variance of the signal is estimated, but from a larger set of N_1 samples, i.e., $N_1 \gg N_2$:

$$\hat{\sigma}^2_{N_1}(t) = \frac{1}{N_1 - 1} \sum_{i=t-N_1+1}^{t} \Big(r(i) - \mu_{N_1}(t)\Big)^2.$$

Care must be taken to ensure that the σ calculation is performed on the nominal signal, i.e., the N_1 window should lag the N_2 by a sufficiently large number of samples, so that the occurrence of the fault should not throw off the calculation of nominal variance.

The $Z-$value has a distribution $N(0,1)$:

$$Z(t) = \frac{\hat{\mu}(t)}{\sigma(t)/\sqrt{N_2}}. \tag{15}$$

The confidence level, defined by α, defines the bound $[z_-, z_+]$: $P(z_- < z < z_+) = 1-\alpha$. This bound can be transformed to another bound $[\mu_-, \mu_+]$ using Eqn. (15), and the approximation $\sigma = \hat{\sigma}_{N_1}(t)$:

$$\mu_- = z_- \frac{\sigma}{\sqrt{N_2}}, \quad \mu_+ = z_+ \frac{\sigma}{\sqrt{N_2}}.$$

The Z-test is employed in the following manner:

$$\begin{aligned} \mu_- \leq \mu(t) \leq \mu_+ \quad &\Rightarrow \quad no\ fault \\ otherwise \quad &\Rightarrow \quad fault. \end{aligned}$$

The advantage of this fault detection approach is that it is computationally simple, and it makes no assumptions concerning the properties of the changed mean value (e.g., it does not have to be constant).

Qualitative Fault Isolation

Once a fault is detected, the qualitative fault isolation (Qual-FI) scheme is triggered to generate the initial fault hypotheses and refine these hypotheses as additional measurement deviations are observed. Our Qual-FI scheme is based on analyzing the transients in the measurements caused by faults, and comparing the expected deviation of measurements from nominal with the actual observed deviations, represented qualitatively using symbols. Specifically, Qual-FI consists of three main steps: (*i*) symbol generation, (*ii*) hypothesis generation, and (*iii*) hypothesis refinement. The

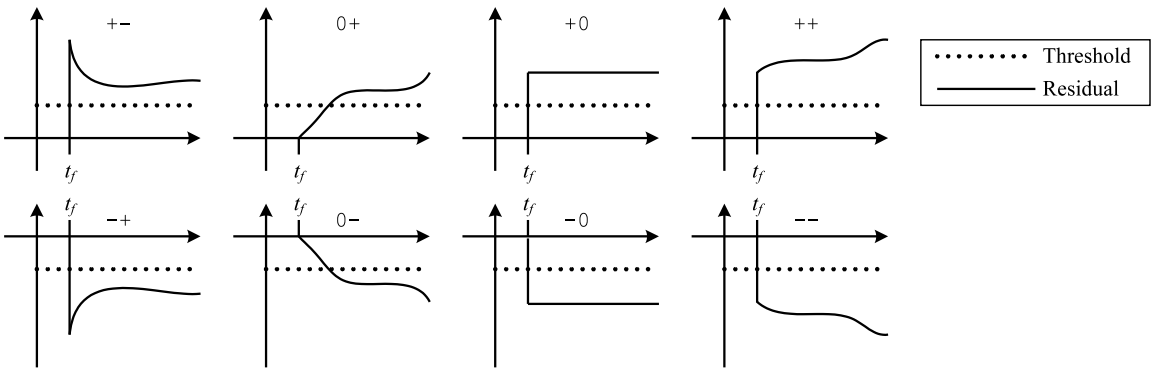


Figure 17: Symbolic abstractions of measurement deviations (adapted from [1]).

hypothesis refinement step consists of fault signature generation and progressive monitoring steps. We present these different steps of Qual-FI in detail below. Note that once a fault is detected, we stop the observer, and use the system model to simulate the system. If the observer is not stopped, fault behaviors may be compensated for, resulting in incorrect or no symbols to be generated.

Symbol Generation

To facilitate qualitative isolation, two features are extracted from each measurement residual to denote how the residual changes over time. The two features are abstracted symbolically as $-$, 0, and $+$ *symbols*, representing above, at or below nominal values, respectively. These two features together form an ordered pair of symbols. The first feature captures the change in the residual magnitude, while the second feature represents the first-order change in the residual [8]. A non-zero first symbol implies the occurrence of a discontinuity, as well as, whether the measurement discontinuously increased or decreased from nominal. Fig. 17 illustrates the possible measurement symbols that can be generated.

The sign of the mean $\mu(t)$ obtained from the Z-test, discussed earlier, gives the first element of the ordered pair, i.e., the symbol for the deviation in the magnitude of the residual. The computation of the symbol for the slope involves some more computation. Let the deviation in the residual be detected at time k_0. The approximate variance of the residual is $\hat{\sigma}_r^2 = \hat{\sigma}_{N_1}^2 (k_0 - N_2)$. It is assumed that the variance does not change due to the fault. A delayed value is used to prevent distortion of the variance estimate. In a noise-free environment, the simple difference $r_0(k_0 + 1) - r(k_0)$ would generate the required slope symbol. However this is rarely the case. Therefore, in a noisy environment, a threshold can be used to prevent bad symbol generation. This threshold depends on the noise. Similar to the generation of the magnitude symbol, a mean value will be used to make

the decision. The size of the window used to calculate the mean is increased until the symbol is successfully generated.

Assume μ_{r_0} is the estimate of the 'initial' residual value after fault detection, and is calculated as the average of the initial N_3 residual samples, i.e.,:

$$\mu_{r_0} = \frac{1}{N_3} \sum_{j=0}^{N_3-1} r(k_0 + j).$$

N_3 is a design parameter. The mean value of the differential residual after fault detection is defined as:

$$\mu_d(k_0 + k) = \begin{cases} \frac{1}{k} \sum_{j=1}^{k} \left(r(k_0 + j) - \mu_{r_0} \right) = \left(\frac{1}{k} \sum_{j=1}^{k} r(k_0 + j) \right) - \mu_{r_0}, & k > N_3 \\ 0, & k \leq N_3 \end{cases}.$$

The variance of μ_d is $\hat{\sigma}_d^2(k_0 + k) \approx \hat{\sigma}_r^2/k$, while the variance of μ_{r_0} is $\hat{\sigma}_{r_0}^2 \approx \hat{\sigma}_r^2/N_3$. That is, the uncertainty of the initial residual value depends on the noise and N_3, while the uncertainty of the mean estimate depends on the noise and the number of samples used in the calculations. Using a confidence value α and the corresponding z_+ and z_- values, just like we did for the Z-test, the $+$ slope symbol is generated when:

$$\mu_d - z_+\sigma_d > \frac{z_+\sigma_r}{\sqrt{N_3}} \Rightarrow \mu_d > z_+\sigma_r \left(\frac{1}{\sqrt{N_3}} + \frac{1}{\sqrt{(k)}} \right).$$

Similarly, the $-$ slope symbol is generated when

$$\mu_d < -z_+\sigma_r \left(\frac{1}{\sqrt{N_3}} + \frac{1}{\sqrt{(k)}} \right).$$

Hypothesis Generation

Hypothesis generation is the first step in the fault isolation scheme. Once the fault detector reports an deviant observation, a *backward propagation* [7] scheme applied to the TCG identifies the set of component parameters with a hypothesized direction of change in the parameter that explains the first measurement deviation observed. These implicated component parameters constitute the initial *fault hypotheses*.

Definition 5. (Fault Hypothesis) *Fault hypotheses* are possible faults in component parameters that explain all of the measurement deviations observed thus far.

Example. For example, in the two-tank system, if the height of tank 1 increases gradually, there can be two possible explanations for this fault, namely the increase in resistance R_{12}, i.e., R_{12}^{+a}, or

an abrupt decrease in the capacity of tank 2, i.e., C_2^{-a}. Hence R_{12}^{+a} and C_2^{-a} are two possible fault hypotheses. However, if at a later stage, the height of tank 2 is observed to increase gradually, then C_2^{-a} is not considered a fault hypothesis anymore since it is not consistent with the gradual increase in the height of tank 2. A C_2^{-a} fault would cause the tank 2 height to increase abruptly instead.

Since we are dealing with dynamic systems, measurements can deviate over time, and not all at once. So, at any time instance, the fault hypotheses are consistent explanations of the deviations in measurements observed up to the current time step. As more measurement deviations are observed and fault hypotheses that are not consistent explanations of the observed deviations are dropped from consideration, thereby refining the fault hypothesis set.

Hypothesis Refinement

Generating Fault Signatures

After the initial fault hypotheses are generated, propagating in the forward direction along the temporal causal graph generates *fault signatures*, i.e., symbolic representations of the possible effects of the hypothesized faults on observable measurements at the point of failure [7]. The transients produced by abrupt faults can only have discontinuities at the time point of failure. For all other times, the system behavior is continuous and continuously differentiable, and the transient response to a fault can be approximated by its Taylor series expansion:

$$y(t) = y(t_f) + y'(t_f)\frac{(t-t_f)}{1!} + y''(t_f)\frac{(t-t_f)^2}{2!} + \ldots + y^{(k)}(t_f)\frac{(t-t_f)^k}{k!} + \ldots, \tag{16}$$

where t_f is the time point of fault occurrence, and $t > t_f$.

If $|y^{(k+1)}|$ is bounded and t is close to t_f, the Taylor series is a good approximation of the true signal $y(t)$. The time-varying residual signal, $r(t) = y(t) - \hat{y}(t_f)$, where $\hat{y}(t_f)$ is the predicted measurement value at time point t_f, is computed as

$$r(t) = y(t_f) - \hat{y}(t_f) + y'(t_f)\frac{(t-t_f)}{1!} + y''(t_f)\frac{(t-t_f)^2}{2!} + \ldots + y^{(k)}(t_f)\frac{(t-t_f)^k}{k!} + \ldots,$$

i.e., the magnitude difference, $y(t_f) - \hat{y}(t_f)$, and k derivative values $(y'(t_f), y''(t_f), \ldots, y^{(k)}(t_f))$. After a fault occurs, the nominal system model cannot be used to calculate the numeric values of the derivatives. Instead, we use the TCG model to express the fault residual as qualitative magnitude and derivative changes [7,8]. This becomes the basis for establishing a *qualitative fault signature* for a fault transient [8].

Table 2: Fault signatures from abrupt faults in the two-tank system

Fault	F_1	F_2
C_1^{-a}	$+-$	$0+$
R_{12}^{+a}	$0+$	$0+$
C_2^{-a}	$0+$	$+-$

Definition 6 (Qualitative Fault Signature)**.** Given a fault ϕ, and measurement m, a *qualitative fault signature*, $FS(\phi, m)$, of order k, is an ordered $(k+1)$-tuple consisting of the predicted magnitude and 1^{st} through k^{th} order time-derivative effects of a residual signal of measurement m, at the point of failure of fault f, expressed as qualitative values: below normal $(-)$, normal (0), and above normal $(+)$. Typically, k is chosen to be the order of the system.

Example. Table 2 shows some fault signatures[3] of the nonlinear two-tank system. The signature, $(+-)$ of fault C_1^{-a} for measurement, F_1, implies that an abrupt decrease in capacity of the tank 1 will cause a discontinuous increase in the outflow from that tank at the time of fault occurrence. Similarly, the fault signature, $(0+)$, of the same fault for measurement F_2 implies that a C_1^{-a} fault would result in the flow out of tank 2 to increase gradually.

After fault detection, online fault isolation compares the magnitude and slope of measurement residual signals to derived fault signatures. Computing higher order derivatives from noisy measurement signals is unreliable [92]. For this measurement scheme, we have shown that all of the discriminatory evidence for fault isolation is provided by the first change in residual magnitude from the point of failure detection [8]. This reduces the possible fault signatures for a measurement to the set of symbols, $\Sigma = \{(+,-),\ (-,+),\ (0,+),\ (0,-)\}$. The first two fault signatures correspond to a discontinuous change in a signal while the last two signatures imply that at the point of failure, no discontinuous jump in the measurement residual will be observed. Fault signatures, $(+,+)$ and $(-,-)$, are not considered because they imply positive feedback loops, and hence, unstable systems.

Progressive Monitoring

After the fault signatures are generated, as additional measurements deviate from nominal, a progressive monitoring scheme compares the generated symbol deviations for these measurements to the fault signatures of the generated fault hypotheses, and if any fault signature is inconsistent with the observed symbol for that measurement, the fault hypothesis is dropped. Specifically, a fault hypothesis is deemed inconsistent if the observed deviation in a residual is different from the pre-

[3] In the remainder of this dissertation, we refer to "qualitative fault signatures" simply as "fault signatures".

dicted effect of that fault hypothesis on that measurement, as represented by its fault signature for that particular measurement. This process is continued till the fault hypothesis set is refined to a singleton set, or till the fault hypothesis cannot be refined any further. In recent work, the discriminatory power of the measurements have been improved using the concept of *relative measurement orderings*, taking into account the partial order in which the measurements deviate [93]. However, the use of relative measurement orderings is beyond the scope of this dissertation.

Fault Identification

Fault identification, i.e., parameter estimation is run as an iterative nonlinear estimation process. If multiple single fault hypotheses are valid at the end of the hypothesis refinement step explained earlier, then the fault identification step also helps in isolating the unique single fault. For each fault parameter, an estimation scheme is run that updates the parameter value in a way that minimizes the least squares error between the actual and predicted observation [91]. The parameter estimator that converges to the smallest least squares error is declared to be the true candidate. Each bond graph element, P, is defined in terms of an associated parametrized *degradation function*, $Degradation(P_{fault})$ with one input argument, the *degradation parameter*, P_{fault}. For example, an abrupt fault in resistance, R, can be defined as $R = Degradation(R_{fault}) = R_{fault}$. Therefore, the degradation function defines the change in the characteristics of a bond graph element due to a fault.

Fig. 18 shows the details of the fault identification procedure. Once the fault is detected at time t_f, the current state variables of the system, $X(t_f)$, as well as the following N samples of the system input signal, $U(t_f, t_f+1, \ldots, t_f+N-1)$, and the system output signal, $Y(t_f, t_f+1, \ldots, t_f+N-1)$, are stored. The *system model* consists of the state-space equations. In addition to the *measurements*, X, U and Y, and the system model, there is a *simulator* that is capable of simulating the system from an arbitrary initial state and generating the estimated output $\hat{Y}$. Note that the simulator can also be parameterized, thereby allowing the modification of the degradation parameter for any bond graph element through the parameter index I and the corresponding degradation parameter P_{fault}. Once the estimates $\hat{Y}$ are obtained, the least squares error between Y and $\hat{Y}$ are calculated by the *Least Squares Error Calculator*. Finally, the *iteration engine* uses iterative optimization techniques to estimate the value of P_{fault}, which corresponds to the (possible global) minimum of an error surface.

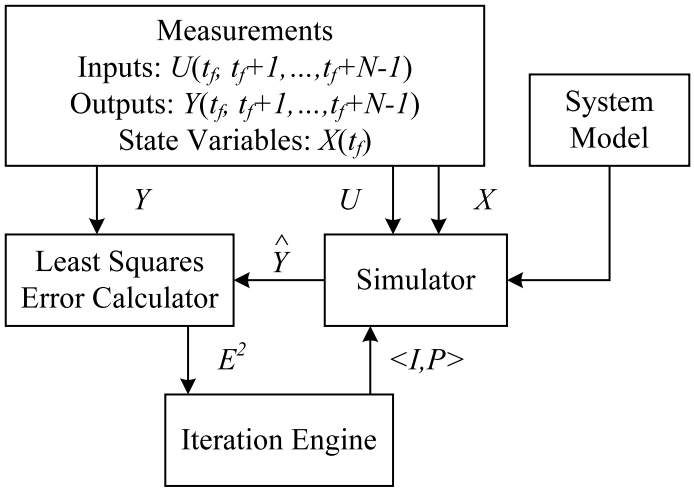


Figure 18: Block diagram of the fault identification procedure.

The choice of the iteration engine poses a trade-off. The more general the optimization algorithm, the slower is the operation. However, this performance can be improved by building in *a priori* domain knowledge into the engine. The properties of the error surfaces to be optimized are good criteria for selecting the engine. The current iteration engine assumes that the error surface is well behaved and assumes a parabolic form.

The minimum of the error surface, $E^2(P_{fault})$, can be determined by scanning the possible parameter range and determining the minimum value of E^2. This calculation of each point $E^2(P_{fault})$ on the error surface involves a run of the simulator with parameter P_{fault}. Since each run of the simulator to determine a particular error value involves a lot of calculations, the number of simulator runs must be kept at a minimum in order to be able to run this algorithm online. One way of making the number of simulator runs practically feasible is to find the minimum of the error surface by calculating a small number of points on it. The assumption of error surface lying close to a parabola (which is valid for points close to the minimum, but not for the whole surface) allows for performing the optimization by a series of parabolic fits, with a relatively small number of simulator runs.

Extending TRANSCEND To Include Diagnosis of Incipient Faults

A large class of realistic faults occur because of degradation and wear and tear due to system use. In these situations, unlike abrupt faults, the associated component parameter changes gradually, and such faults are termed *incipient*. As part of this research, we have extended TRANSCEND's hypothesis generation and fault signature derivation scheme to encompass incipient faults. We have formulated these extensions in such a way that these extensions are transparent to the core

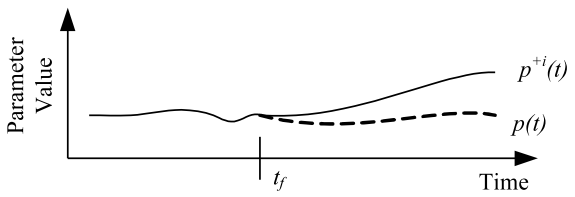


Figure 19: Incipient fault profile.

progressive monitoring step of TRANSCEND . We first define incipient faults, and then, present in detail the changes we have included in the hypothesis generation and fault signature derivation steps to accommodate incipient fault diagnosis.

Modeling Incipient Faults

The mathematical model for incipient faults used in our approach is defined below, and its fault profile is illustrated in Fig. 19. An incipient fault in component parameter, p, in a bond graph model is denoted by $p^{\pm i}$, such that p^{+i} denotes an incipient increase in the parameter p, and p^{-i} denotes an incipient decrease in parameter p.

Definition 7. (Incipient fault) An *incipient fault* is a slow change (i.e., the rate of change is comparable to, or slower than, the dynamics of the system) in a system parameter, p (with nominal parameter value function, $p(t)$). Without loss of generality, we approximate this gradual change as a linear function[4] with a constant slope, Δ_p^i, added to the nominal component parameter value function, $p(t)$, i.e.,

$$p^{\pm i}(t) = \begin{cases} p(t) & t < t_f \\ p(t) \pm \Delta_p^i \times (t - t_f) & t \geq t_f, \end{cases} \tag{17}$$

where t_f is the time of fault occurrence, and $p^{\pm i}(t)$ is the temporal profile of parameter p with an incipient fault.

Example. Consider the two tank system shown in Fig. 12(b). A common incipient fault is a gradual blocking in a pipe due to accumulation of sediments. Such a fault can be denoted as R_{12}^{+i}.

Extending Hypothesis Generation

The first step to diagnosing incipient faults is the generation of incipient fault candidates. This incipient fault hypothesis generation is based on the deviation in the measurement residual that resulted in the detection of a fault. Since, an incipient fault is a first order change in a component

[4] A more accurate approximation may be a piecewise linear function.

parameter, it can only have a first or higher order effect on the measurements. Hence, incipient fault candidates are generated much like the abrupt fault candidates, except when a measurement deviation shows a discontinuity. This also follows from our Taylor series expansion of the measurement residual (Equation 16), which indicates that measurement discontinuity can only occur for abrupt faults.

Example. Consider the two-tank system shown in Fig. 12(b). Recall how a gradual increase in the height of tank 1 resulted in the implication of R_{12}^{+a} and C_2^{-a} as possible fault hypotheses. Under the extended hypothesis generation scheme, we also generate two additional fault hypotheses, R_{12}^{+i} and C_2^{-i}, corresponding to the abrupt fault candidates that were generated. Note, however, that if the fault was detected because of a discontinuous change in a measurement, no additional incipient fault hypotheses would have been generated.

Extending Fault Signature Derivation

As shown above, since an incipient fault is modeled as a first order change in a parameter, the effect an incipient fault parameter, p^{+i}, has on a system variable, e_1, is equivalent to the effect an abrupt change in that parameter would have on the same variable, e_1, but after passing through a delay element. Graphically, this delay in fault effect can be expressed through the TCG fragments shown below:

$$f_1 \xrightarrow{R^{+i}} e_1 \quad = \quad f_1 \xrightarrow{\Delta_R^i dt + R} e_1$$

$$e_1 \xrightarrow{\frac{1}{R^{+i}}} f_1 \quad = \quad e_1 \xrightarrow{\frac{1}{\Delta_R^i dt + R}} f_1$$

$$f_1 \xrightarrow{\frac{dt}{C^{+i}}} e_1 \quad = \quad f_1 \xrightarrow{\frac{dt}{\Delta_C^i dt + C}} e_1$$

$$e_1 \xrightarrow{\frac{dt}{I^{+i}}} f_1 \quad = \quad e_1 \xrightarrow{\frac{dt}{\Delta_I^i dt + I}} f_1$$

In each situation, an incipient fault edge label can be expanded to include an additional delay element, represented by dt, compared to an abrupt change. As a result, incipient faults cannot produce discontinuities in measurements, and their fault signatures are always of the form (0τ), where τ is the first non-zero symbol in the fault signature of an abrupt fault in the same system parameter and for the same direction of change.

Example. For example, consider the fault signature matrix shown in Table 3. The signature of fault R_1^{+i} for flow F_1, $FS(R_1^{+i}, F_1) = (0-)$, since that of the abrupt fault R_1^{+a}, $FS(R_1^{+a}, F_1) = (-+)$.

Table 3: Selected fault signatures for the two-tank system

Fault	F_1	F_{12}	F_2
R_1^{+a}	$-+$	$0+$	$0+$
R_1^{+i}	$0-$	$0+$	$0+$
R_{12}^{+a}	$0+$	$-+$	$0-$
R_{12}^{+i}	$0+$	$0-$	$0-$
R_2^{+a}	$0+$	$0-$	$-+$
R_2^{+i}	$0+$	$0-$	$0-$

Similary, $FS(R_{12}^{+i}, F_{12}) = (0-)$, since $FS(R_{12}^{+a}, F_{12}) = (-+)$. However, $FS(R_2^{+i}, F_1) = (0+)$ is the same as that of $FS(R_2^{+a}, F_1)$, since both R_2^{+i} and R_2^{+a} results in a gradual increase in flow F_1.

Once the fault signatures are derived for incipient fault hypotheses, the progressive monitoring scheme described earlier can be applied to isolate both incipient and abrupt faults within an unified qualitative diagnosis framework.

Diagnosability Analysis

The use of fault signatures for fault isolation may not always result in unique isolation of the true fault, since many faults may have the same fault signatures for all measurements in the system. *Diagnosability* is a measure of how well faults in a system can be uniquely discriminated by a diagnosis framework.

Our analysis of a system diagnosability in the TRANSCEND diagnosis framework is based on the *fault signature matrix*, as we show below. Given the set of possible faults, $F = \{\phi_1, \ldots, \phi_l\}$, and the set of measurements, $M = \{m_1, \ldots, m_n\}$, the *fault signature matrix*, $FSM_{(F,\ M)} = [FS(\phi_i, m_j)]_{l \times n}$, is a $l \times n$ matrix with rows corresponding to faults and columns corresponding to measurements, and $FS(\phi_i, m_j)$, the fault signature of fault ϕ_i for measurement m_j, as its elements. A *fault signature tuple*, $\langle FS(\phi_i, \widehat{M})\rangle$, defined for fault ϕ_i and a measurement set $\widehat{M} = \{m_1, m_2, \ldots, m_k\} \subseteq M$, can be extracted from row i of the $FSM_{(F,\ M)}$ by selecting only those elements that are in the columns corresponding to the measurements in $\widehat{M}$. Formally, $\langle FS(\phi_i, \widehat{M})\rangle = \langle FS(\phi_i, m_1),\ FS(\phi_i, m_2),\ \ldots,\ FS(\phi_i, m_k)\rangle$.

We define a system as $S = (F, M, FSM_{(F,\ M)})$, where F is the set of possible faults, M is the set of measurements, and $FSM_{(F,\ M)}$ is the fault signature matrix for F and M. The system S is said to be *diagnosable* if every fault $f_i \in F$ can be uniquely isolated using the measurements in M. Formally, in the TRANSCEND diagnosis approach, *diagnosability* is defined as follows.

Definition 8 (Diagnosability)**.** Given the set of available measurements, M, and the set of faults, F, a system is *diagnosable* in the TRANSCEND diagnosis framework if all single faults in F can be uniquely isolated using measurements in M, i.e., $\forall(f_h, f_g \in F, f_h \neq f_g),\ \langle FS(f_h, M)\rangle \neq \langle FS(f_g, M)\rangle$.

Example. Consider the two-tank system in Fig. 12(b). Assume that the faults of interest are $F = \{C_1^{-a}, C_2^{-a}, R_{12}^{+a}\}$ and $M = \{F_1, F_2\}$. From Table 2 we see that measurement F_1 can discriminate between faults C_1^{-a} and R_{12}^{+a}, but not C_2^{-a} and R_{12}^{+a}. Similarly, F_2 can discriminate between $C_2^{-a}$and R_{12}^{+a}, but not C_1^{-a} and R_{12}^{+a}. However, together, F_1 and F_2 can uniquely isolate all single faults in F. Therefore, a system with faults F is diagnosable using M. If F_2 was the only measured variable, i.e., $M = \{F_2\}$, then faults C_1^{-a} and R_{12}^{+a} cannot be uniquely isolated, and hence, F is not diagnosable using M in this scenario.

The discriminatory power of signatures is the basis for constructing measurement selection algorithms [94] for finding the minimum number of measurements that establish complete diagnosability for a given set of faults. In Chapter IV, we analyze the diagnosability property of systems to design qualitative distributed diagnosers that generate globally correct diagnosis results locally, without a centralized coordinator. Our extensions to the hypothesis generation and fault signature derivation procedures facilitate the qualitative diagnosis of both incipient and abrupt faults. However, the generation of the incipient fault hypotheses, many of which have the same fault signatures as their abrupt counterparts results in a loss of discriminatory power of the qualitative isolation scheme. Hence, the qualitative isolation scheme alone may not be able to uniquely discriminate between incipient and abrupt fault hypotheses. In Chapter V, we combine the qualitative TRANSCEND diagnosis scheme with Bayesian inference to improve the diagnosability of our qualitative diagnosis approach.

Summary

In this chapter, we presented in detail, the TRANSCEND qualitative diagnosis framework, on which our current research builds upon. We first reviewed how TRANSCEND performed the tasks of fault detection, isolation, and identification of abrupt faults. We then presented how we extend TRANSCEND to include the generation of incipient fault hypotheses, and the derivation of fault signatures for these incipient fault hypotheses, based on the fault signatures of their corresponding abrupt fault hypotheses. These extensions facilitate the diagnosis of incipient and abrupt faults within a unified qualitative framework. We also presented the approach to analyze the diagnosability properties of a system in the TRANSCEND diagnosis framework. In the next chapter, we present how the

diagnosability properties of a system is analyzed to partition the system into multiple subsystems, such that when the TRANSCEND diagnosis approach is applied to each subsystem independently, the distributed TRANSCEND diagnosers generate globally correct diagnosis results through local analysis, without a centralized coordinator, with no exchange of partial diagnosis results amongst the diagnosers, and with minimal sharing of measurements.

The TRANSCEND approach presented in this chapter is a centralized diagnosis approach, and like other centralized approaches, it suffers from scalability, computational complexity, and robustness issues. Our approach to alleviate these issues by distributing the diagnosis task is presented in the next chapter. Through careful design of our distributed diagnosers, we guarantee globally correct diagnosis results through local analysis, without a centralized coordinator, with no exchange of partial diagnosis results between the distributed diagnosers, and through a minimal exchange of measurement information amongst the distributed diagnosers.

CHAPTER IV

DISTRIBUTED QUALITATIVE DIAGNOSIS OF CONTINUOUS SYSTEMS

Like most model-based diagnosis schemes for continuous systems, the TRANSCEND diagnosis approach is centralized with one monolithic diagnoser that uses a global system model and all the available system measurements [2,5]. Centralized model-based diagnosis schemes have several drawbacks. They are *expensive* in memory and computational requirements. Reliable transmission of measurements to a centralized computer may incur high *costs* for shielding and protection of the cables to maintain signal quality, especially in harsh environments. These approaches *scale poorly* for continuous systems as changes in the system configuration and components may cause significant changes in the system's dynamic behavior, requiring the diagnoser to be redesigned. A centralized approach also creates a *single point of failure*, such that a glitch or failure in the supporting computational units may disable the entire diagnosis system.

The drawbacks of centralized diagnosis schemes motivate the need for distributed diagnosis approaches, where the diagnosis task is broken down into subtasks and executed on separate processors. The distributed diagnosis approach fits well with present day embedded systems architectures, where each subsystem has associated local processors, memory, and sensors for monitoring and control of that subsystem (e.g., electronic control units in aircrafts). In a distributed diagnosis scheme for systems with relatively slow dynamics, such as chemical processes, individual diagnosers implemented for each component can operate independently. The large time constants associated with the global interactions make the subsystem behaviors relatively independent, and the individual diagnosers converge to correct isolation results before the fault effects propagate across subsystem boundaries. Such an approach also works in well-instrumented systems where sensors are placed in close proximity to possible fault sources in individual units, but the cost of employing a large number of sensors may be prohibitive. For system with fast dynamics, such as electromechanical and aerospace systems, fault effects propagate across component boundaries relatively fast, and ignoring component interactions will result in incorrect diagnosis. Hence, extra analysis is required to design distributed diagnosers for such systems.

In this chapter, we extend the centralized qualitative TRANSCEND diagnosis scheme we presented in the previous chapter to a distributed, model-based fault qualitative fault diagnosis scheme for abrupt and incipient faults in continuous systems. In our diagnosis scheme, the local diagnosers gen-

erate globally correct diagnosis results without a centralized coordinator, with minimal exchange of measurement information amongst themselves, and no exchange of partial diagnosis results between themselves. We propose two algorithms for designing the distributed diagnosers. The first algorithm uses predefined subsystem structure to generate, for each subsystem, a local diagnoser that produces globally correct diagnosis results with minimal exchange of information with the other local diagnosers. The second algorithm constructs the system partition structure and local diagnosers simultaneously. The set of diagnosers do not exchange any information between themselves to produce globally correct diagnosis results. We apply both algorithms to a complex, real-world system, the Advanced Water Recovery System developed at the NASA Johnson Space Center [17]. The experimental results demonstrate the computational efficiency and reduction in communication overhead achieved by our distributed diagnosis approach.

The rest of this chapter is organized as follows. The following section presents our distributed diagnosis architecture, where each distributed diagnosis is essentially the qualitative TRANSCEND diagnoser that uses a subset of observations to diagnose a subset of faults. Through the careful design of these distributed diagnosers, we guarantee that each distributed diagnoser will generate globally correct diagnosis results through local analysis, without a centralized coordinator, and no exchange of partial diagnosis results, but through the communication of minimal number of measurement information. The formulation of the diagnoser design problems and the algorithms for designing these distributed diagnosers are described in the next two sections. This is followed by a set of studies that demonstrate the usefulness of this distributed diagnosis approach. The chapter concludes with a summary and comparison of our distributed qualitative diagnosis scheme to other related work in distributed diagnosis. The research contributions presented in this chapter have been published in [11, 95].

The Distributed Diagnosis Approach

The architecture of our distributed diagnosis approach is shown in Fig. 20. Each distributed diagnoser essentially implements the qualitative TRANSCEND diagnoser. In this chapter, we formally define the TRANSCEND diagnoser, $D_{(F,\ M)}$, where F is the set of all possible faults in the system, and M is the set of available measurements. Given M, we define the qualitative measurement residual as $\Sigma^{|M|}$, the $|M|$-dimensional cartesian product of elements in $\Sigma = \{(+,-),\ (-,+),\ (0,+),\ (0,-)\}$, the set of possible symbols representing the magnitude and lowest-order non-zero derivative of individual measurement residuals. Hence, we define $D_{(F,\ M)} = (F, M, H)$, where $H : \Sigma^{|M|} \rightarrow 2^F$ is a mapping

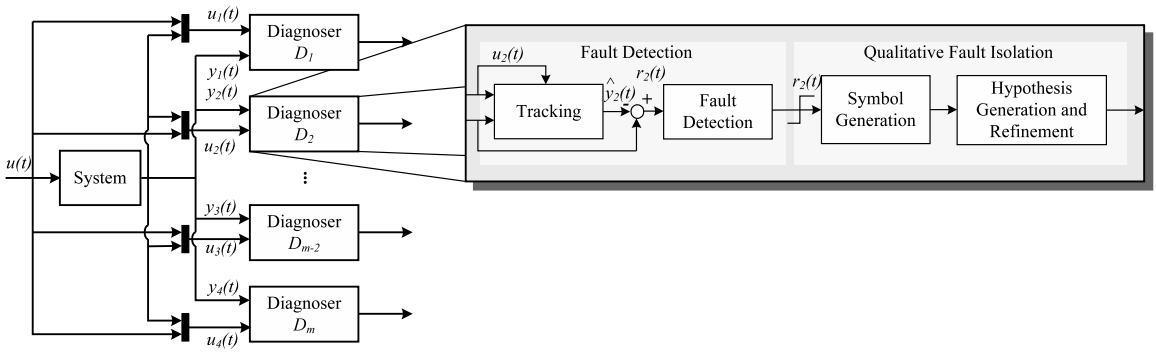


Figure 20: Architecture of the distributed qualitative fault diagnosis approach.

from the qualitative measurement residuals in M to the fault hypotheses set. In TRANSCEND, the mapping H is implemented as described in Chapter III[1]. In Fig. 20, each distributed diagnoser, D_i, denotes a TRANSCEND diagnoser $D_{(F_i,\ M_i)}$, such that $\cup_i F_i = F$, and $\cup_i M_i = M$, and each fault in F_i is globally diagnosable from all faults in F using the its local measurements, M_i, as we present in the remainder of this chapter.

Our objective is to decompose the TRANSCEND fault detection and isolation task into smaller subtasks that can be performed by local diagnosers, such that the distributed diagnosis scheme satisfies the following properties: (*i*) all single faults of interest in the system can be diagnosed, (*ii*) the local diagnosis results obtained by a distributed diagnoser are globally correct, (*iii*) the number of measurements communicated between the distributed diagnosers to satisfy the first two conditions is minimal, and (*iv*) a distributed diagnoser does not need to communicate its partial diagnosis results to any other distributed diagnoser to satisfy properties *i* and *ii*. Properties *i*-*iii* enable distributed diagnosis without a centralized coordinator. We ensure that our approach satisfies the above stated properties through careful design of the distributed diagnosers.

In the following, we use a six-tank system, shown in Fig. 21, as a running example to illustrate the different concepts and algorithms we present in this chapter. Fig. 22 shows the bond graph model of this example six-tank fluid system connected by pipes, with a source of flow, F_{in}, into the first tank, and drain pipes at the bottom of each tank. Pipe R_i drains tank C_i and pipe R_{ij} connects tanks C_i and C_j. In the hydraulic domain, the effort, e, corresponds to pressure, and the flow, f, corresponds the fluid flow rate. Fig. 23 shows the temporal causal graph (TCG) for the six-tank system. While our approach applies to the detection and isolation of both abrupt and incipient faults, for simplicity,

[1]A least squares approach based fault identification step is used in TRANSCEND to not only identify the fault magnitude, but also refine the fault hypothesis set further in case multiple fault hypotheses remain after the qualitative fault isolation step of TRANSCEND. However, the fault identification step is beyond the scope of the work presented in this chapter, and hence not described.

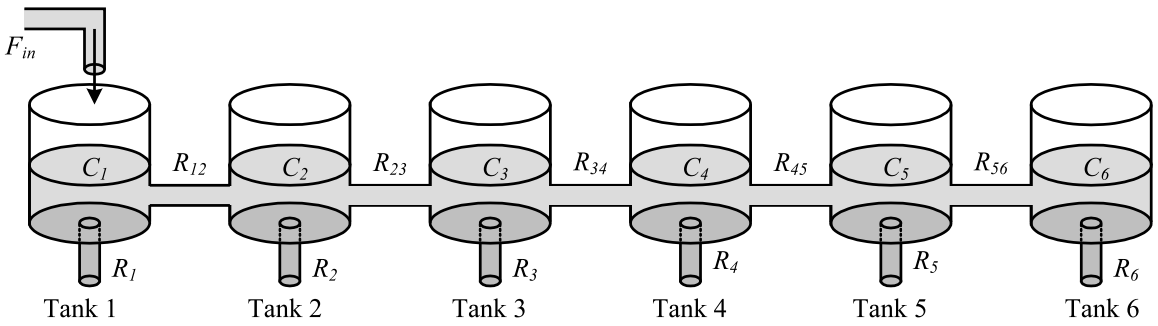


Figure 21: The six-tank system.

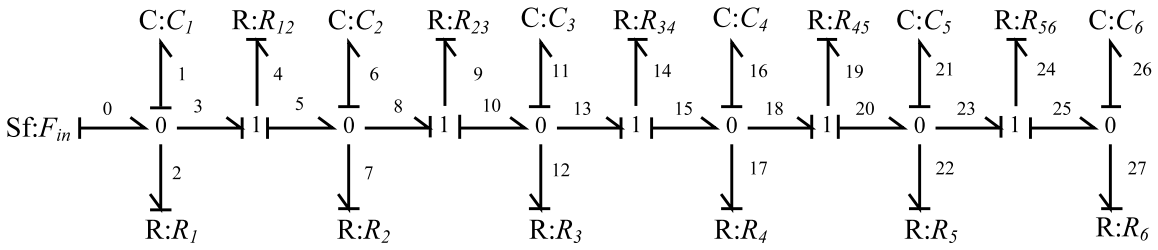


Figure 22: Bond graph model of the six-tank system.

we restrict our examples to abrupt faults only, and consider the fault set $F = \{C_1^{-a}, \ldots, C_6^{-a}, R_1^{+a}, \ldots, R_6^{+a}, R_{12}^{+a}, \ldots, R_{56}^{+a}\}$. The measurements available for diagnosis include the pressures at the bottom of each tank, and fluid flow out of each output pipes, R_1, R_2, ..., R_6. The flow and pressure sensors are not shown in the bond graph in Fig. 22, and hence, in the remainder of this chapter, we refer to a measurement in the six-tank system by its corresponding effort or flow variable in the six-tank system's bond graph, e.g., e_1 denotes the pressure at the bottom of tank C_1, f_2 is the flow out of pipe R_1, and so on. Table 4 lists these measurements, and Table 5 shows some fault signatures from tanks 1 and 2 of the six-tank system obtained using the TCG shown in Fig. 23.

Formulating the Design Problem for Distributed Diagnosis

We assume that the system is *diagnosable* (see Definition 8 in Chapter III) with a centralized diagnosis architecture. The subsystem of the six-tank system, with $F = \{C_1^{-a}, C_2^{-a}, R_2^{+a}\}$, $M = \{e_1, e_6\}$, and the FSM given in Table 5, is diagnosable. However, note that measurement e_1 can discriminate between faults C_1^{-a} and R_2^{+a}, but not C_2^{-a} and R_2^{+a}. Similarly, e_6 can discriminate between $C_2^{-a}$and R_2^{+a}, but not C_1^{-a} and R_2^{+a}. However, e_1 and e_6 together can uniquely isolate all single faults in F, i.e., the system with faults F is diagnosable using the measurements in M. This is a key observation we utilize in our diagnoser design.

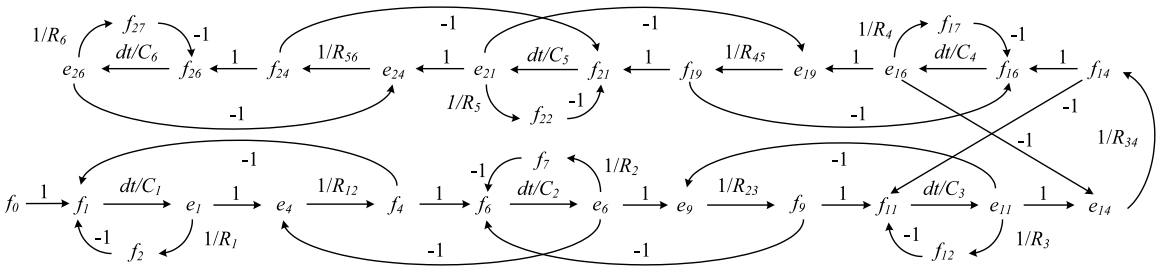


Figure 23: Temporal causal graph for the six-tank system.

Table 4: Six-tank measurements available for diagnosis

Measurement	Description
e_1	Pressure at the bottom of tank 1
e_6	Pressure at the bottom of tank 2
e_{11}	Pressure at the bottom of tank 3
e_{16}	Pressure at the bottom of tank 4
e_{21}	Pressure at the bottom of tank 5
e_{26}	Pressure at the bottom of tank 6
f_2	Flow out of pipt R_1
f_7	Flow out of pipt R_2
f_{12}	Flow out of pipt R_3
f_{17}	Flow out of pipt R_4
f_{22}	Flow out of pipt R_5
f_{27}	Flow out of pipt R_6

Given a system, $S = (F, M, FSM_{(F,\ M)})$, we define subsystem as $S_i = (F_i, M_i, FSM_{(F_i,\ M_i)})$, where F_i is the set of faults, M_i is the set of measurements, and $FSM_{(F_i,\ M_i)}$ is the fault signature matrix corresponding to F_i and M_i. The F_i's and M_i's together form partitions of the set of faults, F, and all measurements, M, respectively. A subsystem, S_i, is *globally diagnosable* if every single fault, $\phi \in F_i$, can be *uniquely isolated with respect to the global fault set* F using the measurements, $\widetilde{M}_i \subseteq M$. We use "global diagnosability" in the context of fault isolability. We can have faults in a subsystem that are "locally" diagnosable from other faults in the subsystem, but which may not be "globally" diagnosable from faults outside the subsystem. Formally, *global diagnosability*, which extends the notion of *diagnosability* in Definition 8, is defined as follows:

Definition 9 (Global Diagnosability)**.** Given the set of all faults, F, $F_i \subseteq F$, is *globally diagnosable* by $\widetilde{M}_i \subseteq M$ if $\widetilde{M}_i$ can uniquely isolate every single fault, $\phi \in F_i$, from all other faults in F, i.e., $\forall(\phi_h \in F_i, \phi_g \in F, \phi_h \neq \phi_g),\ \langle FS(\phi_h, \widetilde{M}_i)\rangle \neq \langle FS(\phi_g, \widetilde{M}_i)\rangle$.

Example. For the six-tank system in Fig. 21, assume $F = \{C_1^{-a}, C_2^{-a}, R_2^{+a}\}$ and $M = \{e_1, e_6\}$. For a subsystem S_1 with $F_1 = \{C_1^{-a}, R_2^{+a}\}$, and $M_1 = \{e_1\}$, S_1 is not globally diagnosable as the fault signature tuples $\langle FS(R_2^{+a}, M_1)\rangle$ and $\langle FS(C_2^{-a}, M_1)\rangle$ are equal. However, S_1 is globally diagnosable with $\widetilde{M}_1 = \{e_1, e_6\}$ since the fault signature tuples $\langle FS(\phi_i, \widetilde{M}_1)\rangle$ for every fault, $\phi_i \in F_1$, are unique, globally.

Table 5: Fault signatures from tanks 1 and 2 of the six-tank system

Fault	e_1	e_6
C_1^{-a}	$+-+-+-+$	$0+-+-+-$
R_2^{+a}	$0+-+-+-$	$0+-+-+-$
C_2^{-a}	$0+-+-+-$	$+-+-+-+$

Each *local diagnoser*, $D_{(F_i,\ \widetilde{M}_i)} = (F_i, \widetilde{M}_i, H_i)$, must satisfy the global diagnosability condition, i.e., all faults in F_i must be globally diagnosable by measurements in $\widetilde{M}_i$. The local diagnosers are each implemented using the TRANSCEND scheme with a distributed, decentralized, extended Kalman filter-based observer (e.g., [69]), a fault detection module, and a symbol generation module. The local diagnosers run independently, and when a measurement deviates, the qualitative fault isolation scheme is triggered for all local diagnosers, which use that measurement for their diagnosis.

We now describe how these local diagnosers generate a global diagnosis result without a coordinator. Assume we have κ local diagnosers $D_{(F_i,\ \widetilde{M}_i)}$, $i = 1, 2, \ldots, \kappa$, such that the fault sets, F_i, form a partition of the set of faults F. For the centralized diagnosis scheme, a diagnosis result is attained when the fault hypothesis set is reduced to a singleton set. In the distributed diagnosis scheme, since the fault sets F_i form a partition of F, we expect only the local diagnoser responsible for diagnosing the true fault to establish a single fault diagnosis, and the others to return empty diagnoses. In practice, we do not have to wait for all the diagnosers to have reached their final diagnosis results. A global diagnosis result is obtained when:

1. *All* measurements for a local diagnoser have deviated and the fault hypothesis set is reduced to a singleton fault set, or,

2. A local diagnoser's hypothesis set is reduced to a singleton but all of its measurements have not deviated, *and* all other diagnosers produce a *null hypothesis*, i.e., their candidate sets are empty.

Each local diagnoser reports its single or null hypothesis result independently, and the system diagnosis result is determined once conditions 1 or 2 are satisfied. The local diagnosers do not have to communicate with one another to establish their diagnosis results.

We assume that the system under consideration is *diagnosable*, and develop two different problems for designing distributed diagnosers:

1. In the first problem, we assume the system partition is known and construct local diagnosers for each subsystem that exchange minimal information to globally diagnose each subsystem.

2. In the second problem, we create the system partition structure and local diagnosers simultaneously, in a way that no measurements are shared between the subsystems.

The first problem applies to designing diagnostic schemes for distributed systems with known partition structures. The second problem is more open-ended, and the system partition structure and corresponding diagnosers are derived simultaneously at design time to ensure efficient distributed diagnosis.

In situations when the system is not globally diagnosable for a set of measurements, we can define the notion of "aggregate faults". An aggregate fault includes all single faults that have the same fault signatures for the available measurements, and hence, are not distinguishable from one other. Our diagnosis methodology can be applied to the reduced fault set with the indistinguishable faults represented as aggregate faults.

Formally, the two problems can be defined as follows:

Problem 1 (Partitioned System Diagnoser Design)**.** Given κ subsystems, $S_i = (F_i, M_i, FSM_{(F_i,\ M_i)})$, $1 \leq i \leq \kappa$, construct, for each subsystem, a measurement set $\widetilde{M_i} \subseteq M$ such that *(i)* $\widetilde{M_i} - M_i$ is minimal, and *(ii)* all single faults in F_i are globally diagnosable by measurements in $\widetilde{M_i}$. Given F_i and $\widetilde{M_i}$, we construct a local diagnoser, $D_{(F_i,\ \widetilde{M_i})}$, for each subsystem. By ensuring that $\widetilde{M_i} - M_i$ is minimal, the local diagnosers share minimal information with one another.

Problem 2 (Unpartitioned System Diagnoser Design)**.** Given a system $S = (F, M, FSM_{(F,\ M)})$, partition F and M into fault and measurement subsets, F_i and $\widetilde{M_i}$, respectively, such that all single faults in F_i are globally diagnosable using measurements only in $\widetilde{M_i}$. From each F_i and $\widetilde{M_i}$ subset pairs, we then construct local diagnosers $D_{(F_i,\ \widetilde{M_i})}$ that do not share any measurements.

These two problems are variations of the *measurement selection* problem [94], with applications in control engineering [96], structural dynamics [97], and robotics [98], among others. The measurement selection problem is an instance of the set covering problem [99], which is known to be NP-complete. Our goal, while designing the local diagnosers, is to select fault-measurement sets that together make the system completely diagnosable, with an emphasis on minimizing the sharing of measurements across sets. For Problem 1, measurement selection is applied to each subsystem with the constraint that the local diagnosis results must be globally correct. Problem 2 represents a "double" measurement selection problem because of the simultaneous partitioning of the fault and measurement sets to ensure that the local diagnosers generate globally correct diagnosis results with no information exchange. To avoid the exponential complexity, we use heuristics that exploit our knowledge of system dynamics to derive less expensive solutions for both problems.

Algorithm 1 Designing Diagnosers for a Partitioned System

Input: κ local subsystems, $S_i = (F_i, M_i, FSM_{(F_i,\ M_i)})$
for each S_i **do**
  identify $remFaults_i \in F_i$ that cannot be uniquely isolated using M_i.
for each $remFaults_i$ **do**
  $\delta = 1$; $\widetilde{M}_i = M_i$
  while $remFaults_i \neq \varnothing$ **do**
    identify measurement set $\widehat{M}_i$ from measurements of subsystems S_i at a distance $d \leq \delta$ that isolates maximal $r \in remFaults_i$ and $\widetilde{M}_i - \widehat{M}_i$ is minimal.
    $\widetilde{M}_i = \widetilde{M}_i \cup \widehat{M}_i$
    $remFaults_i = remFaults_i - r$
    if $remFaults_i \neq \varnothing$ **then**
      $\delta = \delta + 1$
  construct $D_{(F_i,\ \widetilde{M}_i)}$

Designing the Distributed Diagnosers

We present the two algorithms for generating the distributed diagnosers for continuous systems.

Implementing Distributed Qualitative Isolation

We now describe how these local diagnosers generate a global diagnosis result without a coordinator. Assume we have κ local diagnosers $D_{(F_i,\ \widetilde{M}_i)}$, $i = 1, 2, \ldots, \kappa$, such that the fault sets, F_i, form a partition of the set of faults F. For the centralized diagnosis scheme, a diagnosis is reached when the fault hypothesis set is reduced to a singleton set. In the distributed diagnosis scheme, since the fault sets F_i form a partition of F, we expect only the local diagnoser responsible for diagnosing the true fault to establish a single fault diagnosis, and the others to return empty diagnoses. In practice, we do not have to wait for all the diagnosers to have reached their final diagnosis results. A global diagnosis result is obtained when:

1. *All* measurements for a local diagnoser have deviated and the fault hypothesis set is reduced to a singleton fault set, or,

2. A local diagnoser's hypothesis set is reduced to a singleton but all of its measurements have not deviated, *and* all other diagnosers produce a *null hypothesis*, i.e., their candidate sets are empty.

Each local diagnoser reports its single or null hypothesis result independently, and the system diagnosis result is determined once conditions 1 or 2 are satisfied. The local diagnosers do not communicate with one another to establish their diagnosis results.

Table 6: Fault signatures for the six-tank system example

$Fault$	e_1	f_2	e_6	f_7	e_{11}	f_{12}	e_{16}	f_{17}	e_{21}	f_{22}	e_{26}	f_{27}
C_1^{-a}	+−	+−	0+	0+	0+	0+	0+	0+	0+	0+	0+	0+
C_2^{-a}	0+	0+	+−	+−	0+	0+	0+	0+	0+	0+	0+	0+
C_3^{-a}	0+	0+	0+	0+	+−	+−	0+	0+	0+	0+	0+	0+
C_4^{-a}	0+	0+	0+	0+	0+	0+	+−	+−	0+	0+	0+	0+
C_5^{-a}	0+	0+	0+	0+	0+	0+	0+	0+	+−	+−	0+	0+
C_6^{-a}	0+	0+	0+	0+	0+	0+	0+	0+	0+	0+	+−	+−
R_{12}^{+a}	0+	0+	0−	0−	0−	0−	0−	0−	0−	0−	0−	0−
R_{23}^{+a}	0+	0+	0+	0+	0−	0−	0−	0−	0−	0−	0−	0−
R_{34}^{+a}	0+	0+	0+	0+	0+	0+	0−	0−	0−	0−	0−	0−
R_{45}^{+a}	0+	0+	0+	0+	0+	0+	0+	0+	0−	0−	0−	0−
R_{56}^{+a}	0+	0+	0+	0+	0+	0+	0+	0+	0+	0+	0−	0−

Designing Diagnosers for a Partitioned System

Problem 1 designs a local diagnoser for each subsystem $S_i = (F_i, M_i, FSM_{(F_i,\ M_i)})$ using the local measurements, M_i and additional measurements, if required. The goal is to minimize the number of additional measurements, while ensuring that each subsystem is globally diagnosable. For each subsystem S_i, we identify the faults that are not globally diagnosable given M_i, and then, search for a minimal number of additional measurements that will make these faults globally diagnosable.

The search is simplified by defining a notion of proximity among subsystems and using this information to prioritize the selection of additional measurements for a local diagnoser. We represent the system, S, as a graph of connected subsystems. Each subsystem, S_i, forms a node of the graph, and an undirected edge (S_g, S_h) implies direct energy or information exchange between S_g and S_h. The proximity d is defined as the minimum path length from S_g to S_h. The search for additional measurements starts from closer subsystems.

The procedure for designing diagnosers for a partitioned system is presented in Algorithm 1. For each subsystem S_i, we assign to $remFaults_i$ the faults in F_i that cannot be uniquely isolated using measurements in M_i. When $remFaults_i$ is not empty, we start by assigning $\widetilde{M_i}$ equal to M_i, and generating a working measurement set $\widetilde{M_i}^{d\leq 1}$ by pooling in measurements from all subsystems, S_l, at a distance $d \leq 1$ from subsystem S_i, i.e., $\widetilde{M_i}^{d\leq 1} = \bigcup_l M_l$. Using the measurement selection algorithm in [94] we select additional measurements from $\widetilde{M_i}^{d\leq 1} - M_i$ to reduce the number of faults in $remFaults_i$. When different measurement combinations provide the same reductions, we pick the measurement set $\widehat{M_i}$ that adds minimal number of external measurements to M_i while making the maximum number of faults in $remFaults_i$ globally diagnosable. The set $\widetilde{M_i}$ is expanded, and $remFaults_i$ is reduced to a smaller set. If $remFaults_i$ is non-empty, d is incremented, and the procedure is repeated till $remFaults_i$ is empty. At this point, we have the local diagnoser $D_{(F_i,\ \widetilde{M_i})}$.

The search algorithm is complete as it will always find the measurements required to diagnose all faults in $remFaults_i$.

We apply this algorithm to the six-tank system example of Fig. 21 with $F = \{C_1^{-a}, \ldots, C_6^{-a}, R_{12}^{+a}, \ldots, R_{56}^{+a}\}$ and $M = \{e_1, f_2, e_6, f_7, e_{11}, f_{12}, e_{16}, f_{17}, e_{21}, f_{22}, e_{26}, f_{27}\}$. The fault signature matrix for the fault and measurement sets appear in Table 6. Each tank and the pipe connecting it to the tank on its right defines a subsystem. The six subsystems include the fault sets $F_1 = \{C_1^{-a}, R_{12}^{+a}\}$, $F_2 = \{C_2^{-a}, R_{23}^{+a}\}$, $F_3 = \{C_3^{-a}, R_{34}^{+a}\}$, $F_4 = \{C_4^{-a}, R_{45}^{+a}\}$, $F_5 = \{C_5^{-a}, R_{56}^{+a}\}$, and $F_6 = \{C_6^{-a}\}$, and the measurement sets $M_1 = \{e_1, f_2\}$, $M_2 = \{e_6, f_7\}$, $M_3 = \{e_{11}, f_{12}\}$, $M_4 = \{e_{16}, f_{17}\}$, $M_5 = \{e_{21}, f_{22}\}$, and $M_6 = \{e_{26}, f_{27}\}$.

Algorithm 1 generates the following local diagnosers: ($\{C_1^{-a}, R_{12}^{+a}\}$,$\{e_1, f_2, \mathbf{e_6}\}$, H_1), ($\{C_2^{-a}, R_{23}^{+a}\}$, $\{e_6, f_7, \mathbf{e_{11}}\}$, H_2), ($\{C_3^{-a}, R_{34}^{+a}\}$, $\{e_{11}, f_{12}, \mathbf{e_{16}}\}$, H_3), ($\{C_4^{-a}, R_{45}^{+a}\}$, $\{e_{16}, f_{17}, \mathbf{e_{21}}\}$, H_4), ($\{C_5^{-a}, R_{56}^{+a}\}$, $\{e_{21}, f_{22}, \mathbf{e_{26}}\}$, H_6), ($\{C_6^{-a}\}$, $\{e_{26}, f_{27}\}$, H_7)). The external measurements required for global diagnosability appear in bold. A capacitance fault for the i^{th} tank is diagnosable by the effort variable of that tank, but to achieve global diagnosis of the interconnecting pipe faults, the algorithm adds the pressure variable e_{i+1} of the adjoining tank to the measurement set of tank i.

The distributed diagnosis scheme improves the centralized diagnosis approach. Given the system $S = (F, M, FSM_{(F,\ M)})$, we define the size of a centralized diagnoser, $D_{(F,\ M)}$, as the size of its FSM, i.e., $|D_{(F,\ M)}| = |F| \times |M|$. On the other hand, with κ local diagnosers, $D_{(F_i,\ \widetilde{M}_i)}$, the total FSM size is $\sum_i |D_{(F_i,\ \widetilde{M}_i)}| = \sum_i (|F_i| \times |\widetilde{M}_i|)$. Hence, the total space requirement for all local diagnosers generated using Algorithm 2 will never exceed that of a centralized diagnoser, i.e., $\sum_i |D_{(F_i,\ \widetilde{M}_i)}| \leq |D_{(F,\ M)}|$. Only a few measurements are communicated between local diagnosers, so there is considerable savings with the distributed diagnosers.

The computational complexity for deriving the diagnosers for subsystem S_i depends on the number of faults $|F_i|$. The algorithm to find $remFaults_i$ is $O(|F_i|^2)$. To diagnose every element of $remFaults_i$, which in the worst case, can be of size $O(|F_i|)$, we assume m is the maximum number of measurements in subsystems at a distance of $d = 1$. In the worst case, the algorithm will have to generate all possible combinations of these measurements, i.e., $O(m^{\lfloor \frac{m}{2} \rfloor})$, and the algorithm to identify the measurement combination that isolates maximal faults in $remFaults_i$ while adding the least number of external measurements has complexity $O(m^{\lfloor m \rfloor}|F_i|^2 + |F_i|^2)$. Usually $|remFaults_i| << |F_i|$, and using the measurement selection method in [94] reduces the complexity of this operation to a much smaller value. In the worst case, for all of the κ subsystems, the complexity of the algorithm is $O(\kappa|M|^{\lfloor |M| \rfloor}|F_i|^2 + \kappa|F_i|^2)$, but the average run-time performance of this algorithm is much better. In continuous systems we seldom need to look beyond the immediate neighbors of

Algorithm 2 Designing Diagnosers for an Unpartitioned System

Input: Global system $S = (F, M, FSM_{(F,\ M)})$
generate *root node* $R = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), ..., (\widetilde{M_{|M|}}, F_{|M|})]$ s.t. $|\widetilde{M_i}| = 1$
level $l = 1$
while true **do**
  check for *goal node*, $G = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), ..., (\widetilde{M_n}, F_n)]$, at level l, s.t. $\cup_i F_i = F$
  if *goal node* G is found **then**
    for each $F_i \in G$ such that $F_i \neq \varnothing$ **do**
      construct $D_{(F_i,\ \widetilde{M_i})}$
    return
  else
    identify node N s.t. $FC_N = h$
    expand node N to generate level $l + 1$ of search tree

each subsystem for measurements that diagnose all faults in that subsystem. The tractability of the approach is illustrated in our case study on the ALS system.

Designing Diagnosers for an Unpartitioned System

Problem 2 assumes no prior knowledge of subsystem structure for the system S. The goal is to partition the system into subsystems, and construct local diagnosers for each subsystem that satisfy global diagnosability, and do not have to share measurements to achieve global diagnosability. Algorithm 2 solves this problem by generating the maximum number of local diagnosers that do not share measurements, with an added constraint that the measurement subsets are balanced across the diagnosers.

Let $P_I(M)$ denote a partition for the set of measurements, M, in a system, and assume F_i is the set of faults that are globally diagnosable using every $\widetilde{M_i} \in P_I(M)$. Note that F_i can be empty. If $\bigcup_i F_i = F$, for every non-empty F_i, we can construct a set of local diagnosers, $D_{(F_i,\ \widetilde{M_i})} = (F_i, \widetilde{M_i}, H_i)$, that make the system globally diagnosable. The solution to Problem 2 is developed as a tree search algorithm. Each node N of the tree is defined as $N = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), \ldots, (\widetilde{M_n}, F_n)]$ such that $\widetilde{M_i} \in P_N(M)$ and F_i is globally diagnosable with $\widetilde{M_i}$. Our goal is to construct the largest number of local diagnosers which together can globally diagnose all faults in F. Hence our goal node is a node N that partitions M into the largest number of subsets, i.e., $|P_N(M)|$ is maximal, and $\bigcup_i F_i = F$.

The *root node*, R, of the tree is $R = [(\widetilde{M_1}, F_1),\ (\widetilde{M_2}, F_2), ..., (\widetilde{M_{|M|}}, F_{|M|})]$, where each $\widetilde{M_i}$ is represented by a single measurement, i.e., $|\widetilde{M_i}| = 1$. For each $\widetilde{M_i}$, we derive the corresponding F_i such that $\widetilde{M_i}$ produces a global diagnosis for F_i. For a *goal node*, $G = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), ..., (\widetilde{M_n}, F_n)]$, the fault sets F_i cover the set of all faults F, i.e., $\bigcup_i F_i = F$.

The search algorithm generates nodes at level $l+1$ of the tree by creating all possible pairs of measurement sets from the parent nodes at level l, and computing the corresponding fault-sets for the new measurement sets. For example, if for node $N_1 = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), (\widetilde{M_3}, F_3)]$, the following nodes will be formed as children of this node: $N_2 = [(\widetilde{M_1} \cup \widetilde{M_2}, F_{12}), (\widetilde{M_3}, F_3)]$, $N_3 = [(\widetilde{M_1} \cup \widetilde{M_3}, F_{13}), (\widetilde{M_2}, F_2)]$, and $N_4 = [(\widetilde{M_2} \cup \widetilde{M_3}, F_{23}), (\widetilde{M_1}, F_1)]$. Note that F_{ij}, the set of faults that are globally diagnosed by measurements in $\widetilde{M_i} \cup \widetilde{M_j}$, can include more faults than $F_i \cup F_j$. This is because the two sets of measurements may uniquely diagnose more faults than the sum of the faults that each can diagnose.

For every new level added to the tree, the algorithm checks if any of the new nodes is a goal node. If there are none, the merge process is repeated at the next level of search till a *goal node* is found. Exhaustive expansion of all nodes at each level would result in an algorithm whose search space and search time are doubly exponential. To reduce computational complexity, our algorithm imposes a greedy heuristic to choose a single node for expansion. If $[N]_l$ represents the set of all nodes at a level l in the search tree, we define our heuristic function $h_l = \max_{\forall N \in [N]_l}(FC_N)$, where $FC_N = |\bigcup_i F_i|$ denotes the total number of faults that are diagnosable in node N by the measurements in $P_N(M)$. Intuitively, at any level, the greedy approach prefers nodes whose local diagnosers can together diagnose the maximum number of faults, i.e., the node with the largest FC_N value is chosen for expansion. The process is repeated till a *goal node* is found.

For a goal node, $G = [(\widetilde{M_1}, F_1), (\widetilde{M_2}, F_2), \ldots, (\widetilde{M_n}, F_n)]$, we construct local diagnosers, $D_{(F_i,\ \widetilde{M_i})}$, for every fault measurement subset pair, if F_i is not empty. If a fault is uniquely diagnosable by more than one $\widetilde{M_i}$, we assign the fault to the local diagnoser that uses the smallest $\widetilde{M_i}$. This results in balanced diagnosers. It should be noted that for tightly coupled systems, it is possible that the the only solution found by Algorithm 2 is $G = [(M, F)]$, i.e., the system cannot be partitioned.

Algorithm 2 applied to the six-tank system produces seven local diagnosers: $(\{C_1^{-a}\},\ \{e_1\},\ H_1)$, $(\{C_2^{-a},\ R_{12}^{+a}\},\ \{e_6\},\ H_2)$, $(\{C_3^{-a},\ R_{23}^{+a}\},\ \{e_{11},\ f_7\},\ H_3)$, $(\{C_4^{-a},\ R_{34}^{+a}\},\ \{e_{16},\ f_{12}\},\ H_4)$, $(\{C_5^{-a},\ R_{45}^{+a}\},\ \{e_{21},\ f_{17}\},\ H_5)$, $(\{C_6^{-a}\},\ \{f_{27}\},\ H_6)$, $(\{R_{56}^{+a}\},\ \{e_{26},\ f_{22}\},\ H_7)$. When one compares the number of node expansions required to generate the solutions, an exhaustive search used 183,074 node expansions, and Algorithm 2 derived its solution with 203 node expansions. We have run a number of other experiments with n-tank systems ($6 \leq n \leq 15$), and in almost all cases, the heuristic algorithm expanded 1% of the nodes that would be generated by the exhaustive algorithm. This demonstrates that the heuristic algorithm is efficient and generates acceptable solutions.

Like Algorithm 1, $\sum_i |D_{(F_i,\ \widetilde{M}_i)}| = \sum_i (|F_i| \times |\widetilde{M}_i|)$, the size of the local diagnosers is smaller than $|D_{(F,\ M)}|$. Hence, there is considerable space complexity improvement using distributed diagnosers designed by Algorithm 2.

To analyze the time complexity of Algorithm 2, assume $|F| = l$ and $|M| = n$. The root node has n local diagnosers. For each measurement set, $\widetilde{M}_i$, we identify the set of faults, F_i, diagnosable by the measurements in $\widetilde{M}_i$. The faults in F_i have unique fault signatures for the measurements in $\widetilde{M}_i$ and they are computed by traversing the columns of the fault signature matrix, $FSM_{(F,\ M)}$, that correspond to the measurements in $\widetilde{M}_i$. This operation can be computed in $O(l^2 n)$ time. To expand the node N, we merge all pairs of $\widetilde{M}_i \in P_N(M)$ to obtain the measurement sets of the children nodes. Therefore, we have $\binom{n}{2}$ nodes in the next level and each node will have $(n-1)$ measurement sets, $\widetilde{M}_i$. Identifying the fault sets, F_i, for each node at this level is also $O(l^2 n)$. Since, we are expanding only one node, we will have only $\binom{n-1}{2}$ children. The number of nodes generated is $\binom{n}{2} + \binom{n-1}{2} + \binom{n-3}{2} + \ldots + \binom{2}{2} = O(n^3)$ as there are at most n levels. Hence the complexity of Algorithm 2 is $O(l^2 n^4)$, which is polynomial in the number of faults and measurements.

Case Study: The Advanced Water Recovery System

System Overview

We apply our distributed diagnosis approach to a large real-world system, the Advanced Water Recovery System (AWRS), designed and built at the NASA Johnson Space Center (JSC) as part of Advanced Life Support (ALS) Systems for long duration manned missions [17]. The AWRS, shown in Fig. 24, is a closed loop system that converts wastewater to potable water in microgravity conditions.

The conversion of wastewater, stored in the *Wastewater Tank*, is a multi-step process that starts with a *Biological Waste Processor* (BWP), which removes organic matter and ammonia from the wastewater, followed by a *Reverse Osmosis Subsystem* (RO), which removes inorganic and particulate matter using a *high pressure membrane filtration* system. The concentrated brine that collects in the RO is passed into the *Air Evaporation Subsystem* (AES), which recovers the remaining water using a cyclic *evaporation* and *condensation* process. Finally, the *Post Processing Subsystem* (PPS) uses ultraviolet light treatment to remove trace impurities and infectants from the RO and AES effluents, and the potable water produced is collected in the *Potable Water Tank*.

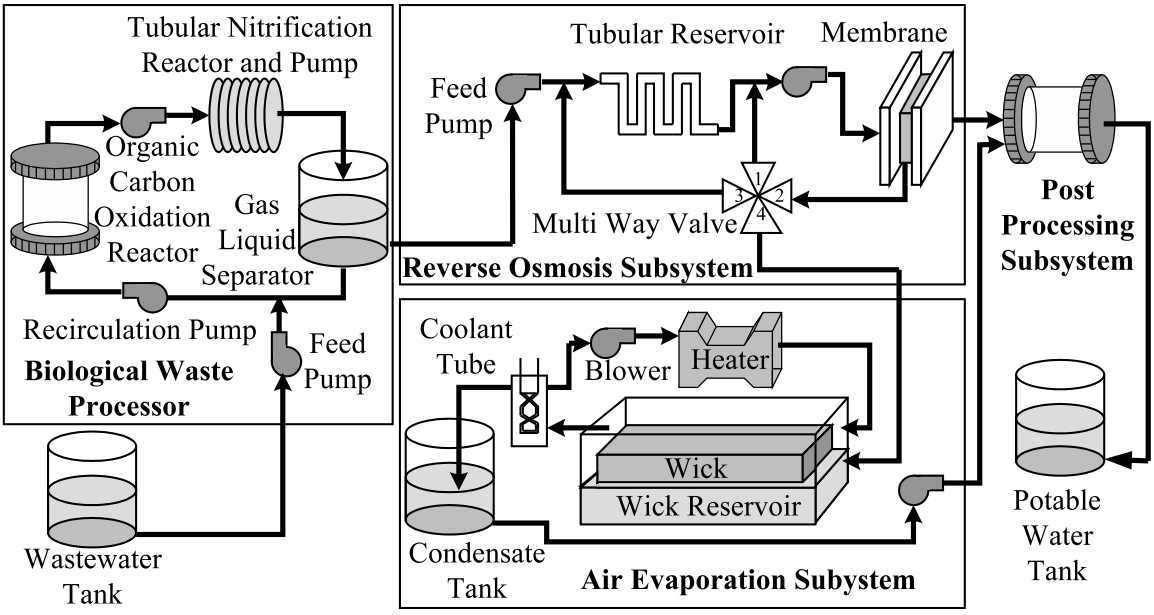


Figure 24: Schematic of the Advanced Water Recovery System.

Biological Water Processor

The bond graph model of the BWP is shown in Fig. 25. A *feed pump*, modeled as a constant flow pump using the single flow source, Sf_{fp}, feeds wastewater into the BWP. The *Organic Carbon Oxidation Reactor* (OCOR), which oxidizes the organic carbon, is modeled as a tank, C_{ocor}. The effluent from OCOR enters the *Nitrification Reactor* (NR) through the R_{ocor} pipe. The NR has four parallel tubes (NR_i, $1 \leq i \leq 4$) with nitrifying microorganisms packed into each tube, and a boost pump that maintains the flow. The resistance R_{NR_i} of NR_i is modeled to increase as wastewater flows through the pipe, simulating the growth of microorganims as they feed on the organic waste[2]. The effluent of the NR is sent to an ambient pressure *gas-liquid separator* (GLS), modeled as C_{gls}, where the majority of the water effluent is recycled back to the OCOR by the *recirculation pump*, and a smaller stream, equal to the initial feed during steady state operations, is transferred to the RO subsystem for further processing. The recirculation pump is modeled as a simple boost pump with two bond graph elements: an effort source, Se_{rp}, and the pump rotor intertia, I_{rp}. R_{pipe1} and R_{pipe2} model the pipes between the feed pump and the OCOR, and the GLS and the recirculation pump, respectively.

[2]Note that in the bond graphs, components modulated as a function of system variables have a prefix M added to their names, e.g., MR_{NR_i} denotes that R_{NR_i} is modulated by the flow of water through it. An arrow pointing from the system variable to a modulated component represents this modulation graphically.

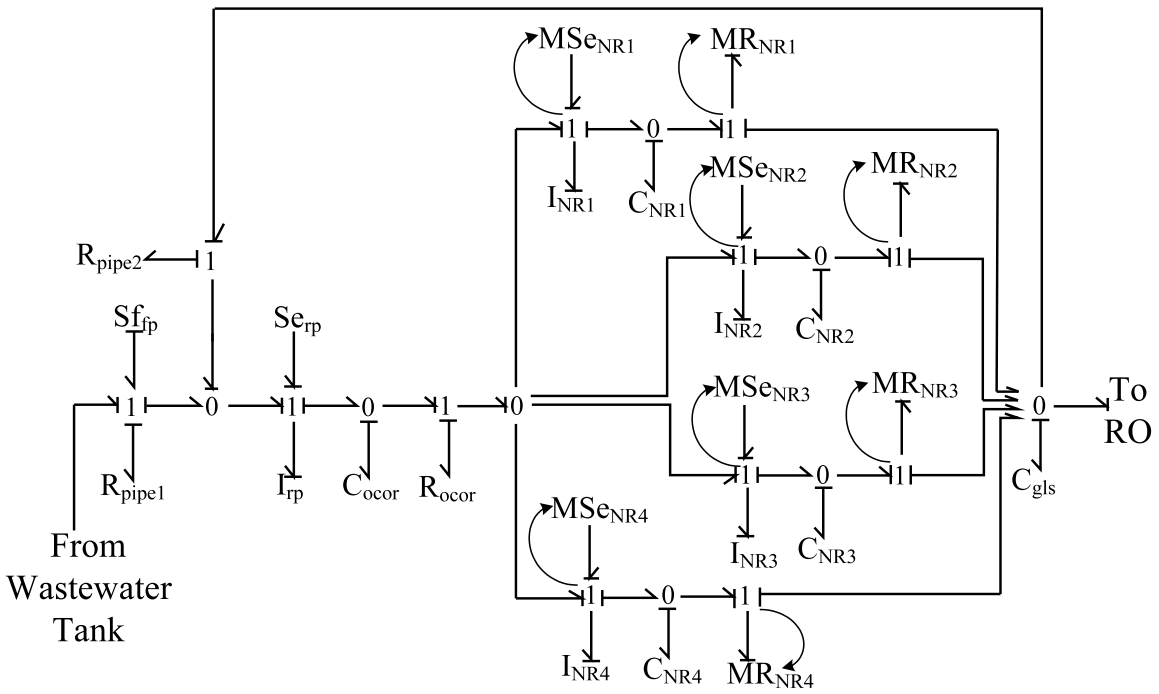


Figure 25: Bond graph model of the Biological Water Processor.

Reverse Osmosis Subsystem

Fig. 26 shows the bond graph model for the RO subsystem. The *feed pump* that moves effluent from the BWP into the RO is modeled as a source of effort, Se_{fp}, with rotor inertia, I_{fp}, and resistance, R_{fp}, to model frictional losses. The transformer, TF_{fp}, models the conversion of rotational speed to fluid flow. A coiled pipe, modeled as C_{tubRes}, acts as a tubular reservoir to help reduce fluctuations in liquid flow through the system. The connecting pipe is modeled as a resistance R_{pipe}. The RO subsystem operates in multiple modes, determined by the 4-way multi-position valve, but in this work, we restrict the RO to the primary mode of operation where the water circulates in a longer loop. The *recirculation pump* has parameters Se_{rp}, R_{rp}, I_{rp}, and GY_{rp}. The *membrane* assembly is modeled as a fixed chamber with capacitor, C_{memb}, and a variable resistance, R_{memb}, that models the resistance to flow through the membrane. Dirt accumulates as waste water flows through the membrane causing R_{memb} to increase, and the outflow of clean water to decrease with time. Hence, the resistance, R_{memb}, is modulated by the *conductivity* (K) of the water flowing in the system. The water that does not pass through the membrane has a greater concentration of impurities, and is recirculated through the pipe, R_{brine}.

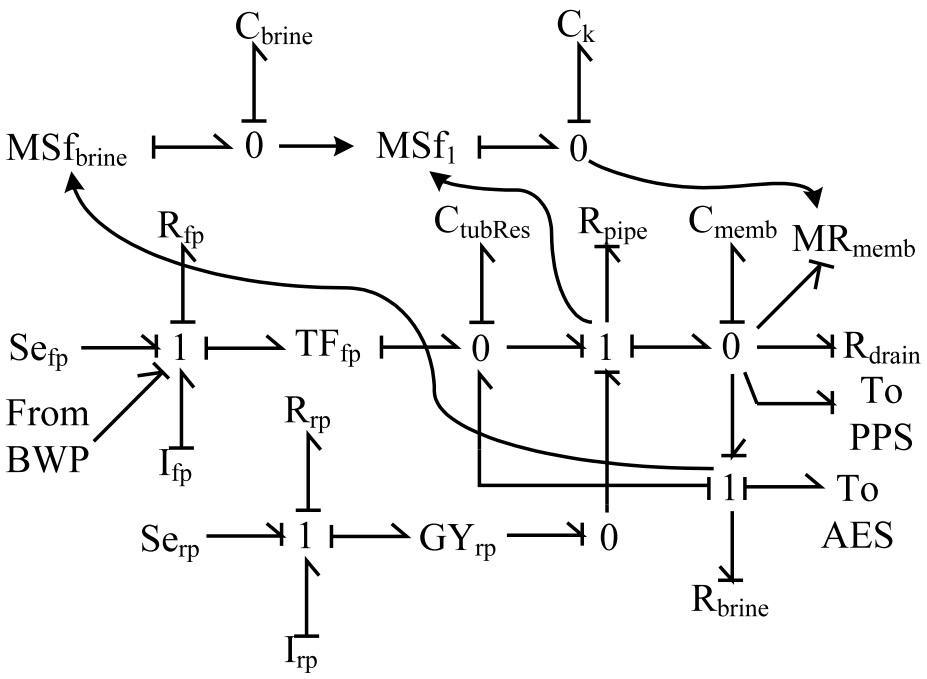


Figure 26: Bond graph model of the Reverse Osmosis Subsystem.

Air Evaporation Subsystem

The bond graph, shown in Fig. 27, models the AES. It includes the *wick*, a porous element modeled as C_{wick}, which dips into the brine that is collected in a tank. Warm air blown over the wick evaporates some of the water. C_{steam} represents the quantity of vapor generated due to the evaporation. The moisture laden air is then passed through a chilled water heat exchanger, and clean condensate is collected in the condensate tank, $C_{condensate}$. The *condensate pump*, modeled as a simple source of flow, $Sf_{condFlow}$, pumps water to the PPS in a continuous stream. A blower (modeled as Se_{blower}) is used in the airflow loop to maintain the flowrate, and a heater (Se_{heater}) heats up the air cooled in the exchanger to ensure that its capacity to absorb moisture remains high. The transformers, TF_{blower} and TF_{heater}, model the efficiency of the blower and the heater, respectively. The energy exchanges and temperature content at different parts of the air in the AES is modeled as capacitors C_{air_i} $(1 \leq i \leq 3)$. $R_{airFlow}$ models the resistance to the flow of air in the AES heat exchange loop.

Post Processing Subsystem

The PPS disinfects the effluent from the RO and the AES components through a five step treatment procedure to generate potable water. Since the PPS does not have interesting flow dynamics, we do not include it in our diagnosis model.

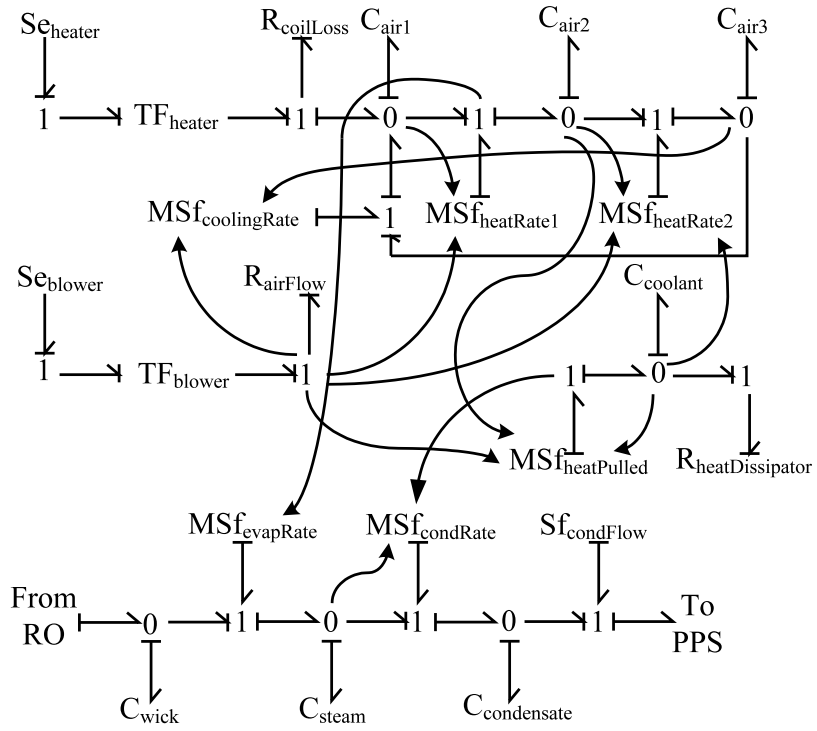


Figure 27: Bond graph model of the Air Evaporation System.

The multi-domain bond graph models represent the mechanical and hydraulic domains in the BWP, RO, and AES. The RO bond graph also models the fluid conductivity domain, to simulate the changing concentration of impurities and their effects on the flow process. The AES bond graph models the exchange of heat between the water absorbed by the wick, the air, and the coolant liquid in the thermal domain.

The AWRS is a large, complex, physical system with interacting subsystems, each containing a large number of components. These interactions cause fault effects to propagate across subsystem boundaries, eventually affecting all parts of the system. Hence, a centralized approach, when applied to this system, will have high memory and computation requirements. On the other hand, the well-defined subsystem structure of the AWRS lends itself well to our distributed diagnosis approach.

Diagnoser Design Experiments

The AWRS testbed is well instrumented. Table 7 shows the list of measurements and faults that we chose for these experiments. In the following, we first derive diagnosers for the three AWRS subsystems using three measurements sets. Then diagnoser-design experiments are run assuming the subsystem structure is unknown.

Table 7: Measurements and faults chosen for the experiments

Subsystem	Measurement	Description	Fault	Description
BWP	$BWP.P_{rp}$	Recirculation pump output pressure	$BWP.C^{-a}_{gls}$	Buildup of sediments in the GLS tank
	$BWP.P_{gls}$	Pressure in the GLS tank	$BWP.R^{+a}_{ocor}$	Blockage in the R_{ocor} pipe
	$BWP.P_{ocor}$	Pressure at the output of the OCOR	$BWP.R^{+a}_{pipe1}$	Blockage in the R_{pipe1} pipe
	$BWP.P_{fp}$	Pressure at the output of the feed pump	$BWP.R^{+a}_{pipe2}$	Blockage in the R_{pipe2} pipe
	$BWP.F_{NR_1}$	Flow through nitrifier tube 1	$BWP.R^{+a}_{NR_1}$	Blockage in nitrifier tube 1
	$BWP.F_{NR_2}$	Flow through nitrifier tube 2	$BWP.R^{+a}_{NR_2}$	Blockage in nitrifier tube 2
	$BWP.F_{NR_3}$	Flow through nitrifier tube 3	$BWP.R^{+a}_{NR_3}$	Blockage in nitrifier tube 3
	$BWP.F_{NR_4}$	Flow through nitrifier tube 4	$BWP.R^{+a}_{NR_4}$	Blockage in nitrifier tube 4
RO	$RO.P_{memb}$	Pressure in the membrane	$RO.R^{+a}_{brine}$	Blockage in the pipe carrying brine
	$RO.F_{permeate}$	Permeate flow rate	$RO.C^{-a}_{memb}$	Buildup of sediment in the membrane
	$RO.P_{rp}$	Pressure at the output of the feed pump	$RO.R^{+a}_{memb}$	Blockage of flow through the membrane
	$RO.P_{back}$	Backflow pressure	$RO.R^{+a}_{pipe}$	Blockage in pipe carrying water to the membrane
			$RO.TF^{-a}_{fp}$	Decrease in efficiency of the feed pump
			$RO.GY^{-a}_{rp}$	Decrease in efficiency of the recirculation pump
AES	$AES.V_{air}$	Velocity of air flowing through the wick	$AES.TF^{-a}_{blower}$	Decrease in efficiency of the blower
	$AES.P_{wick}$	Pressure in the wick	$AES.TF^{-a}_{heater}$	Decrease in efficiency of the heater
	$AES.P_{steam}$	Pressure of the steam generated	$AES.C^{-a}_{wick}$	Buildup of sediment in the wick
	$AES.P_{condensate}$	Pressure in the condensate tank	$AES.C^{-a}_{steam}$	Decrease in the capacity to produce steam
	$AES.T_{coolant}$	Temperature of the coolant	$AES.C^{-a}_{condensate}$	Buildup of sediment in the condensate tank
			$AES.R^{+a}_{airFlow}$	Reduction of airflow

Table 8: Results for Experiments 1-A, 1-B, and 1-C

Faults	Maximal number (17) of Measurements Considered	Minimal number (14) of Measurements Considered	Intermediate number (16) of Measurements Considered
$BWP.C^{-a}_{gls}$, $BWP.R^{+a}_{ocor}$	$BWP.P_{rp}$, $BWP.P_{gls}$		$BWP.P_{rp}$
$BWP.R^{+a}_{pipe1}$, $BWP.R^{+a}_{pipe2}$	$BWP.P_{ocor}$, $BWP.P_{fp}$	$BWP.P_{ocor}$, $BWP.P_{fp}$	$BWP.P_{ocor}$, $BWP.P_{fp}$
$BWP.R^{+a}_{NR_1}$, $BWP.R^{+a}_{NR_2}$	$BWP.F_{NR_1}$, $BWP.F_{NR_2}$	$BWP.F_{NR_1}$, $BWP.F_{NR_2}$	$BWP.F_{NR_1}$, $BWP.F_{NR_2}$
$BWP.R^{+a}_{NR_3}$, $BWP.R^{+a}_{NR_4}$	$BWP.F_{NR_3}$, $BWP.F_{NR_4}$	$BWP.F_{NR_3}$, $BWP.F_{NR_4}$	$BWP.F_{NR_3}$, $BWP.F_{NR_4}$
$RO.R^{+a}_{brine}$, $RO.TF^{-a}_{fp}$	$RO.P_{memb}$	$RO.P_{memb}$, $RO.P_{rp}$	$RO.P_{memb}$, $RO.F_{permeate}$
$RO.C^{-a}_{memb}$, $RO.R^{+a}_{memb}$	$RO.F_{permeate}$, $\mathbf{BWP.P_{rp}}$	$RO.P_{back}$, $\mathbf{BWP.P_{ocor}}$	$RO.P_{rp}$, $RO.P_{back}$
$RO.R^{+a}_{pipe}$, $RO.GY^{-a}_{rp}$	$RO.P_{rp}$, $RO.P_{back}$	$\mathbf{AES.P_{wick}}$	$\mathbf{BWP.P_{rp}}$
$AES.TF^{-a}_{blower}$, $AES.TF^{-a}_{heater}$	$AES.V_{air}$	$AES.V_{air}$	$AES.V_{air}$
$AES.C^{-a}_{wick}$, $AES.C^{-a}_{steam}$	$AES.P_{wick}$, $AES.P_{steam}$	$AES.P_{wick}$, $AES.P_{steam}$	$AES.P_{wick}$, $AES.P_{steam}$
$AES.C^{-a}_{condensate}$, $AES.R^{+a}_{airFlow}$	$AES.P_{condensate}$, $AES.T_{coolant}$	$AES.P_{condensate}$, $AES.T_{coolant}$	$AES.P_{condensate}$, $AES.T_{coolant}$

Table 9: Results for Experiment 2-A (17 measurements)

Faults	Measurements
$BWP.C^{-a}_{gls}$, $BWP.R^{+a}_{ocor}$	$BWP.P_{gls}$, $BWP.P_{ocor}$
$BWP.R^{+a}_{pipe1}$, $BWP.R^{+a}_{pipe2}$	$BWP.P_{fp}$
$BWP.R^{+a}_{NR_1}$, $BWP.R^{+a}_{NR_2}$	$BWP.F_{NR_1}$, $BWP.F_{NR_2}$
$BWP.R^{+a}_{NR_3}$, $BWP.R^{+a}_{NR_4}$	$BWP.F_{NR_3}$, $BWP.F_{NR_4}$
$RO.R^{+a}_{brine}$, $RO.TF^{-a}_{fp}$	$BWP.P_{rp}$, $RO.P_{memb}$
$RO.C^{-a}_{memb}$, $RO.R^{+a}_{memb}$	$RO.F_{permeate}$
$RO.R^{+a}_{pipe}$, $RO.GY^{-a}_{rp}$	$RO.P_{rp}$, $RO.P_{back}$
$AES.C^{-a}_{wick}$	$AES.P_{wick}$
$AES.C^{-a}_{steam}$, $AES.TF^{-a}_{blower}$, $AES.TF^{-a}_{heater}$	$AES.V_{air}$, $AES.P_{steam}$
$AES.C^{-a}_{condensate}$	$AES.P_{condensate}$
$AES.R^{+a}_{airFlow}$	$AES.T_{coolant}$

Table 10: Results for Experiment 2-B (14 measurements)

Faults	Measurements
$BWP.R^{+a}_{NR_1}$, $BWP.R^{+a}_{NR_2}$ $BWP.C^{-a}_{gls}$, $BWP.R^{+a}_{ocor}$ $BWP.R^{+a}_{pipe1}$, $BWP.R^{+a}_{pipe2}$	$BWP.F_{NR_1}$, $BWP.F_{NR_2}$ $BWP.P_{ocor}$, $BWP.P_{fp}$
$BWP.R^{+a}_{NR_3}$, $BWP.R^{+a}_{NR_4}$ $RO.C^{-a}_{memb}$, $RO.R^{+a}_{memb}$ $RO.R^{+a}_{pipe}$, $RO.GY^{-a}_{rp}$ $RO.R^{+a}_{brine}$, $RO.TF^{-a}_{fp}$ $AES.C^{-a}_{wick}$, $AES.TF^{-a}_{blower}$ $AES.TF^{-a}_{heater}$	$BWP.F_{NR_3}$, $BWP.F_{NR_4}$ $RO.P_{memb}$,$RO.P_{back}$ $RO.P_{rp}$, $AES.P_{wick}$ $AES.P_{steam}$, $AES.V_{air}$
$AES.C^{-a}_{condensate}$, $AES.R^{+a}_{airFlow}$ $AES.C^{-a}_{steam}$	$AES.P_{condensate}$, $AES.T_{coolant}$

We use the bond graph model described above to systematically derive the TCG for the AWRS. The distributed diagnosers are derived from this model using a Python implementation of the design algorithms.

Designing Diagnosers for a Partitioned System

We assume the AWRS to be partitioned into the BWP, RO, and AES subsystems. We run three experiments, for the same fault set (see Table 8), but with different measurement sets. The prefixes *BWP*, *RO*, and *AES*, in Table 8, indicate that the measurement or fault is associated with the BWP, RO, and AES subsystem, respectively.

Experiment 1-A is run with measurements shown in Table 8, column 2. The BWP and AES measurements are sufficient to generate global diagnosis results for these subsystems. However, the RO subsystem diagnoser needs the pressure at the BWP recirculation pump, $\mathbf{BWP.P_{rp}}$, to uniquely isolate all of its faults.

Experiment 1-B uses a measurement set generated by the measurement selection algorithm [94]. These 14 measurements listed in Table 8, column 3, are the minimum number of measurements required to isolate all faults. The diagnosers for the BWP and the AES are the same as in Experiment 1-A. However, the RO diagnoser now needs two external measurements, $\mathbf{BWP.P_{ocor}}$, and $\mathbf{AES.P_{wick}}$, to uniquely isolate all of its single faults.

Experiment 1-C uses 16 measurements (column 4 of Table 8). Like Experiment 1-A, only $\mathbf{BWP.P_{rp}}$ needs to be communicated to the RO for complete diagnosability. This shows that the extra measurement in Experiment 1-A provides no additional diagnostic information.

The derived local diagnoser structures match our intuition. Comparing the results of the experiments with 14 measurements to that with 16 measurements, it is clear that additional measurements provide more redundancy of information, and make the diagnosers more independent. The trade-off between the cost of additional sensors versus greater communication overhead and independence of the local diagnosers is evident.

Designing Diagnosers for an Unpartitioned System

For the case where we did not assume any subsystem information, we again ran three experiments for the measurement sets and faults listed in Table 7.

Experiment 2-A to 2-C produced 11, 3 and 4 local diagnosers, respectively (see Tables 9-11).

It is clear that additional measurements increases redundancy, which Algorithm 2 exploits to create smaller diagnosers. Tables 9 and 10 results show that the balance heuristic works well. The Table 11 result is different, because the algorithm derived one large, one medium, and two very small diagnosers. A different set of 16 measurements would very likely have produced a more balanced result.

Comparing the results of the experiments with 14 measurements, the partition structure created by Algorithm 2 is found to be similar to that generated by Algorithm 1, even though Algorithm 2 rearranges the faults and measurements between the diagnosers to ensure that less measurements are needed for each diagnoser. For the experiments with additional measurements, Algorithm 2 tends to use the redundant information to create a larger number of smaller diagnosers, to improve the overall computational efficiency.

Table 11: Results for Experiment 2-C (16 measurements)

Faults	Measurements
$BWP.R^{+a}_{NR_1}$, $BWP.R^{+a}_{NR_2}$ $BWP.R^{+a}_{NR_3}$, $BWP.R^{+a}_{NR_4}$ $BWP.C^{-a}_{gls}$, $BWP.R^{+a}_{ocor}$ $BWP.R^{+a}_{pipe1}$, $BWP.R^{+a}_{pipe2}$ $RO.R^{+a}_{brine}$, $RO.R^{+a}_{pipe}$ $RO.GY^{-a}_{rp}$, $RO.TF^{-a}_{fp}$ $AES.C^{-a}_{condensate}$	$BWP.P_{rp}$, $BWP.P_{ocor}$ $BWP.P_{fp}$, $BWP.F_{NR_1}$ $BWP.F_{NR_2}$, $BWP.F_{NR_3}$ $BWP.F_{NR_4}$, $RO.P_{rp}$ $RO.F_{back}$, $AES.P_{condensate}$
$RO.C^{-a}_{memb}$	$RO.P_{memb}$
$RO.P^{+a}_{memb}$	$RO.F_{permeate}$
$AES.C^{-a}_{steam}$, $AES.R^{+a}_{airFlow}$ $AES.C^{-a}_{wick}$, $AES.TF^{-a}_{blower}$ $AES.TF^{-a}_{heater}$	$AES.P_{wick}$, $AES.P_{steam}$ $AES.T_{coolant}$, $AES.V_{air}$

Table 12: Some fault signatures for the AWRS diagnosis experiment

Fault	$BWP.$ P_{ocor}	$RO.$ P_{memb}	$RO.$ P_{back}	$RO.$ P_{rp}	$AES.$ P_{wick}
$BWP.C^{-a}_{gls}$	0−	0+	0−	0+	0+
$RO.C^{-a}_{memb}$	0−	+−	+−	0−	0+
$RO.R^{+a}_{memb}$	0−	0+	0+	0−	0+
$RO.R^{+a}_{brine}$	0+	0+	0+	0−	0+
$RO.R^{+a}_{pipe}$	0−	0−	0−	0−	0−
$RO.TF^{-a}_{fp}$	0+	0−	0+	0−	0+
$AES.TF^{-a}_{blower}$	0+	0−	0−	0−	0−
$AES.TF^{-a}_{heater}$	0+	0−	0−	0−	0−

Distributed Fault Isolation

We illustrate the online operation with one set of distributed diagnosers. We show how the local diagnosers generated in Experiment 1-B isolate a block in the pipe ($RO.R_{pipe}$) that connects the tubular reservoir to the membrane in the RO subsystem. The three local diagnosers are implemented as described above.

For this demonstration, we use a MATLAB® Simulink® simulation model of the AWRS that was systematically derived from the bond graph models described in Section VI-A [89]. The fault, modeled as a 20% abrupt increase in the $RO.R^{+a}_{pipe}$ pipe resistance, is introduced at time $t = 21,000$ seconds. The simulation is run for 86,400 simulation seconds. Measurement noise is Gaussian with a noise power level set at 2% of the average signal power for each measurement. The measurements are sampled at 1 Hz. Table 12 gives some of the relevant fault signatures for this experiment.

The diagnosis steps are shown in Table. 13. A block causes decreased flow through the pipe initially. As a result, $RO.P_{memb}$, the pressure in the membrane, decreases, but not discontinuously (0−). The deviation in $RO.P_{memb}$ is first detected by the RO diagnoser. The candidate set, at

Table 13: Diagnosis results for 20% abrupt fault $RO.R^{+a}_{pipe}$ at $t_f = 21000$ seconds.

Step	Symbols	$D_{(F_1,\ \widetilde{M_1})}$ (BWP) Candidate set	$D_{(F_2,\ \widetilde{M_2})}$ (RO) Candidate set	$D_{(F_3,\ \widetilde{M_3})}$ (AES) Candidate set
0	$RO.P_{memb}:\ (0-)$	No measurement deviation detected	$RO.R^{+a}_{pipe}$, $RO.TF^{-a}_{fp}$	No measurement deviation detected
1	$RO.P_{back}:\ (0-)$	No measurement deviation detected	$RO.R^{+a}_{pipe}$	No measurement deviation detected
2	$RO.P_{rp}:\ (0-)$	No measurement deviation detected	$RO.R^{+a}_{pipe}$	No measurement deviation detected
3	$AES.P_{wick}:\ (0-)$	No measurement deviation detected	$RO.R^{+a}_{pipe}$	$AES.TF^{-a}_{blower}$, $AES.TF^{-a}_{heater}$
4	$BWP.P_{ocor}:\ (0-)$	$BWP.C^{-a}_{gls}$	$RO.R^{+a}_{pipe}$	$AES.TF^{-a}_{blower}$, $AES.TF^{-a}_{heater}$

this time, includes $RO.P^{+a}_{pipe}$, and a decrease in the RO feed pump efficiency, $RO.TF^{+a}_{fp}$, the only faults whose fault signatures are consistent with the observed (0−) change. Subsequently, measurement $RO.P_{back}$, i.e., the pressure in the RO loop also deviates as (0−). The fault signature of $RO.TF^{-a}_{fp}$ for this measurement is not consistent with this change and hence this fault is dropped from the candidate list. At this point, $RO.R^{+a}_{pipe}$ is the only fault candidate, but all measurements of $D_{(F_2,\ \widetilde{M_2})}$ have not deviated, therefore, we cannot be sure that we have the final diagnosis result. The measurement deviation, (0−), in $RO.P_{rp}$ is consistent with the candidate. The fourth measurement deviation observed, is a drop in the pressure in the wick reservoir, i.e., $AES.P^{+a}_{wick}$. The observed deviation (0−) continues to be consistent with the $RO.R^{+a}_{pipe}$ fault candidate. Since this measurement is also accessible to AES, it triggers the fault diagnoser $D_{(F_3,\ \widetilde{M_3})}$ and generates the fault candidate set of size 2. Finally, when $BWP.P_{ocor}$ is observed to deviate as (0−), diagnoser $D_{(F_1,\ \widetilde{M_1})}$ is initiated with a single fault, $BWP.C^{-a}_{gls}$, in the candidate set. Since *all* measurements of $D_{(F_2,\ \widetilde{M_2})}$ have deviated, and it has one fault candidate remaining, the system supervisor declares $RO.R^{+a}_{pipe}$ as the true fault, and this corresponds to the correct global diagnosis. The plots for the measurement deviations are shown in Fig. 28.

Summary and Conclusions

In this chapter, we presented a novel model-based distributed diagnosis approach, where local diagnosers generate globally correct local diagnosis results, with minimal exchange of information, and no coordination. Since no coordination is required, the computational complexity of the overall diagnosis task is significantly reduced. Moreover, minimal exchange of information also guarantees reduction in communication overhead. We proposed two approaches to design distributed diagnosers. In the first approach, we assumed knowledge of subsystem structure, especially the measurements and faults that belong to each subsystem, and based on this information, we designed a local diagnoser for each subsystem such that it required minimal number of additional *external* measurements to diagnose *all* the faults assigned to that subsystem. In the second approach, we assumed no prior partitioning information. Instead, we generated the maximal number of distributed diagnosers, such that, each local diagnoser could operate independently without sharing measurements.

In literature, model-based diagnosis approaches have been broadly classified into *centralized*, *decentralized*, and *distributed* schemes (e.g., [2,22,100,101]). Centralized schemes (e.g., [2]), construct a single diagnoser from a global system model. Decentralized schemes, such as [9], use a global system model but distribute the diagnosis computations among several local diagnosers. The local

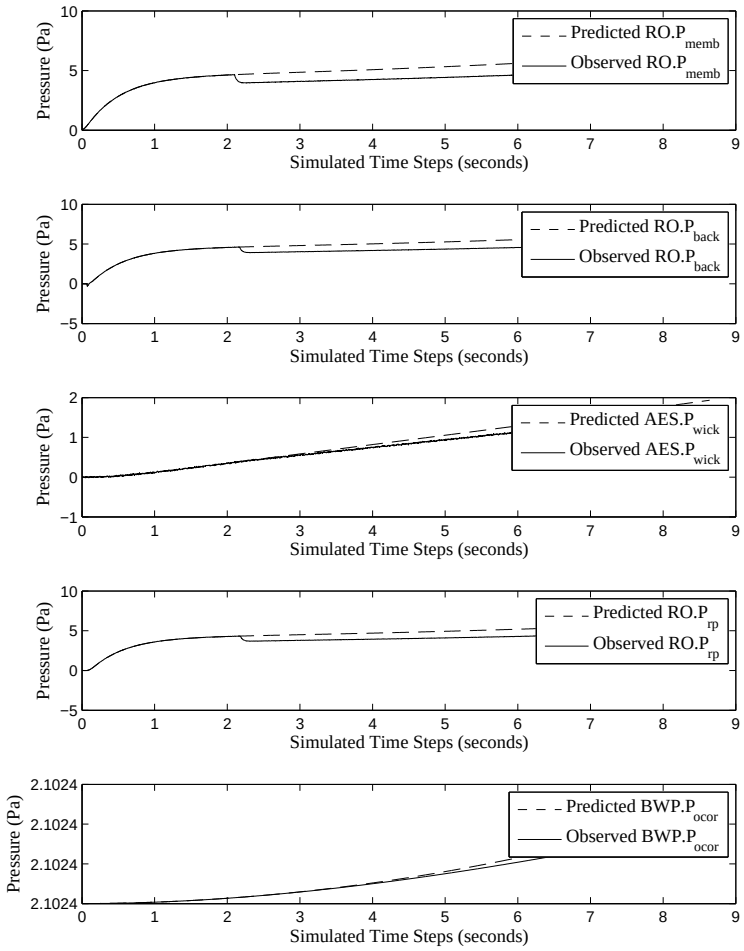


Figure 28: Experimental observations.

diagnosis decisions based on a subset of observations are communicated to other diagnosers, or to a central coordinator, which use the global model to generate globally consistent solutions. Distributed diagnosis approaches use subsystem models and assume the global model is unknown [102–104]. Local diagnosers for each subsystem communicate their diagnosis results to each other to arrive at the global solution.

Most decentralized and distributed diagnosis algorithms have been developed in the discrete-event framework [9, 10, 102–105]. In [9], the authors discuss three coordinated decentralized protocols for diagnosis that extend the centralized diagnosis method developed in [5]. Each local diagnoser is built from the global system model and uses only a subset of observable events. Coordination is necessary in the first and second protocols to generate the correct diagnosis result, but the third protocol generates correct results without a coordinator. All three protocols, under certain assumptions, produce the same results as a centralized diagnoser.

The approaches presented in [105] and [10] avoid coordination between local diagnosers by representing the system as a network of communicating finite state machines. First, the observable events for each subsystem are used to generate the individual subsystem diagnoses. Then, the subsystem diagnoses are merged to generate the global diagnosis result. The offline approach presented in [105] assumes all observable events are received in the same order that they were transmitted. The online approach described in [10] achieves efficiency by avoiding merge operations for independent subsystems. Its incremental algorithm does not make the assumption that the ordering of observations is preserved.

In [102], the authors describe an approach where each local diagnoser generates a set of local diagnoses, and then communicates with its neighbors to reduce the number of hypotheses. The graph of constraints between the fault hypotheses and the observations is partitioned to minimize communication between local diagnosers. A similar approach is presented in [103], where the partitioning is based on physical connections.

Our approach, designed for diagnosing faults in large continuous systems, differs from [9,10,102–105]. Abrupt parametric faults, i.e., a step change in a plant parameter value, produce transients in the system dynamics. Capturing these fault-generated transient behaviors in a discrete-event model by quantizing the measurement or state-space can result in state explosion [35]. We adopt a different approach, where we use the continuous model to derive fault effects as qualitative magnitude and higher-order effects on individual measurements. This produces a compact model for online diagnosis.

We use the global system model to design local diagnosers offline. At runtime, the local diagnosers operate independently to generate local diagnosis results that are globally correct. Our approach does not require a coordinator, there is minimal or no exchange of information among the diagnosers, and no partial diagnosis result is exchanged between the diagnosers. This is similar to the third protocol in [9], and a failure in a local diagnoser does not affect the diagnosis capability of the other diagnosers. Therefore, our approach operates like other online distributed diagnosis schemes (e.g., [10]).

In both the centralized and distributed qualitative diagnosis approaches we have presented thus far, the generation of incipient and abrupt additional fault hypotheses, with similar fault signatures leads to a loss of diagnosability, since the incipient and abrupt fault candidates may have the same qualitative effect on the measurements. Hence, it is not always possible to refine the fault hypothesis set to a singleton based on the qualitative information alone. In this chapter, we have considered these indistinguishable faults to be "aggregate faults". In the next chapter, we propose a

combined qualitative-quantitative Bayesian diagnosis approach that combines the TRANSCEND qualitative fault isolation scheme with Dynamic Bayesian network-based parameter and state estimation approach to better discriminate between abrupt and incipient faults, and other aggregate faults, in a computationally efficient manner. This combined qualitative-quantitative diagnosis scheme is based on probabilistic reasoning techniques, and hence is robust to uncertainties, caused by sensor noise. In Chapter VI, we distribute this centralized qualitative-quantitative approach to further improve the computational efficiency, and address the other drawbacks inherent in the centralized Bayesian scheme.

CHAPTER V

CENTRALIZED BAYESIAN DIAGNOSIS OF COMPLEX SYSTEMS

In Chapter III, we presented the TRANSCEND qualitative diagnosis scheme for isolating abrupt and incipient faults in continuous, dynamic systems [7, 83, 95, 106]. We saw that TRANSCEND's fault isolation approach suffers from the ambiguity problem, i.e., the inability to uniquely isolate the true fault hypothesis from a set of fault hypotheses due to the lack of discriminatory ability of the qualitative fault signatures. Quantitative estimation schemes produce more precise diagnoses, and can be made robust to uncertainties. But, for large systems with complex dynamics, these quantitative approaches are computationally very expensive for online analysis. Besides, these approaches may also suffer from convergence problems [3, 4].

This chapter addresses this ambiguity problem of the qualitative fault isolation scheme (Qual-FI) of TRANSCEND and improves its robustness to uncertainties by combining the TRANSCEND Qual-FI scheme with a stochastic, Dynamic Bayesian Network (DBN)-based state and parameter estimation method for quantitative fault hypotheses refinement and identification (Quant-FHRI) in dynamic systems. The DBN-based particle filtering scheme [12, 77] is employed to accommodate nonlinear systems with non-Gaussian probability distributions. However, DBN-based diagnosis approaches are computationally expensive, mainly because these approaches use a single DBN model that include *all* possible fault hypotheses as different random variables, which results in a the DBN having a large number of nodes (i.e., system variables and possible fault hypotheses) [58]. We reduce the computational expense of the quantitative scheme by employing our qualitative fault signature-based isolation methodology to reduce the number of potential hypotheses based on the observed measurement deviations, creating a *separate* DBN model for each fault hypothesis in the reduced hypotheses set, and then applying the quantitative scheme on these smaller DBN fault models in parallel, thus providing considerable computational savings.

Fig. 29 presents the computational architecture of our combined qualitative-quantitative Bayesian diagosis scheme. In this work, the extended Kalman filtering (EKF)-based observer used for tracking nominal system behavior in TRANSCEND (see Chapter III) is replaced by a DBN-based observer because it is more general than the EKF approach. We adopt the particle filtering scheme for state estimation using DBNs. For diagnosis, we start by first generating the DBN model of the nominal system automatically from its temporal causal graph (TCG). Recall that the TCG itself can be

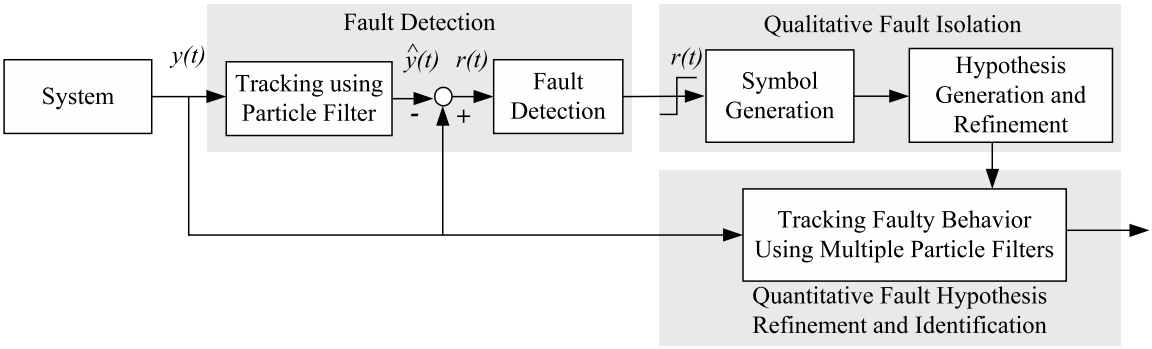


Figure 29: The diagnosis architecture.

automatically generated from the system bond graph. The DBN-based observer is then invoked to track the nominal system model to track nominal behavior. The difference between nominal measurement estimates and the actual observations define the residual signals that are used by the fault detectors to detect statistically significant non-zero residual values.

Detection of a fault triggers the extended TRANSCEND Qual-FI scheme (for isolating both incipient and abrupt faults) and possible fault hypotheses that explain the observed measurement deviations are generated. These hypotheses are pruned as additional measurement deviations occur. Once the number of fault hypotheses is refined to less than a user-specified lower bound, or the fault hypotheses set cannot be reduced any further, the Quant-FHRI scheme is invoked to identify the true fault hypotheses, and if possible, further refine the fault hypotheses set. If the Qual-FI scheme refines the fault hypotheses set to a singleton, the Quant-FHRI scheme performs the function of fault identification only.

The first step of Quant-FHRI is the generation of a faulty DBN for each remaining fault hypothesis, typically done by modifying the nominal DBN model to include the faulty parameter as a stochastic variable in the DBN [107]. Under the single fault assumption, a separate DBN-based observer is then invoked for each fault hypotheses model to track the observed measurement values using a particle filtering scheme. If the estimated measurements significantly deviates from the observed faulty measurements, that corresponding fault hypothesis is deemed inconsistent, and removed from the set of fault hypotheses. Ideally, only the particle filter estimator using the true fault model produces measurement value estimates that converge to the observed faulty measurement values. A side-effect of this process is the estimation of the fault parameter value. Also, as the Qual-FI scheme (which is still running) continues to refine its fault hypotheses, the particle filters corresponding to the inconsistent fault models are terminated. This pruning of inconsistent fault

hypotheses based on the qualitative and quantitative analysis of measurements helps in the efficient and accurate diagnosis of the true incipient or abrupt fault.

In Chapter III, we presented the diagnosability analysis of TRANSCEND's Qual-FI approach based on the available fault signatures. The diagnosability property of our Bayesian diagnosis scheme, however, also depends on the correct estimation of the faulty parameter by the Quant-FHRI scheme. Correctness of state estimates is guaranteed if a system is *observable*. Classical techniques for determining if a system is observable require the knowledge of parameter values, and these approaches are usually not applicable to nonlinear systems [88]. *Structural observability* depends on the system structure, and not on parameter values, and can be systematically determined by analyzing the bond graph of nonlinear systems [14]. Hence, in this chapter, we describe the property of structural observability, and how it can be analyzed using bond graphs.

The rest of this chapter is organized as follows. The next section discusses our systematic approach for deriving the nominal DBN models from the system TCG. Then, the following sections present the different steps of the combined qualitative-quantitative diagnosis schemes. Since the fault detection and Qual-FI are essentially the same as those used in TRANSCEND, but implemented using a particle filtering-based observer, we briefly present these approaches before presenting the Quant-FHRI approach in detail. We propose three different approaches for fault identification that are aimed at addressing the issue of sample impoverishment inherent in particle filtering schemes applied to fault diagnosis [4]. Next, we present a discussion on structural observability, and how the system bond graph can be analyzed to determine the existence of structural observability properties in a system. We then present a set of experimental results that demonstrate the effectiveness of our combined qualitative-quantitative Bayesian diagnosis scheme over the extended TRANSCEND diagnosis approach. Finally, we conclude the chapter with a discussion on how this Bayesian diagnosis approach compares to other particle filtering-based diagnosis. The research contributions presented in this chapter have been published in [84, 85].

Deriving Dynamic Bayesian Networks For Complex Systems

This section presents the systematic procedure for generating the nominal DBN model of a system from its temporal causal graph. The procedure for generating DBN fault models by modifying the nominal system DBN is presented in a later section of this chapter. In the remainder of this section, we use the nonlinear two-tank system shown in Fig. 12(a) as an illustrative example.

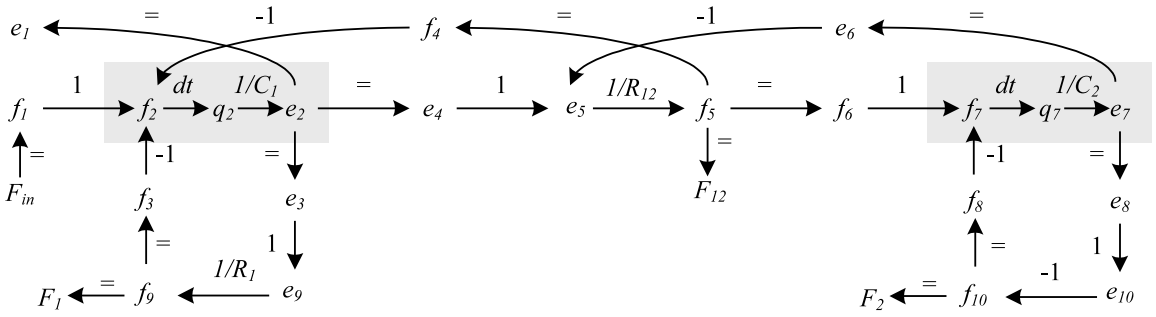


Figure 30: Two tank TCG containing displacement variables q_2 and q_7.

The DBN of a system can be systematically constructed from its TCG using a three-step procedure: (*i*) for every effort (respectively, flow) variable associated with a C-element (respectively, I-element) in integral causality, insert a corresponding displacement (respectively, momentum) variable in the system TCG, (*ii*) "simplify" this TCG so that it contains the state, measured, and input variables only, and (*iii*) construct the system DBN from this simplified TCG.

In Chapter III, we showed how the nominal state-space equations obtained from bond graphs include as state variables, the displacement and momentum variables, denoted by $q_i = \int f_i dt = Ce_i$, and $p_i = \int e_i dt = If_i$, respectively, where e_i and f_i denote the effort and flow variables, respectively, that are associated with bond i. In the DBN models, p_i and q_i variables also represent the state variables with the across time links. However, recall that in a TCG, a node denotes either an e_i or f_i variable. Since, we derive the nominal system DBN from its TCG, we need to include these displacement and momentum variables in the system TCG before deriving the corresponding DBN. The inclusion of displacement and momentum variables involves replacing every $e_i \xrightarrow{dt/I} f_i$ link in a TCG with the $e_i \xrightarrow{dt} p_i \xrightarrow{1/I} f_i$ TCG fragment. Similarly, every $f_i \xrightarrow{dt/C} e_i$ link in a TCG with the $f_i \xrightarrow{dt} q_i \xrightarrow{1/C} e_i$ TCG fragment. This transformation of the TCG follows directly from the definition of the p and q variables [18].

Example. Consider the two-tank TCG shown in Fig. 16 that contains only effort and flows as its nodes. Fig. 30 shows the two-tank TCG with the variables q_2 and q_7 are inserted in between the variables $f_2 - e_2$ and $f_7 - e_7$, respectively, where q_2 and q_7 denote the displacement variables in tanks C_1 and C_2, respectively. As a result, the edges $f_2 \xrightarrow{dt/C_1} e_2$ and $f_7 \xrightarrow{dt/C_2} e_7$ are replaced with pairs of edges, $f_2 \xrightarrow{dt} q_2 \xrightarrow{1/C_1} e_2$ and $f_7 \xrightarrow{dt} q_7 \xrightarrow{1/C_2} e_7$, respectively.

Once the appropriate p and q variables have been inserted into the system TCG, this TCG is simplified using the approach presented in [108] that allows TCG abstractions to be generated by

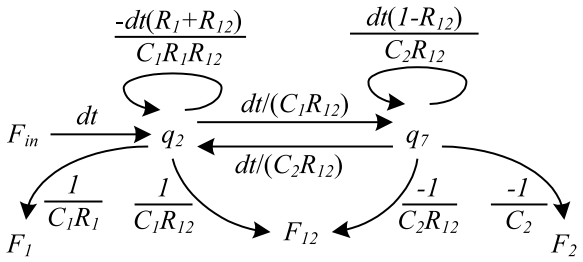


Figure 31: Simplified Temporal Causal Graph of the two-tank system.

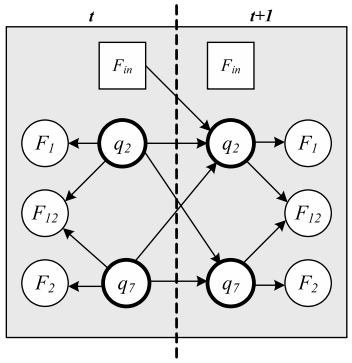


Figure 32: Dynamic Bayesian network of a two-tank system.

iteratively deleting vertices from the detailed TCG. While the simplified TCG contains fewer nodes, it retains the causal and temporal information present in the detailed TCG. Fig. 31 shows the simplified TCG generated from the detailed TCG shown in Fig. 16. Notice how this simplification results in more complex edge labels than in the detailed TCG. Hence, we cannot easily implicate individual parameter deviations as possible causes of measurement deviations using this simplified TCG. However, the objective for generating the simplified TCG is only to enable the systematic and automatic generation of DBNs from the TCG, and not for qualitative isolation. The qualitative isolation is based on the detailed TCG. Hence, the complex edge labels are acceptable in the simplified TCG.

The system DBN can be constructed from this simplified TCG using the method outlined in [58]. The set of nodes, $\mathbf{N}$, in the simplified TCG includes only state variables, measured variables, and system inputs. For each $N \in \mathbf{N}$, we instantiate nodes N_t and N_{t+1} in the consecutive time slices of the DBN. Then, for every pair of variables, $N, N' \in \mathbf{N}$ that are algebraically related, causal links $N_t \rightarrow N'_t$ and $N_{t+1} \rightarrow N'_{t+1}$ are constructed in each DBN time slice. For every pair of variables, $N, N' \in \mathbf{N}$ having an integrating relation (i.e., a delay), the across-time $N_t \rightarrow N'_{t+1}$ link is added to the DBN.

Example. We now describe the procedure for generating the two-tank DBN shown in Fig. 32 from the two-tank TCG shown in Fig. 31. For every node, N, in the simplified TCG, we instantiate two nodes, N_t and N_{t+1}, e.g., we instantiate nodes F_{in_t}, $F_{in_{t+1}}$, q_{2_t}, $q_{2_{t+1}}$, and so on, based on the simplified TCG shown in Fig. 31. In this simplified TCG, F_{12} is algebraically related to q_2 and q_7. Hence, we draw causal links $q_{2_t} \rightarrow F_{12_t}$, $q_{7_t} \rightarrow F_{12_t}$, $q_{2_{t+1}} \rightarrow F_{12_{t+1}}$, and $q_{2_{t+1}} \rightarrow F_{12_{t+1}}$ in the two-tank DBN. Similarly, we have links $q_{2_{t+1}} \rightarrow F_{1_{t+1}}$, and $q_{7_{t+1}} \rightarrow F_{2_{t+1}}$, involving the two other measurement variables, F_1 and F_2. All other edge labels of the simplified TCG contains the dt label indicating an integrating relation, e.g., the edge $F_{in} \xrightarrow{dt} q_2$. Hence, we draw a corresponding inter-time-slice causal link, $F_{in_t} \rightarrow q_{2_{t+1}}$, in the DBN. Similar edges are drawn in the TCG for all integrating edges in the simplified TCG. The two-tank DBN shown in Fig. 32 can be represented as $D = (\{q_2, q_5\}, \{F_{in}\}, \{F_1, F_{12}, F_2\})$, with $\mathbf{X} = \{q_2, q_5\}$, $\mathbf{U} = \{F_{in}\}$, and $\mathbf{Y} = \{F_1, F_{12}, F_2\}$.

While the above approach generates the structural form of the DBN based on the appropriate TCG, the conditional probability density functions, are obtained from the system's state space equations (in difference form). For example, the conditional probability density functions in the two-tank DBN can be obtained from the state space equations in difference form shown in equations 13 and 14, with $\Delta t = 1$.

Example. It is well known that if a random variable U is the linear combination of two other normal random variables, $V \sim N(\mu_V, \sigma_V^2)$ and $W \sim N(\mu_W, \sigma_W^2)$, such that $U = aV + bW$, then $U \sim N(a\mu_V + b\mu_W, a^2\sigma_V^2 + b^2\sigma_W^2)$ is also normally distributed. Hence, given the state-space equations of the two tank system, we have $P(F_1(t)|q_2(t)) = N(\mu_{q_2}(t)/(C_1R_1), \sigma_{q_2}^2/(C_1^2R_1^2))$. However, it is not always possible to obtain closed-form, analytical representations of the conditional probability. For example, since resistor R_2 of the two-tank system is nonlinear, and depends on the pressure in tank 2, i.e., $R_2(t) = 3e_7(t)^2 = \frac{3q_7(t)^2}{C_2^2}$, we cannot obtain a closed form representation of $P(q_7(t+1)|q_2(t), q_7(t))$ as $q_7(t+1) = \frac{q_2(t)}{C_1R_{12}} + q_7(t) - \frac{C_2}{3q_7(t)} - \frac{q_7(t)}{C_2R_{12}}$. This justifies our choice of adopting particle filters. Similarly, $P(F_2(t)|q_7(t))$ does not have a closed form analytical representation since $F_2(t) = \frac{C_2}{3q_7(t)}$, with the normal random variable, q_7, in the denominator.

We now present the different steps involved in our Bayesian diagnosis scheme shown in Fig. 29, and show how the DBN models we derive from the system TCG are used in every step of our diagnosis approach.

Fault Detection and Qualitative Fault Isolation

The fault detection and qualitative fault isolation schemes invoked as part of our combined qualitative-quantitative diagnosis approach is essentially the same as that used in the extended TRANSCEND diagnosis approach described in Chapter III. The only difference is the use of the DBN-based observer instead of the extended Kalman filter. The nominal DBN model includes nodes corresponding to state variables, observed variables, and inputs only. The system component parameters are either constant, or possibly nonlinear, deterministic algebraic functions of other state variables. The conditional probability density functions in the DBN may or may not have a closed-form analytic representation.

A fault is detected when the detector reports that at least one measurement residual is significantly different from 0. Once a fault is detected, the Qual-FI scheme is triggered to generate the initial fault hypotheses and refine these hypotheses as additional measurement deviations are observed. The Quant-FHRI scheme is triggered when any of the following four conditions are satisfied:

1. All measurements in the system have been observed to deviate from nominal, and hence, the fault hypotheses set cannot be refined any further by the Qual-FI scheme.

2. Not all measurements have been observed to deviate, but the set of fault hypotheses cannot be refined any further using the Qual-FI scheme. The diagnosability analysis of TRANSCEND (discussed in Chapter III), or, alternatively, related measurement orderings [1, 109], can be used to determine if the set of fault hypotheses cannot be refined any further.

3. The fault hypotheses set is refined to a pre-defined size, k, a design parameter. In our work, we typically set k to 10% of the total number of fault hypotheses generated after a fault is detected.

4. A pre-specified s simulation timesteps have elapsed, after which the Quant-FHRI scheme is invoked to refine the hypothesis set further, and also identify the true fault.

Therefore, the maximum time after which the Quant-FHRI scheme is initiated is $t_d + s$ time steps, where t_d is the time of fault detection. However, on average, we have noticed that the Quant-FHRI is invoked earlier than time $t_d + s$, and mostly depends on our choice of k. We need to choose k and s carefully because if k is too large and s is too small, the large number of remaining fault candidates would make the Quant-FHRI inefficient. On the other hand, if k is very small, and s is large, the isolation and identification task will be delayed. In the following, we describe the Quant-FHRI scheme in greater detail.

Quantitative Fault Isolation and Identification

Fig. 33 shows the architecture of our Quant-FHRI scheme, which performs both fault isolation and identification tasks. However, if the Qual-FI schemes isolates the true single fault hypothesis, Quant-FHRI performs the task of fault identification only. The first step of Quant-FHRI is to develop a DBN model for each fault hypothesis, $\phi_j \in \{\phi_1, \phi_2, \ldots, \phi_w\}$ that remain when Quant-FHRI is initiated. The goal of Quant-FHRI is to estimate the value of the hitherto unknown fault parameter, and hence the DBN model must facilitate the estimation of this faulty parameter. As we discuss later in the chapter, depending on how the faulty parameter is included in the DBN model, we could have a separate DBN model, DBN_i, for each ϕ_i, or a single DBN_i could model multiple fault hypotheses, e.g., a DBN that includes a parameter, p, as a state variable can be used to represent both $p^{\pm a}$ and $p^{\pm i}$ fault hypotheses. Hence, the w fault hypotheses are modeled by v DBNs, where $v \leq w$. Next, for each DBN_i, we use a separate DBN-based observer implemented using a particle filtering scheme, PF_i, to estimate the augmented state variables that includes the fault hypothesis. We can instantiate a separate particle filter estimator for each hypotheses since we assume only single faults occur in the system, and these faults can be independent. These estimates are compared with the actual observed measurements to generate measurement residuals for each fault model. A Z-test is then invoked on the residuals obtained from each particle filter estimator to test for statistically significant divergence of measurement estimates from the actual measurements. Since, even the correct fault model will need some time before the particles start converging to the observed measurement values, we expect the measurement estimates to diverge at the point of fault occurrence for a few time steps before it eventually converges to the the observed measurements. We typically assume that the measurement estimates obtained using the true fault model will converge to the observed measurements within s_d time steps from the time the divergence was detected. Hence, we declare that a fault model is inconsistent when the Z-test indicates that the measurement estimates obtained using the fault model does not converge within s_d time steps from when the estimates diverged. Ideally, after a finite number of time-steps, only the measurement estimates obtained using the correct fault model should converge to the observed values of the measurements. Depending on the magnitude of the fault, and which parameter is faulty, it might not be possible for Quant-FHRI to refine the possible fault hypotheses to a singleton set. In that scenario, we report all remaining fault hypotheses as declared equally probable. If multiple fault hypotheses are modeled by the same DBN, additional post processing of the converged estimates is required to further discriminate between the different fault hypotheses this DBN models. Finally, the output of

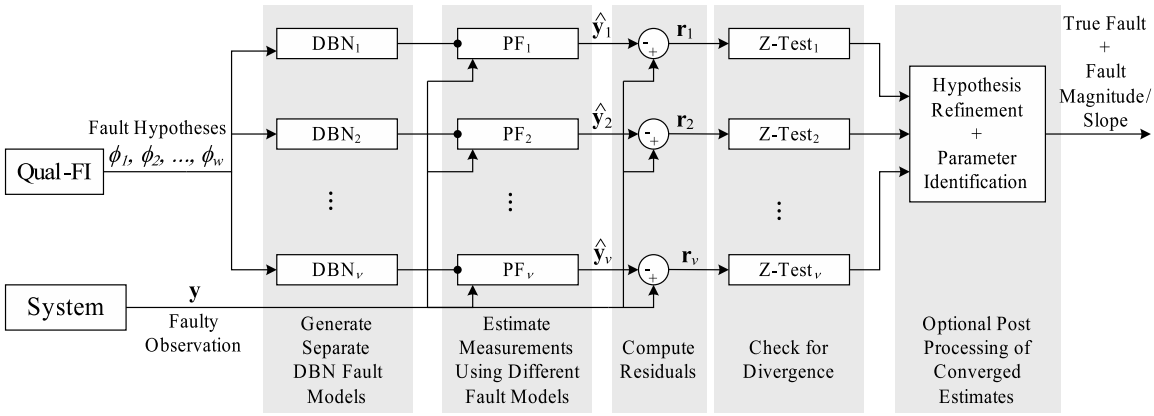


Figure 33: Computational Architecture of Quant-FHRI.

our Quant-FHRI scheme would be the true fault parameter and the magnitude, or slope, with which this parameter has changed. Note that the Qual-FI scheme is not terminated once the Quant-FHRI scheme is invoked. Hence, our combined qualitative-quantitative diagnosis scheme drops a fault candidate if: (i) the Qual-FI drops that fault candidate, or (ii) the measurements estimated by that fault model significantly deviates from the observed faulty measurements.

In Chapter II, we showed that particle filtering schemes used for fault diagnosis suffer from the *sample impoverishment* [4, 13] problem. The main idea is that the transition probability to fault states is typically very low, and hence particles are unlikely to cover the fault state mode during the prediction stage of a particle filter. Further, even if some particles do get instantiated correctly in the vicinity of the faulty state, the particles have very low weights to begin with, and these particles are very likely to be dropped during the resampling phase of the particle filtering. Several different schemes for addressing this issue have been suggested. In [13], the faults are denoted by discrete fault modes, and the authors suggest increasing the number of particles in the particle filter to ensure that the discrete state space is covered better. However, this may not be a feasible idea for online diagnosis, since particle filtering is an "any-time" algorithm, with its computational expense being proportional to the number of particles it uses [13]. Moreover, the faults can be represented as a finite set of discrete fault modes which make the sampling process easier. Another solution was to have an oracle tell the particle filter what are the most likely set of fault modes, so that, during resampling, there are *always* some particles in the states provided by the oracle. Other approaches to addressing the sample impoverishment include risk-sensitive particle filters (RSPFs) and variable-resolution particle filters (VRPFs) [4]. In RSPF, high costs are assigned to fault states, and by including a cost-model during generating particles, RSPFs ensure that particles are generated to represent faults, even if these particles have low weight. In variable-resolution particle filters, multiple similar

states are tracked by a single abstract particle, allowing a limited number of particles to sufficiently represent large portions of the state-space when the likelihood of occupying that part of the state-space is low. When the likelihood of the grouped state increases, the abstract particles are refined to represent individual states. In RSPF and VRPF, again, the faults are represented as discrete with only a finite number of possible values.

In our fault diagnosis scheme, faults are represented by continuous random variables, and not discrete fault modes. Hence unlike the above mentioned approaches, our faulty parameters cannot be restricted to have a finite set of possible values. Hence, it is not feasible for us to ensure that after resampling, particles representing every possible value of the fault parameter are always present, as proposed in [13]. A similar argument can be made to explain why RSPFs and VRPFs are not applicable to our particular problem.

Once a fault has occurred, and till the magnitude of the fault is correctly identified, the system model is unknown to us. Since we do not know what value the faulty parameter is after the fault has occurred, we face a more fundamental problem of what the variance for this unknown faulty parameter should be in order to ensure that enough support points (or samples) are generated in the vicinity of its correct, yet faulty value. Another issue we face is the setting of the initial conditions. Therefore, if we started tracking faulty measurements from time t_d onwards, our tracking will be off, since we would possibly start at wrong initial state, especially for the unknown fault variable. To overcome this problem, we start our tracking of measurements from time $t_d - \Delta_t^{max}$, where $\Delta_t^{max} \geq t_d - t_f$ is the maximum delay possible between the time of occurrence of fault, t_f, and the time of its detection, and the initial state vector is set to the estimated values of the state variables obtained from the nominal particle filter at time $t_d - \Delta_t^{max}$. The idea is that if Δ_t^{max} is set to be at least as large as $t_d - t_f$, then $t_d - \Delta_t^{max} \leq t_f \leq t_d$, and our initial state setting will be closer to the correct value, since it is based on state estimates from a known system model. The values of the observed measurements are stored from $t_d - \Delta_t^{max}$, and used by the particle filters using the different fault hypotheses. Δ_t^{max} is another design parameter, and determined empirically based on how well the fault detectors have been tuned. Even if the fault detector is tuned well, sometimes, the dynamics of the system results in the fault detector to take more than Δ_t^{max} time steps to detect the fault, i.e., $t_d - t_d > \Delta_t^{max}$. In such scenarios, even if we start tracking measurements from $t_d - \Delta_t^{max}$, our estimates of the initial state would be very close to the actual initial states at that time, since the deviation in the system state is relatively small in the first place, to have resulted in this delayed detection.

Tracking the observed measurements from before the occurrence of fault in the system is beneficial in terms of setting the initial state vector values. However, this implies that the process noise for each state variable must be set large enough to allow the generation of enough particles in the vicinity of the true faulty parameter value. Fig. 34 illustrates how our standard deviation is defined around the approximate time of fault occurrence. Recall that we introduce the faulty parameter as an additional state variable in the DBN fault model. Let us consider the case where we need to identify an abrupt fault in parameter p. Since our assumption is that the fault could have occurred any time between $t_d - \Delta_t^{max}$ and t_d, we have to have a large enough variance, σ_p^2, to allow the generation of particles near the now faulty value of p. If we use a small variance instead, the particle filter will have a weak support, and it is unlikely the faulty parameter will be estimated correctly. However, a constant large variance would result in a large variance in the estimated faulty parameter value, which can be made up for by using a large number of particles. But, the larger the number of particles used, the more computationally expensive the tracking process becomes, which is not desirable for our online fault diagnosis scheme. Hence, we have adopted an approach of starting with a high value for the standard deviation, σ_p^{max} and gradually reducing it linearly to minimum σ_p^{min} value over a period of r seconds. A very large r implies a slow decrease in σ_p, which would result in a large variance in the estimates of the faulty parameter. In contrast, a very small r implies a fast decreasing variance, which might not allow the generation of a strong support for the particle filter. The decision about how fast the variance should be decreased is taken through an empirical study of the system. From our experience, it is beneficial to err on the side of caution, and decrease the variance slowly, rather than quickly. Note that this variance is increased only for the unknown fault parameter, now included as an additional state variable in the DBN fault model.

We have developed a heuristic approach to determine the σ_p^{max} and σ_p^{min} values. For p^{-a} and p^{-i} faults, i.e., faults caused by abrupt or incipient decrease in parameter values, we leverage the fact that the parameter value must always be positive. Since the faulty parameter value lies anywhere between its nominal value and slightly above zero, we assign $\sigma_p^{max} \geq \mathcal{P}/3$, where $\mathcal{P}$ is the nominal value of parameter p for a p^{-a} fault. The denominator 3 is due to the fact that 99.7% of the values in a normal distribution lies within three standard deviations of the mean. For a p^{-i} fault, we assign $\sigma_p^{min} \geq \mathcal{P}\Delta t/3$, where Δt is the time difference between two slices of the DBN. For p^{+a} and p^{+i} faults, we assume the max possible values for Δ_p^a and Δ_p^i, denoted by $\Delta_{p_{max}}^a$ and $\Delta_{p_{max}}^i$, respectively. So, for a p^{+a} fault, $\sigma_p^{max} \geq \Delta_{p_{max}}^a/3$, and for a p^{+i} fault, $\sigma_p^{max} \geq \Delta_{p_{max}}^i \Delta t/3$. Once σ_p^{max} values have been determined, we usually set $\sigma_p^{min} = \sigma_p^{max}/10$. Note that in the incipient fault model, it is the noise that accounts for generating particles in the vicinity of the true parameter value at each

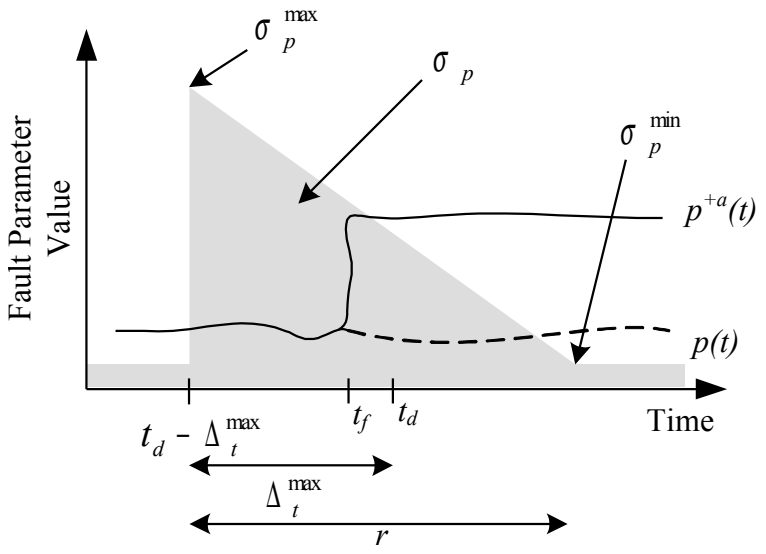
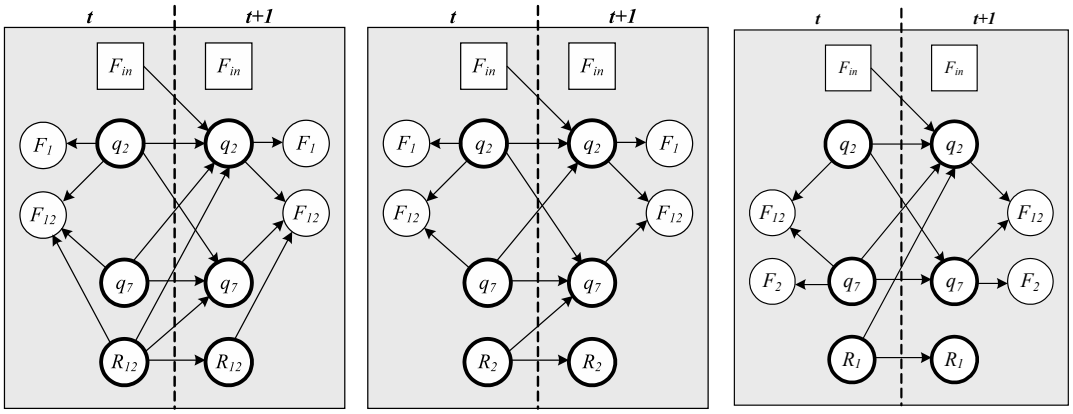


Figure 34: Profile of standard for our particle filtering-based fault identification.

time step. Hence, we must be careful to ensure that σ_p^{min} is not smaller than the actual $\Delta_p^i \times \Delta t$ for that parameter, as otherwise, the gradually changing fault parameter will not be tracked correctly.

In the remainder of this section, we present three different Quant-FHRI schemes. These approaches differ from one another in three main respects: (*i*) the structure of the DBN fault model, with a focus on how the faulty parameter is included in this model, (*ii*) the particle filtering schemes to estimate the faulty parameter using the different DBN fault models, and (*iii*) the post processing steps involved in isolating and identifying the true fault hypothesis if multiple fault hypotheses are modeled by the same DBN fault model.

In the first approach, we adopt the standard practice of including the faulty parameter as a state variable. This results in a single DBN model for both incipient and abrupt faults in a parameter. As a result, fewer number of particle filter estimators need to be instantiated to track the faulty measurements to estimate the fault parameter. However, the common DBN model for abrupt and incipient fault hypotheses necessitates the requirement of a post processing step to match the estimated values with the corresponding fault profile, and extract the relevant parameters (e.g., the percentage change in parameter value for an abrupt fault, and the slope of the change for an incipient fault). Next, we present a second approach where each fault hypothesis for a parameter, p (i.e., abrupt, $p^{\pm a}$, and incipient, $p^{\pm i}$, fault hypotheses) uses a separate DBN fault model. This approach requires a larger number of particle filter estimators than the first approach, but since there is no ambiguity in the fault models, no separate post processing step is required to isolate and identify the true fault. Finally, we present preliminary results for a third Quant-FHRI approach,



(a) DBN model for $R_{12}^{\pm a}$ and $R_{12}^{\pm i}$. (b) DBN model for $R_2^{\pm a}$ or $R_2^{\pm i}$. (c) DBN model for $R_1^{\pm a}$ and $R_1^{\pm i}$.

Figure 35: Single DBN model for both abrupt and incipient faults in the same parameter in Quant-FHRI Approach 1.

where the state space is not augmented with additional state variables. Instead, we analytically calculate the maximum likelihood estimate [110] for the unknown parameter, and use this estimate to generate estimates for the other state variables in the system.

Approach 1: Including Faulty Parameter as State Variable

Generating DBN Fault Models

We generate the DBN fault model by augmenting the nominal DBN model with an extra state variable, that denotes the faulty parameter, p. Augmenting the state variable with the unknown parameter is one of the standard technique used in FDI methods for diagnosis [58, 111]. Note that since we include the possibly time varying faulty parameter as a state variable, a single DBN fault model captures both incipient and abrupt faults in a parameter.

Example. The DBN model of a two-tank system with an incipient or abrupt fault in R_{12} is shown in Fig. 35(a). Every occurrence of constant R_{12} in the state space equation is replaced with the random variable $R_{12}(t)$, and links are drawn accordingly. The state transition function for this new state variable is $P(R_{12}(t+1)|R_{12}(t)) \sim N(\mu_{R_{12}}, \sigma^2_{R_{12}})$, i.e., we rely on the process noise variance, $\sigma^2_{R_{12}}$, to generate particles in the vicinity of the actual value of R_{12}. Similarly, the DBN models for faults in R_2 and R_1 are shown in Fig. 35(b) and Fig. 35(c), respectively. Note that the DBNs shown in Figs. 35(a) and 35(b) represent a fault scenario where only flows F_1 and F_{12} are available for fault diagnosis. Similarly, the DBN in Fig. 35(c) represents a fault scenario where only flows F_{12} and F_2 are available, and not flow F_1.

Tracking Faulty Measurements and Post Processing Step

Once the DBN models are generated, we use the standard particle filtering scheme described in Chapter II, but with a gradually decreasing variance (for the unknown fault state variable) to track the observed measurements. A Z-test is used to determine if the measurement estimates by a particular particle filter significantly deviates from the observed measurements. As discussed earlier, a fault hypothesis is dropped from consideration when the sum of mean square estimation error significantly deviates from zero, and does not converge to the observed measurements within s_d time steps.

In this approach, the incipient and abrupt fault hypotheses have the same fault model, we employ a window-based statistical hypothesis testing scheme (similar to that used in symbol generation in Chapter III) to determine if the faulty hypothesis is incipient or abrupt. This hypothesis testing scheme takes as input the difference between the particle filter estimates of the parameter, and its nominal value. If the fault is abrupt, the symbol generator should generate a -0 or $+0$ deviation, with the $-$ or $+$ magnitude symbol implying that the faulty parameter increases/decreases in magnitude, and the 0 slope symbol implying that the parameter estimate converges to a constant value. If the fault is incipient, we expect the symbol generator to yield $0-$ or $0+$ symbols, implying a gradual increase or decrease in the parameter, and not a constant step change. Depending on whether the fault profile is determined to be abrupt or incipient, the same windowing-based scheme can be employed to determine the value of the bias term, Δ_p^a, in case of an abrupt fault $p^{\pm a}$, or the slope of the drift term, Δ_p^i, in case of an incipient fault.

Illustrative Examples

In the following, we present some runs of our Bayesian diagnosis scheme for a few fault scenarios in the nonlinear two tank system shown in Fig. 12(a). In these examples, we ran our diagnosis scheme on a MATLAB simulation, with white Gaussian noise added to the measurements. For the following examples, we seek to diagnose abrupt and incipient faults in pipes R_1, R_2 and R_{12} only, using a strict subset of measurements, flows F_1, F_{12}, and F_2. Table 14 shows the fault signature matrix for these faults and measurements. In these examples, we assume $k = 2$ and $s = 300$ s.

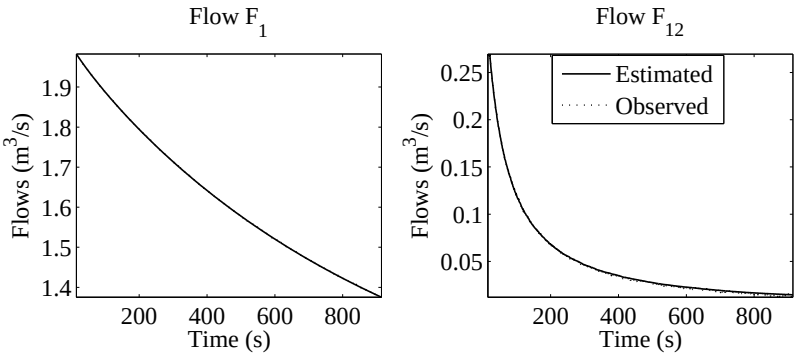
Example. Consider a run of our diagnosis scheme for an incipient fault in pipe, R_{12}, denoted by R_{12}^{+i}, with $\Delta_{R_{12}}^i = 5$, is introduced at time step, $t = 20$ s. The fault detector detects a decrease in measurement F_{12} at $t = 24$ s. Based on this observed decrease in f_5, the fault hypotheses, R_{12}^{+i}, R_2^{+a}, and R_2^{+i} are generated. At $t = 30$ s, F_{12} deviation is mapped onto a $0-$ change. Qual-FI

Table 14: Selected fault signatures for the two-tank system

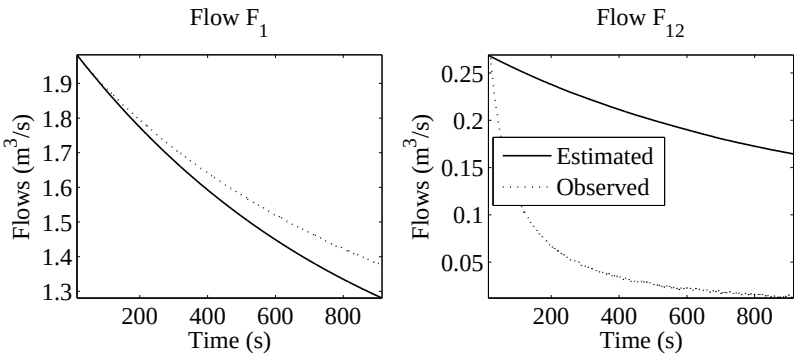
Fault	F_1	F_{12}	F_2
R_1^{+a}	$-+$	$0+$	$0-$
R_1^{+i}	$0-$	$0+$	$0-$
R_{12}^{+a}	$0+$	$-+$	$0+$
R_{12}^{+i}	$0+$	$0-$	$0+$
R_2^{+a}	$0+$	$0-$	$-+$
R_2^{+i}	$0+$	$0-$	$0-$

cannot refine the fault hypotheses further based on this observed deviation. At $t = 51$ s, an increase in flow F_1 is observed, and by $t = 70$ s, this increase is confirmed to be a gradual increase. Again, this observation does not result in any refinement of the fault hypotheses. Since all measurements are observed to have deviated from nominal, Quant-FHRI is invoked. Two separate particle filter estimators, one for R_{12}^{+i}, and the other R_2^{+i} and R_2^{+a} faults are initiated. The DBNs for these fault models are shown in Fig. 35, and explained above. For the two tank system examples, $\Delta_t^{max} = 10$ s. Hence, we start tracking the observed measurements stored from $t = 15$ s onwards with the two different particle filter estimators. The measurement estimates obtained using the two different fault models are shown in Fig. 36. Z-tests indicate a deviation in the measurement estimates obtained using the R_2^{+i}/R_2^{+a} model. Since the fault in R_{12} is incipient, the particle filter estimates made using the R_{12}^{+i} fault model do not show an initial divergence before converging to the observed measurement values. Therefore, the true fault is isolated to R_{12}^{+i}. A window-based symbol generation scheme, when applied to the estimate of R_{12} shows a $0+$ symbol, thus validating that the resistance, R_{12} indeed has an incipient fault. While the actual value of $\Delta_{R_{12}}^i$ is 5, our particle filtering scheme estimates $\Delta_{R_{12}}^i$ to be 4.835. The plot for the estimated R_{12} parameter is shown in Fig. 37. We have repeated this experiment for multiple runs, and the average results are presented in Table 16.

Example. Consider another run of our diagnosis scheme for an abrupt fault in pipe, R_1, denoted by R_1^{+a}, with $\Delta_{R_1}^a = 5$, is introduced at time step, $t = 20$ s. For this experiment, we assume only measurements F_{12} and F_2 are used for diagnosis. The fault detector first detects an increase in measurement F_{12} at $t = 26$ s. This results in the generation of fault hypotheses R_1^{+a} and R_1^{+a}. The DBN model for these fault hypotheses is shown in Fig. 35(c). At $t = 42$ s, F_{12} is determined to show an $0+$ deviation, and F_2 is determined to show an $0-$ deviation at $t = 129$ s. The fault hypotheses are consistent with both observed deviations, and hence cannot be refined further using Qual-FI. Quant-FHRI is initiated at $t = 15$ s, using the single particle filter invoked on the DBN fault model shown in Fig, 35(c). The measurement estimates obtained using this fault model are shown in Fig. 38. Since the true fault is an abrupt fault in R_1, the particle filter estimates diverge



(a) Measurement estimates using R_{12}^{+i} fault model.



(b) Measurement estimates using R_2^{+i}/R_2^{+a} fault model.

Figure 36: Results of diagnosing fault R_{12}^{+i} using Quant-FHRI Approach 1.

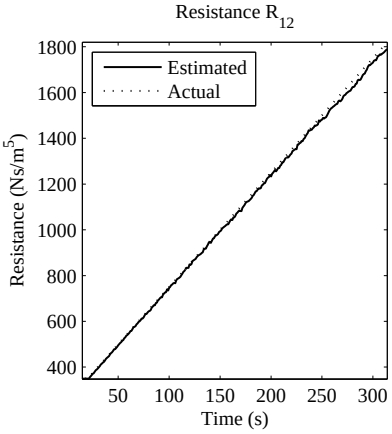


Figure 37: Estimated slope of the true fault R_{12}^{+i} using Quant-FHRI Approach 1.

from the observed measurement values till the unknown R_1 parameter estimates converge to the true value. At this point, the measurement estimates converge to the observed measurements. Whether the true fault is R_1^{+i} or R_1^{+a} is determined using the hypothesis testing scheme, which shows the estimated fault parameter to have a 0 slope, implying the estimate of R_1 has converged to the actual

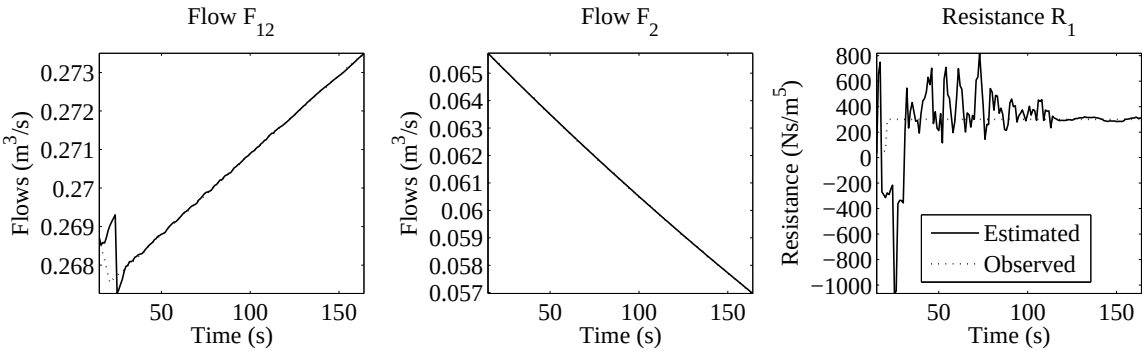


Figure 38: Results of diagnosing fault R_1^{+a} using Quant-FHRI Approach 1.

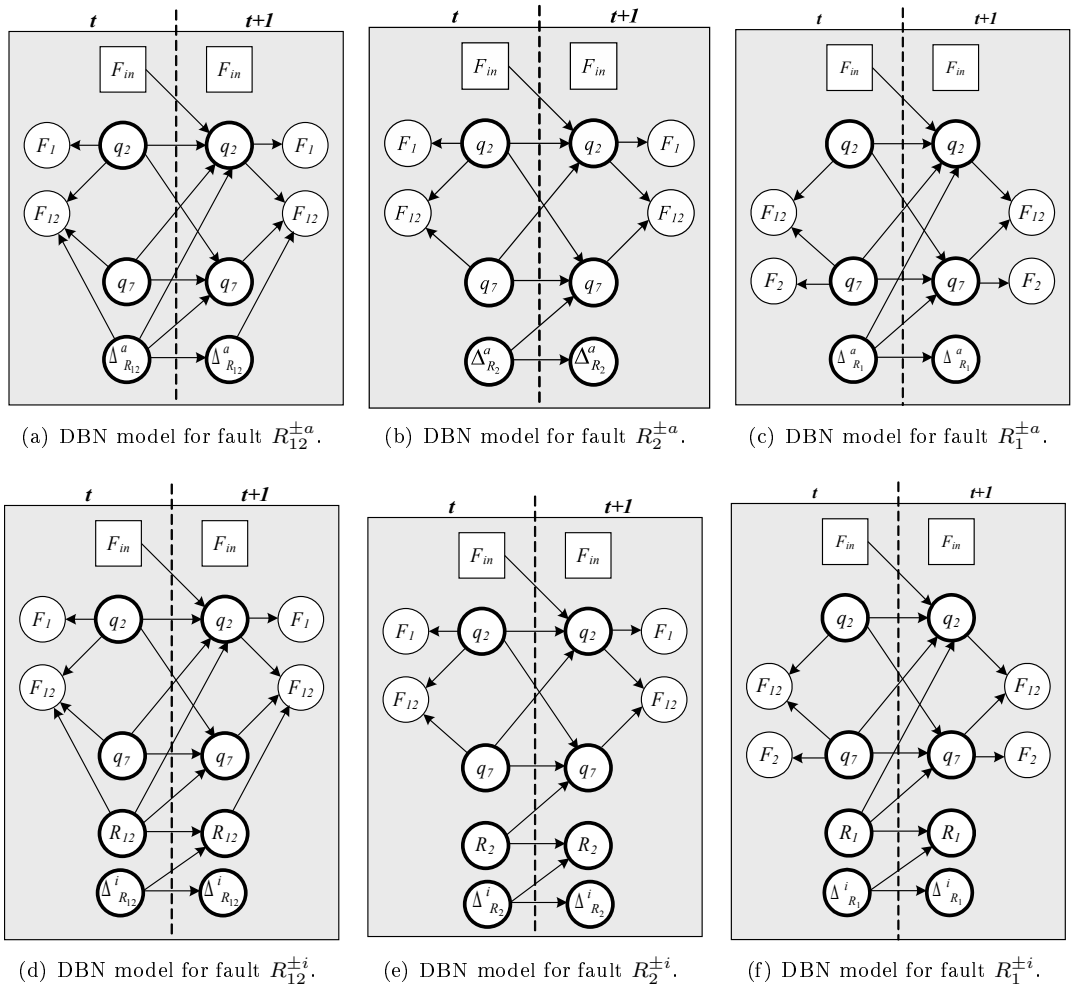
value. The plot for the estimated R_1 parameter is also shown in Fig. 38. While the actual value of $\Delta^a_{R_1}$ is 5, our particle filtering scheme estimates $\Delta^a_{R_1}$ to be 5.192.

Approach 2: Including Fault Magnitude or Slope as State Variable

Generating DBN Fault Models

In this approach, a separate fault model is generated for each abrupt and incipient fault hypothesis. The DBN fault model for abrupt fault $p^{\pm a}$ is derived by augmenting the nominal system model with the abrupt fault magnitude, Δ^a_p, as an extra state variable. The DBN fault model for the incipient fault $p^{\pm i}$ is derived from the nominal DBN model by augmenting it with two additional state variables, the slope of the incipient drift, Δ^i_p, and the faulty parameter, $p(t)$, itself. This additional state variable sums the value of the incipiently evolving parameter over time.

Example. The model of a two-tank system with an abrupt $R_{12}^{\pm a}$ fault includes the additional state variable $\Delta^a_{R_{12}}$. We assume that the magnitude of this bias is constant, i.e., $\Delta^a_{R_{12}}(t+\Delta t) = \Delta^a_{R_{12}}(t)$, where $t \geq t_f$. We generate the faulty system model by replacing all occurrences of R_{12} in the nominal model with $R_{12} \pm \Delta^a_{R_{12}}(t) \times R_{12}$. Fig. 39(a) shows the DBN model for this fault. The model of a two-tank system with an incipient $R_{12}^{\pm i}$ fault includes two additional stochastic variables, $\Delta^i_{R_{12}}$, and R_{12}, when compared to the nominal system. We assume that the slope is constant, i.e., slope $\Delta^i_{R_{12}}(t+\Delta t) = \Delta^i_{R_{12}}(t)$. The fault parameter $R_{12}(t)$ is included as an additional stochastic variable that evolves according to the equations $R_{12}(t+\Delta t) = R_{12}(t) \pm \Delta^i_{R_{12}}(t) \times \Delta t$, and replaces all occurrences of R_{12} in the nominal model. The DBN model for fault $R_{12}^{\pm i}$ is shown in Fig. 39(d). The DBN models for fault hypotheses $R_2^{\pm a}$, $R_2^{\pm a}$, $R_1^{\pm a}$, and $R_1^{\pm i}$ are also shown in Fig. 39.



(a) DBN model for fault $R_{12}^{\pm a}$. (b) DBN model for fault $R_2^{\pm a}$. (c) DBN model for fault $R_1^{\pm a}$.

(d) DBN model for fault $R_{12}^{\pm i}$. (e) DBN model for fault $R_2^{\pm i}$. (f) DBN model for fault $R_1^{\pm i}$.

Figure 39: Separate DBN models for abrupt and incipient faults in the same parameter in Quant-FHRI Approach 2.

Tracking Faulty Measurements and Post Processing Step

Once the DBN fault models are generated, the particle filter estimator is invoked to track the faulty system behavior with the different fault models. In this scheme again, we assume that the variance for the unknown fault slope or magnitude variables gradually decrease over time to a minimum value. Also, in this Quant-FHRI approach, we use an auxiliary particle filtering (APF) scheme for state estimation [12, 112] to obtain an improved performance as compared to the standard particle filtering scheme in the presence of faults. In APF, first the particles are propagated to the next time step (in an attempt to "look-ahead" in the future), and resampled to retain the set of most likely state estimates. The likelihood weights are obtained based on this resampled set of particles. As a result, the posterior density function is better approximated in APFs.

The pseudocode for the generic APF scheme is presented in [12]. Unlike the standard particle filter, in each step of the APF, first a sample of the particle index k is drawn proportional to the likelihood of some reference point, $\mu_t^{(i)}$, that is a characterization of $\mathbf{x}_t$, given $\mathbf{x}_{t-1}^{(i)}$, i.e., the state transition model, such as the mean. Hence,

$$k^{(i)} \sim P(i = k|z_t) \propto w_t^{(i)p(z_t|\mu_t^{(i)})}.$$

These indices are the "auxiliary variables" that are used as in an intermediate step. Once the indices are obtained, these are resampled, to retain the indices that result in the most likely state estimates. Based on the sampled indices, the conditional samples of states are drawn, as follows:

$$x_t^{(i)} \sim P(x_t|x_{t-1}^{k^{(i)}}).$$

Then, as a last step of each iteration, the weights of these particles are updated as follows:

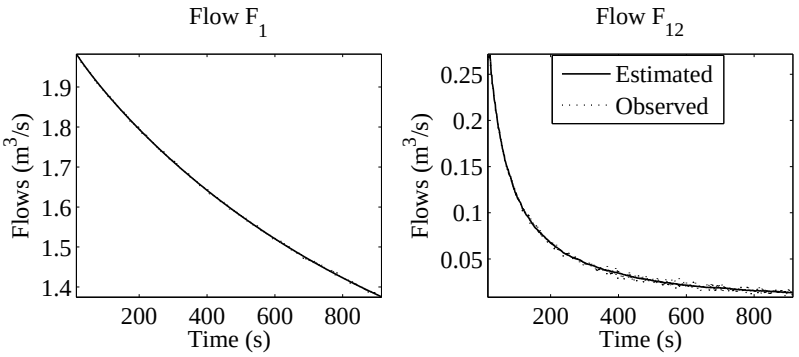
$$w_t^{(i)} \propto \frac{P(z_t|x_t^{(i)})}{P(z_t|\mu_t^{k^{(i)}})},$$

to account for the mismatch between the likelihood at the actual sample and the predicted point $\mu_t^{k^{(i)}}$.

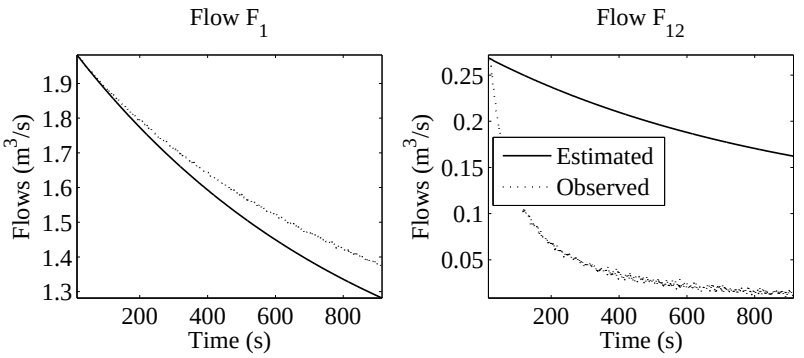
A Z-test is employed to detect significant deviations from nominal in estimated measurements, and drop fault hypotheses corresponding to the fault model used by a particle filter to generate these deviant measurement estimates. Since the Δ_p^a or Δ_p^i variables are included in the model as state variables in every fault models, the magnitude or slope of the true fault is considered to be that estimated directly by the particle filter for the true fault model, and no post processing step is required.

Illustrative Examples

Example. Fig. 40 presents the results of Quant-FHRI using the APF-based approach for the same R_{12}^{+i} fault discussed earlier. As expected, the estimates from the wrong fault model deviated from the observed measurements quickly, isolating R_{12}^{+i} as the true fault. Unlike in the previous example, the fault slope $\Delta_{R_{12}}^i$ is directly estimated. The mean value of $\Delta_{R_{12}}^i$ is obtained to be 5.182, while the real value was 5. The plots comparing the estimates of R_{12} and $\Delta_{R_{12}}^i$ to the actual values is presented in Fig. 41.



(a) Measurement estimates using APF on R_{12}^{+i} fault model.



(b) Measurement estimates using APF on R_2^{+i} fault model.

Figure 40: Results of diagnosing fault R_{12}^{+i} using Quant-FHRI Approach 2.

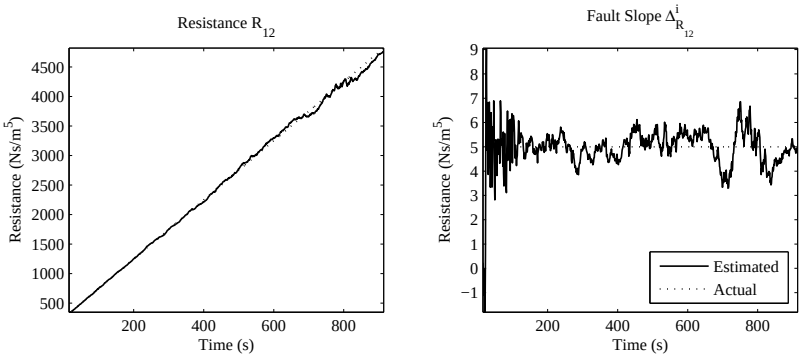
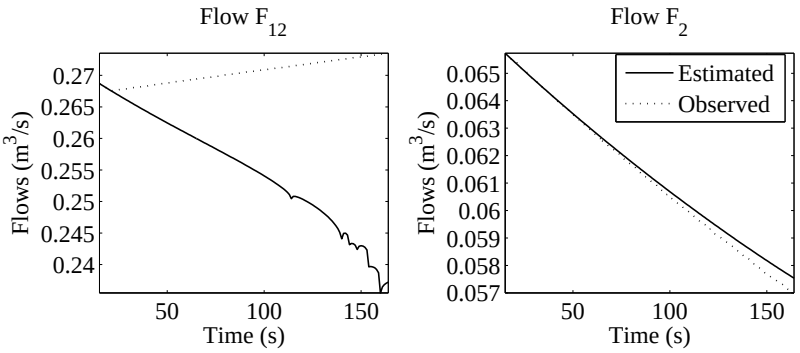
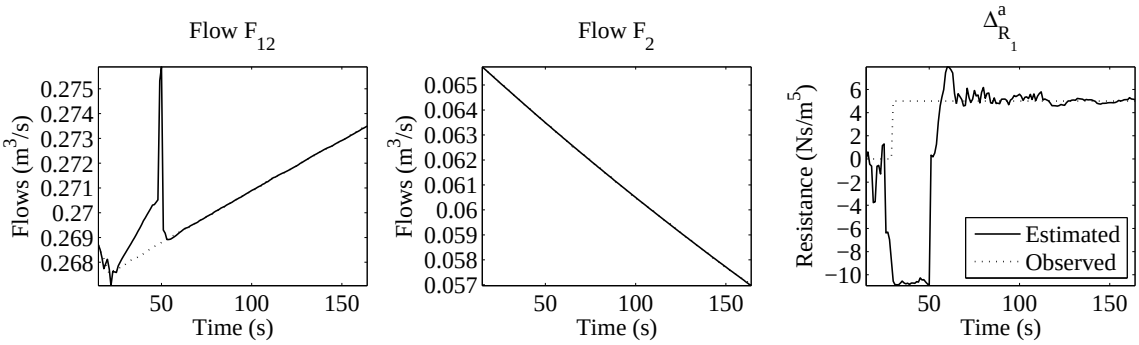


Figure 41: Estimated values of the true fault R_{12}^{+i} and fault slope $\Delta_{R_{12}}^{i}$ using Quant-FHRI Approach 2.

Example. For the R_1^{+a} fault scenario, presented earlier, the APF-based Quant-FHRI scheme requires the invocation of two particle filters instead of just one in the earlier example. One of these APFs operate on the R_1^{+a} model, while the other operates on the R_1^{+i} DBN model, shown in Figs. 39(c) and 39(f), respectively. The estimates of measurements obtained using these two different fault models are shown in Fig. 42. While the estimates using the incorrect fault model, R_1^{+i} deviates from the observed measurements, estimates using the true fault model, R_1^{+a} eventually converge



(a) Estimates by the APF using the R_1^{+i} fault model.



(b) Estimates by the APF using the R_1^{+a} fault model.

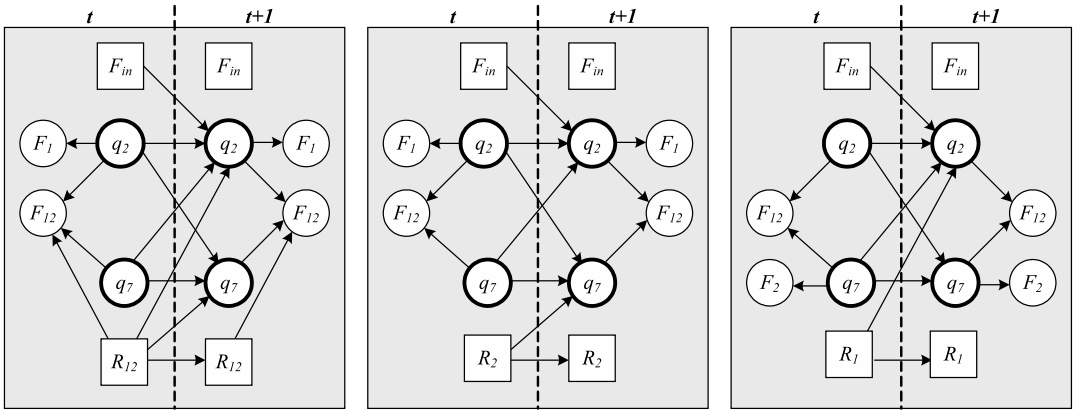
Figure 42: Results of diagnosing fault R_1^{+a} using Quant-FHRI Approach 2.

to the observed measurements, and the fault magnitude parameter, $\Delta_{R_1}^a$ converges to 5.063, which is close to the true value of 5. Hence, we can see that this approach yields a comparable fault identification result as the earlier approach, but is less computationally efficient since this approach required an extra particle filter compared to the earlier approach.

Approach 3: Computing Maximum Likelihood Estimate of Fault Parameter

Generating DBN Fault Models

An alternate approach to state vector augmentation is a maximum likelihood scheme, where the fault parameter is included as an input to the DBN. We derive the analytical expression for the maximum likelihood estimate (MLE) [110] of the faulty parameter in terms of other system variables. The DBN fault model does not contain any extra state variables compared to its nominal DBN model. Similar to Quant-FHRI approach 1, we have the same DBN fault model representing both incipient and abrupt faults in the same parameter. Fig. 43 shows the DBN fault models for approach 3, with



(a) DBN model for $R_{12}^{\pm a}$ and $R_{12}^{\pm i}$. (b) DBN model for $R_2^{\pm a}$ or $R_2^{\pm i}$. (c) DBN model for $R_1^{\pm a}$ and $R_1^{\pm i}$.

Figure 43: Single DBN model for both abrupt and incipient faults in the same parameter in Quant-FHRI Approach 3.

the faulty parameter included as a DBN input. The procedure for calculating the values of the fault parameters is presented next.

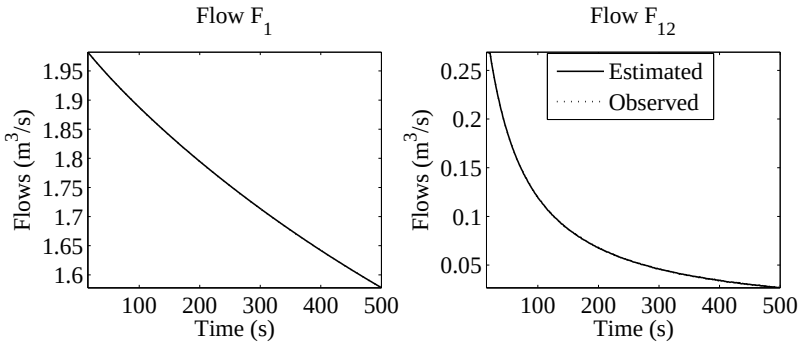
Tracking Faulty Measurements and Post Processing Step

Given this MLE expression, the initial values of the state variables and the unknown parameter, and the observed measurements, we first perform a propagate and update step to get the estimates of the state variables in the present time step. Then, we use these state estimates to compute the MLE value of the unknown parameter. This value of the unknown parameter will be used to estimate the state variable values in the next time step, and so on. Similar to Quant-FHRI approach 1, since the same DBN model can represent multiple fault hypotheses, an additional windowing-based post processing scheme is needed to isolate between the different fault hypotheses the single fault DBN model represents and correctly identify the true fault parameter.

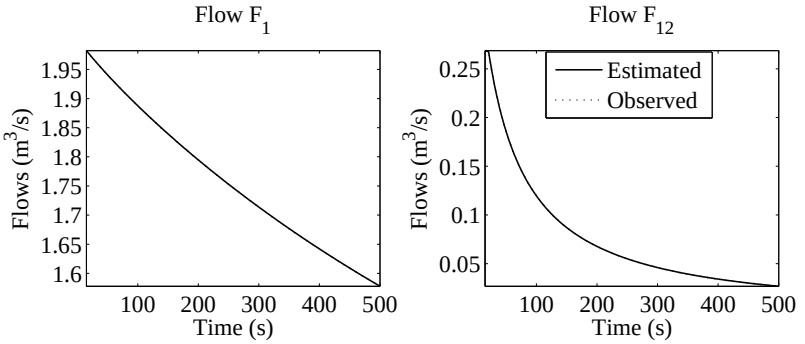
Illustrative Examples

Example. As an example, let us consider identifying the parameter R_{12} in the R_{12}^{+i} incipient fault scenario described in earlier examples in this section. Since we assume that every random variable in the two tank system is sampled from a Gaussian normal distribution, and each measurement variable is independent of other measurements, i.e., the sensor noise covariance matrix is diagonal, the negative log likelihood, $\mathcal{L}$, is computed as follows:

$$\mathcal{L} = \frac{(q_2 - h_{q_2})^2}{\sigma_{q_2}^2} + \frac{(q_7 - h_{q_7})^2}{\sigma_{q_7}^2} + \frac{(f_3 - g_{f_3})^2}{\sigma_{f_3}^2} + \frac{(f_5 - g_{f_5})^2}{\sigma_{f_5}^2} + \frac{(f_8 - g_{f_8})^2}{\sigma_{f_8}^2},$$



(a) Estimates of measurements for R_{12}^{+i} fault scenario using MLE with noise variance of 4×10^{-8} m^6/s^2.



(b) Estimates of measurements for R_{12}^{+i} fault scenario using MLE with noise variance of 5×10^{-12} m^6/s^2.

Figure 44: Results of diagnosing fault R_{12}^{+i} using Quant-FHRI Approach 3.

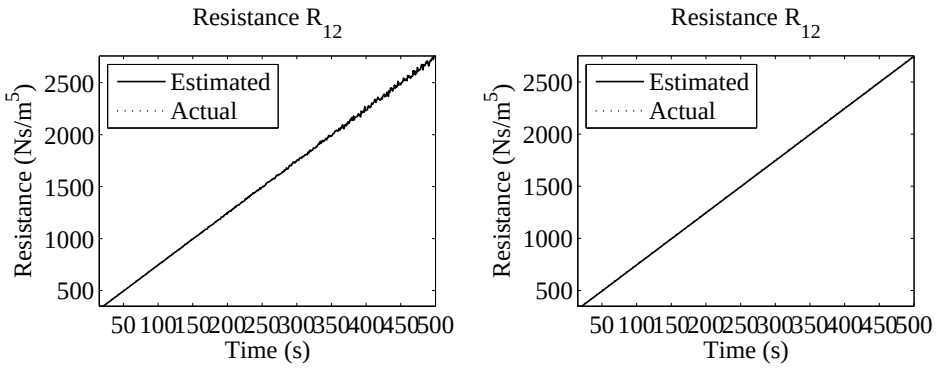
where h_{x_i} denotes the state transition function that defines $x_i(t)|\mathbf{x}(t-1)$, g_{y_j} denotes the observation model $y_j(t)|\mathbf{x}(t)$, and σ_n^2 denotes the variance in the random variable n. To analytically obtain the MLE expression for R_{12}, we equate the partial derivative of $\mathcal{L}$ with respect to R_{12} to 0, i.e., $\dfrac{\partial \mathcal{L}}{\partial R_{12}} = 0$ and solve for R_{12} to obtain

$$R_{12}^{MLE} = \frac{1}{f_5(t)} \left(\frac{q_2(t)}{C_1} - \frac{q_7(t)}{C_2} \right).$$

Fig. 44 shows the estimate of the measurements and R_{12} obtained using this approach for two situations, one where a noise of variance 4×10^{-8} m^6/s^2 was added to each sensor, and the other where the noise variance for each sensor was 5×10^{-12} m^6/s^2. As we can see, this approach is highly sensitive to noise. The estimates for R_{12} parameter for the two scenarios are shown in Fig. 45.

Example. The MLE expression for R_1 can be obtained in a manner similar to the one shown above. By setting $\dfrac{\partial \mathcal{L}}{\partial R_1} = 0$, we get the following expression for R_1:

$$R_1^{MLE} = -\frac{q_2(t-1)}{C_1} \left(q_2(t) - q_2(t-1) + \frac{q_2(t-1)}{C_1 R_{12}} - \frac{q_7(t-1)}{C_2 R_{12}} - F_{in} \right)^{-1}$$



(a) Estimates of R_{12} parameter using MLE with noise variance of 4×10^{-8} m^6/s^2.

(b) Estimates of R_{12} parameter using MLE with noise variance of 5×10^{-12} m^6/s^2.

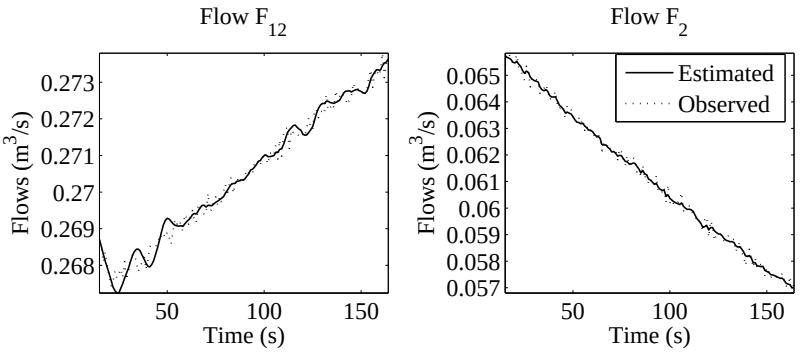
Figure 45: Estimation results of R_{12} using Quant-FHRI Approach 3.

Note that R_1 does not appear in the observation model, and hence, the derivation of its MLE expression is more involved. Fig. 46 shows the estimate of the measurements and R_1 obtained using this approach for two situations, one where the noise is minimal, and the other where the noise is acceptable. As we can see from the plots in Fig. 47, the combined particle filtering and MLE approach for R_1 is not very robust to sensor noise and state estimation errors.

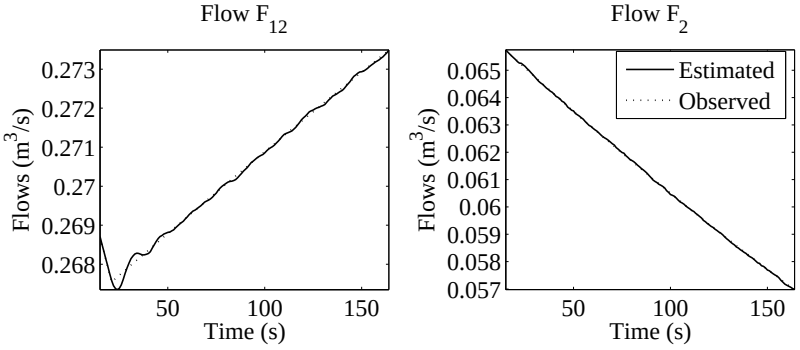
Discussion

We now compare the three Quant-FHRI approaches we presented and evaluate the usefulness of each approach for our combined qualitative-quantitative fault diagnosis approach. In approach 2, the fault magnitude or slope variable are explicitly included in the DBN fault model, resulting in different DBNs to represent abrupt and incipient faults in the same parameter. Hence, unlike approaches 1 and 3, approach 2 does not require any additional post processing step for discriminating between the different fault hypotheses the same DBN model might represent. As the fault models are unique, it is also more likely that the incorrect fault models will result in measurement estimates that quickly diverge from the observed measurements.

However, efficiency is also an important criteria for online diagnosis. Unlike approaches 1 and 3, where a single DBN fault model represents both abrupt and incipient faults in the same parameter, in approach 2, a much larger number of particle filters have to be invoked, one for every unique fault hypotheses. Hence, approaches 1 and 3 are well-suited for our diagnosis approach. Another benefit of approaches 1 and 3 is that the estimates of the fault slope for incipient faults made using approaches 1 and 3 will be less noisy compared to approach 2, since the fault slope included as a state variable in approach 2 is essentially the derivative of the incipiently changing parameter, and

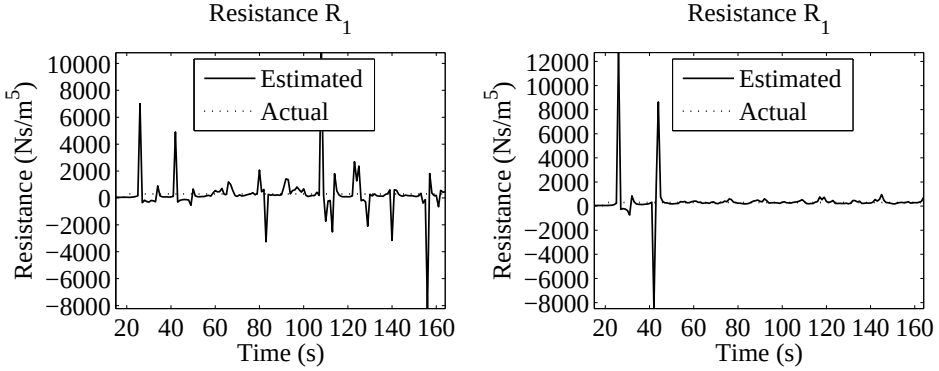


(a) Estimates of measurements for R_1^{+a} fault scenario using MLE with noise variance of 4×10^{-8} m^6/s^2.



(b) Estimates of measurements for R_1^{+a} fault scenario using MLE with noise variance of 5×10^{-12} m^6/s^2.

Figure 46: Results of diagnosing fault R_1^{+a} using Quant-FHRI Approach 3.



(a) Estimates of R_1 parameter using MLE with with noise variance of 4×10^{-8} m^6/s^2.

(b) Estimates of R_1 parameter using MLE with noise variance of 5×10^{-12} m^6/s^2.

Figure 47: Results of diagnosing fault R_1^{+a} using Quant-FHRI Approach 3.

hence prone to being very noisy. When the faulty parameter is included as a state variable, the estimates are less noisy.

Ideally, approach 3 would best suit our needs, computation wise, since the order of the system is not increased for fault identification. However, as we observed from the examples of the two

tank system, Quant-FHRI approach 3 is highly sensitive to noise and estimation errors, and hence fails to achieve our design goal of diagnosis schemes that are robust to sensor noise and modeling inaccuracies. The extent of this sensitivity depends on the system parameters. Also, the generation of analytical expression for maximum likelihood estimates of the faulty parameter, can be tedious for linear systems. For nonlinear systems, the resultant MLE expression can be difficult to derive.

So, we choose approach 1 for our preferred Quant-FHRI scheme. This implies that additional Z-test-based post processing steps will be required to discriminate between the two profiles. However, approach 1 will still be computationally more efficient than having to invoke a separate particle filtering-based tracking scheme for every fault hypotheses, as well as robustness to sensor noise.

Structural Observability

It is well-known that the state variables of a system can be correctly and accurately estimated from the system measurements only if the system is *observable*, i.e., the values of all state variables can be estimated given the past and current values of the available measurements [113]. Accurate estimation of state variables is necessary to correctly track dynamic system behavior, and to correctly estimate fault hypothesis parameter values.

The traditional schemes for analyzing observability are well-defined for linear systems, and depend on the numerical values of the system parameters. However, since our systems can be nonlinear, we explore the *structural observability* property of systems instead, which can be determined through an analysis of the system bond graph [14, 15, 114]. Structural observability is defined for the class of systems having similar structure, and not on the numerical values of the system parameters. Structural observability is also defined for nonlinear systems in which the nonlinearities are in the system components, and not in the system structure. In this section, we establish the theoretical background for structural observability analysis of bond graphs. Structural observability is a less strict property than traditional numeric observability, since structural observability is determined based on system structure and not parameter values. Even if the system is structurally observable, certain values of the parameters can render this system unobservable. However, a slight perturbation of these parameter values can help regain the observability properties, as we explain below.

Consider the basic state-space formulation of a n^{th}-order LTI system:

$$\dot{\mathbf{X}} = \mathcal{A}\mathbf{X} + \mathcal{B}\mathbf{U} \tag{18}$$

$$\mathbf{Y} = \mathcal{C}\mathbf{X} + \mathcal{D}\mathbf{U}, \tag{19}$$

where $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{Y}$ represent the state, input and measurement variables of the system, respectively, and $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ are matrices with appropriate dimensions.

Definition 10. [113] (Observability). A system is termed *observable* if its initial state variables, $\mathbf{X}_{t_0}$, at time t_0, can be derived from the knowledge of inputs, $\mathbf{U}_{t_0:t_f}$, and outputs, $\mathbf{Y}_{t_0:t_f}$, in the time interval $[t_0, t_f]$, where t_f is the current time.

It is well known that a system is observable if its observablity matrix,

$$\mathcal{O} = \begin{bmatrix} \mathcal{C} \\ \mathcal{C}\mathcal{A} \\ \cdots \\ \mathcal{C}\mathcal{A}^{n-1} \end{bmatrix},$$

is of full rank, i.e., $rank(\mathcal{O}) = n$.

From above, system observability is a function of the numeric values of the system parameters. An alternative approach has been proposed that is based on the analysis of the system structure [14,15, 114,115]. This notion of *structural observability* holds for a class of structurally equivalent systems. If a system is structurally observable, but its $\mathcal{O}$ matrix is not of full rank, i.e., $rank(\mathcal{O}) < n$, the rank can be restored to n by perturbing the values of elements of its $\mathcal{A}$ and $\mathcal{C}$ matrices [14].

One of the earliest work on structural observability is presented in [115], where the author analyzes the *structured graph*, $G(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, of a system to determine if the system satisfies the property of structural observability. A structured graph can be completely described by its adjacency matrix, the *structured matrix* M. Because the elements of the adjacency matrix M are either 0 or 1, this matrix is often called a *boolean matrix*, we can denote this adjacency matrix as

$$M = \begin{bmatrix} \mathcal{A}_B & \mathcal{B}_B \\ \mathcal{C}_B & \mathcal{D}_B \end{bmatrix},$$

with the index "B" denoting the boolean counterparts of a matrix.

A graph theoretic approach to analyzing the structural properties of a system was proposed in [115], where the author notes that a system is structurally observable iff:

1. the states (or nodes) are all output reachable
2. term-rank $\begin{bmatrix} \mathcal{A}_B \\ \mathcal{C}_B \end{bmatrix} = n$, where n is the number of state variables in the system.

The *term-rank* of the matrix M is given by the number of elements in a maximal *permutation matrix* contained in M. A permutation matrix is a square $(0,1)$-matrix that has exactly one entry 1 in each row and each column and 0's elsewhere. Each such matrix represents a specific permutation of m elements, and when used to multiply another matrix, can produce that permutation in the rows or columns of the other matrix.

Example. Consider the state matrix of an electrical network may have

$$\mathcal{A} = \begin{bmatrix} -\frac{R}{L_1} & \frac{R}{L_2} \\ \frac{R}{L_1} & -\frac{R}{L_2} \end{bmatrix}.$$

While $\mathcal{A}$ can never be of rank 2, its associated boolean adjacency matrix

$$\mathcal{A}_B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

has a term-rank of 2.

Thus, information is lost when the boolean adjacency matrix is used for structural analysis. Bond graphs provide an elegant approach for determining structural observability of a system [14] without losing the important information present in functional relationship between the elements of the matrix. The notion of *structural rank* (*struct-rank*) is central to establish the necessary and sufficient conditions for this bond graph-based structural analysis procedure

Definition 11. [14] (Structural Rank). *Structural rank* of a matrix is defined as the maximal rank of this matrix as a function of its free parameters, taking into account the relations between parameters.

Example. Consider the same $\mathcal{A}$ matrix we presented in an earlier example to illustrate the notion of term-rank. If we now evaluate the structural rank of this matrix, we obtain

$$\textit{struct-rank}\left(\begin{bmatrix} -\frac{R}{L_1} & \frac{R}{L_2} \\ \frac{R}{L_1} & -\frac{R}{L_2} \end{bmatrix}\right) = 1,$$

since the second row of the matrix is linearly dependent on the first row.

In terms of bond graphs, given the bond graph model of a system with matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$, the system is structurally observable iff [14]:

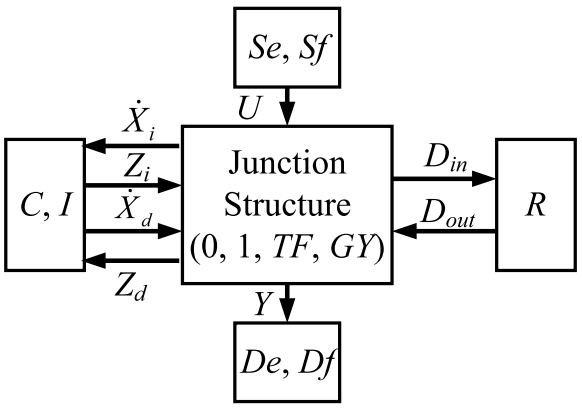


Figure 48: Junction Structure.

1. every dynamical element of the BG in integral causality is causally connected to a measurement sensor, and

2. *struct-rank*$([\mathcal{A}^t\ \mathcal{C}^t]^t) = n$, where n is the number of state variables in the system.

Intuitively, condition 1 is satisfied if for each independent decoupled subsystem, at least one dynamical element in integral causality is causally connected to a measurement.

Condition 2 is satisfied if the causality of *every* I and C element initially in integral causality can be inverted to produce a valid derivative causality assignment for the BG model. In some situations, De and Df elements may have to be changed into their dual form to assign consistent derivative causality to the BG. This procedure for manipulating the BG to directly determine the structural rank of matrix $[\mathcal{A}^t\ \mathcal{C}^t]^t$ has been presented as the following result in [14]:

$$\textit{struct-rank}\left(\begin{bmatrix}\mathcal{A}\\ \mathcal{C}\end{bmatrix}\right) = \textit{rank}\left(\begin{bmatrix}S_{11}\\ S_{21}\\ S_{31}\end{bmatrix}\right) = n - T_d, \tag{20}$$

where S_{11}, S_{21}, and S_{31} are components of the junction structure matrix (introduced below); and T_d is the number of dynamical elements remaining in integral causality after (i) derivative causality assignment is performed, and (ii) the maximal number of output detectors are dualized to eliminate as many storage elements in integral causality as possible. Hence, if every I and C element initially in integral causality can be assigned derivative causality, $T_d = 0$, and condition 2 above is satisfied. Structural analysis of a BG model can help determine the structural rank of $[\mathcal{A}^t\ \mathcal{C}^t]^t$ since the structure of the BG plays a crucial role in determining the state-space equations of a system, as we show below.

The *junction structure* (see Fig. 48) represents the structure of a BG and contains information about the BG elements, and how they are interconnected (this is independent of the numerical values of the parameters). The junction structure can be represented using a *junction structure matrix*, S [15]:

$$\begin{bmatrix} \dot{X}_i \\ D_{in} \\ Y \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & 0 & S_{23} & S_{24} \\ S_{31} & 0 & S_{33} & S_{34} \end{bmatrix} \begin{bmatrix} Z_i \\ \dot{X}_d \\ D_{out} \\ U \end{bmatrix} \tag{21}$$

where the state vector X_i is composed of energy variables $p = \int e\ dt$ (respectively, $q = \int f\ dt$) on I (respectively, C) elements) in integral causality (denoted by subscript i), X_d is the vector of energy elements of I and C elements in derivative causality, Y is the vector of system outputs, and U is the vector of system inputs. D_{in} and D_{out} represent the effort or flow variables imposed upon, and imposed by the R elements, respectively, as shown in Fig. 48. Z_i and Z_d denote the vector of flow (respectively, effort) variables of I (respectively, C) elements in integral and derivative causality, respectively.

Example. Given the bond graph of the two-tank system, shown in Fig. 12(b), we have

$$\dot{X}_i = \begin{bmatrix} \dot{q}_2 \\ \dot{q}_7 \end{bmatrix},\ Z_i = \begin{bmatrix} \dot{e}_2 \\ \dot{e}_7 \end{bmatrix},\ D_{in} = \begin{bmatrix} e_9 \\ e_5 \\ e_{10} \end{bmatrix},\ D_{out} = \begin{bmatrix} f_9 \\ f_5 \\ f_{10} \end{bmatrix},\ Y = \begin{bmatrix} F_1 \\ F_{12} \\ F_2 \end{bmatrix},\ \text{and } U = [f_1]\,.$$

Therefore, junction structure matrix is derived to be:

$$\begin{bmatrix} \dot{q}_2 \\ \dot{q}_7 \\ e_9 \\ e_5 \\ e_{10} \\ F_1 \\ F_{12} \\ F_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e_2 \\ e_7 \\ f_9 \\ f_5 \\ f_{10} \\ f_1 \end{bmatrix} \tag{22}$$

Basic laws associated with each component produce $D_{out} = LD_{in}$, and $Z_i = F_i X_i$, where L is a diagonal matrix composed of R and $\frac{1}{R}$ coefficients, and F_i is composed of $\frac{1}{I}$ and $\frac{1}{C}$ coefficients. We assume that the original system models do not have I an C elements in derivative causality, as these elements can be usually collapsed into one "equivalent" inductor or capacitor in integral causality. Hence, in this work, we assume that X_d and Z_d do not exist. Therefore, the $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ matrices of the state-space equations of a system can be derived from its corresponding junction structure matrix as follows [14]:

$$\begin{aligned}
\mathcal{A} &= [S_{11} + S_{13}L(I - S_{23}L)^{-1}S_{21}]F_i \\
\mathcal{B} &= [S_{14} + S_{13}L(I - S_{23}L)^{-1}S_{24}] \\
\mathcal{C} &= [S_{31} + S_{33}L(I - S_{23}L)^{-1}S_{21}]F_i \\
\mathcal{D} &= [S_{34} + S_{33}L(I - S_{23}L)^{-1}S_{24}].
\end{aligned}$$

To prove Eqn. (20), let us assume $X_i = [X_{i_1}^t X_{i_2}^t]^t$, with $X_{i_1} \in \mathbb{R}^{n-m}$, $X_{i_2} \in \mathbb{R}^m$, and $rank([S_{11}\ S_{12}]) = m$. Given the junction structure shown in Eqn. (21), switching energy storage elements to derivative causality, and retaining consistent causality assignments in the BG model without having to dualize the output detectors yields a new junction structure:

$$\begin{bmatrix} \dot{X}_{i_1} \\ Z_{i_2} \\ D_{in} \\ Y \end{bmatrix} = \begin{bmatrix} 0 & M_1 & 0 & M_2 \\ M_3 & M_4 & M_5 & M_6 \\ 0 & M_7 & M_8 & M_9 \\ M_{10} & M_{11} & M_{12} & M_{13} \end{bmatrix} \begin{bmatrix} Z_{i_1} \\ \dot{X}_{i_2} \\ D_{out} \\ U \end{bmatrix}, \tag{23}$$

where X_{i_1} represents I and C elements remaining in integral causality after derivative causality assignment is performed, and X_{i_2} represents those I and C elements that are assigned derivative causality.

Dualizing of detectors implies decomposing $Y = [Y_1^t\ Y_2^t]^t$, $Y \in \mathcal{R}^p$, and $Y_1 \in \mathcal{R}^{p*}$, where Y_1 represents the sensors that are dualized, and Y_2 represents those sensors that are not dualized.

After dualizing the sensors, a new junction structure is built as follows [14]:

$$\begin{bmatrix} \dot{X}_i \\ Z \\ D^*_{in} \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 & N_1 & 0 & 0 \\ N_2 & N_3 & N_4 & N_5 \\ 0 & N_6 & N_7 & N_8 \\ 0 & N_{10} & 0 & N_{11} \end{bmatrix} \begin{bmatrix} Z_i \\ \dot{X} \\ D^*_{out} \\ U^* \end{bmatrix}, \tag{24}$$

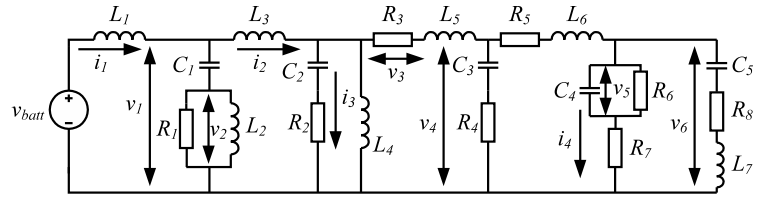
where, $U^* = [U^t\ Y_1^t]^t$. In Eqn. (24), $\dot{X}_i$ depends only on the $\dot{X}$ variables now in derivative causality. Equation (24) can also be obtained using the invertible matrix contained in $[S_{11}^t\ S_{21}^t\ S_{31}^t]^t$. Hence, it can be concluded that $rank([S_{11}^t\ S_{21}^t\ S_{31}^t]^t) = n - T_d$. Finally, we can prove $struct\text{-}rank([A^t\ C^t]^t) = rank([S_{11}^t\ S_{21}^t\ S_{31}^t]^t)$ using the same considerations as in Appendix 1 of [14], but with matrix $[A^t\ C^t]^t$.

The proposed method for analyzing structural observability for linear systems is applicable to nonlinear systems as well, as long as the nonlinearities are not linked to the system structure, and do not change the junction structure [14]. The applicability of this bond graph-based structural observability analysis approach to nonlinear systems is evident from the fact that the elements of the junction structure matrix do not include any information regarding the system parameters. Therefore, the junction structure matrix of the two-tank system would remain unchanged from that given above if the resistance R_2 is assumed to be a constant. The nonlinearity, however, becomes evident when generating the state-space equations, since this step requires the use of F_i and L matrices, whose elements are made of the system parameter values.

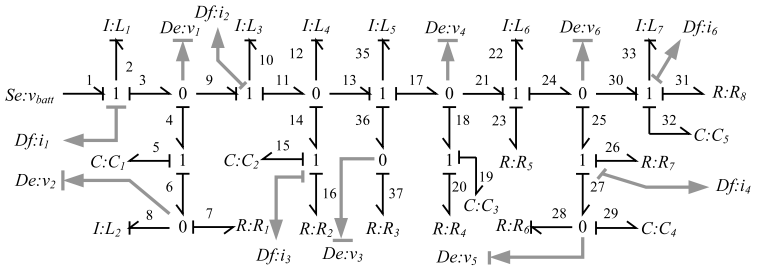
In our combined qualitative-quantitative diagnosis scheme, we ensure that the system to be diagnosed is structurally observable, as otherwise, there is no guarantee that the particle filtering-based observer would generate correct estimates of states.

Case Study: Twelfth-order Electrical Circuit

In this section, we use a twelfth-order electrical circuit shown in Fig. 49(a) as a case study to illustrate the concepts presented in this chapter. Fig. 49(b) shows the bond graph of this circuit. In the electrical domain, the effort variables denote voltage difference across, and flow variables denote current through, BG elements. For example, $f_2 = i_1$ denotes the current through the inductor L_1, and $e_7 = v_2$ denotes the voltage difference across resistor R_1. $e_1 = v_{batt}$ denotes the voltage imposed by the voltage supply. $De : v_2$ is a voltage sensor. Fig. 50 shows the DBN for the example circuit.



(a) Schematic.



(b) Bond graph.

Figure 49: Electrical circuit models.

We present some experimental results obtained by applying our diagnosis approach to the twelfth-order electrical circuit. In such electrical systems, usual faults include degradation in capacitors and inductors, and increase in the resistance of resistors. Table 15 lists the specific faults of interest in this system, the measurements available to diagnose these faults, and the fault signature of each fault for the different measurements. The goal of these experiments is to demonstrate that our combined qualitative-quantitative diagnosis scheme has more discriminatory power than the TRANSCEND qualitative diagnosis scheme.

In our experiments, we assume all random variables, and the prior and conditional probabilities are sampled from Gaussian Normal. The mean and variance of each hidden variable is set based on empirical knowledge of the model. The means and variances of the observed variables, as well as the conditional probabilities, are functions of the estimated system parameters, and the parameters of distributions of the hidden variables. For the experiments below, we set $k = 5$ and $s = 300$ s.

System behavior is generated for a total of 500 time steps using a Matlab Simulink simulation model and data sampled at a rate of 10 Hz. According to standard practice, white Gaussian noise with zero mean and constant variance is added to the measurements. The measurements are saved to a file, and then run through our fault diagnosis scheme (implemented in Matlab) to generate our experimental results.

We present a run of our diagnosis scheme for a specific fault scenario. An abrupt fault in C_2, C_2^{-a}, with $\Delta^a_{C_2} = -0.900$, is introduced at time step, $t = 100$ s. As shown in Fig. 51, a negative

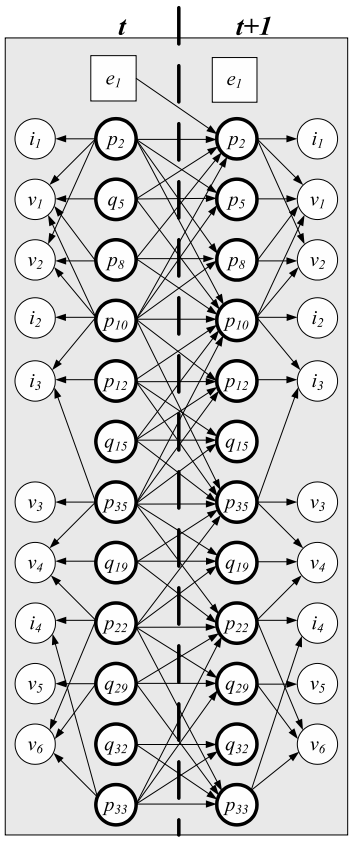


Figure 50: DBN model of the electrical circuit.

deviation is noticed in measurement i_3 at $t = 101.3$ s. Based on this deviation, the following fault hypotheses are generated $\{C_2^{-i}, C_2^{-a}, R_2^{+a}, R_2^{+i}, L_2^{-i}, L_2^{-a}, L_4^{-i}, L_4^{-a}, L_7^{-i}, L_7^{-a}\}$. At $t = 102.5$ s, the deviation in i_3 is confirmed to be a gradual decrease. Hence, fault hypothesis L_4^{-a} is dropped from consideration. At $t = 102.8$ s, measurement i_2 shows a negative deviation, as a result of which, the Qual-FI approach drops fault hypotheses L_4^{-i}, L_7^{-a}, and L_7^{-i} from consideration. At $t = 103.9$ s, measurement v_3 shows a positive deviation. As a result, based on the fault signatures shown in Table 15, the fault hypotheses are refined to $\{C_2^{-i}, C_2^{-a}, R_2^{+a}, R_2^{+i}\}$, after dropping fault hypotheses L_2^{-a} and L_2^{-i}. Since these fault hypotheses cannot be further refined through Qual-FI alone, Quant-FHRI is initiated. We start tracking the observed measurements from time $t = 97.5$ s, and instantiate two particle filters, one using a DBN model for fault C_2^{-i}/C_2^{-a}, and the other using a DBN model for fault R_2^{+i}/R_2^{+a}, where the faulty parameter, i.e., in this case, parameters C_2 and R_2 are introduced as additional state variables in the nominal system DBN shown in Fig. 50. Fig. 52 shows the error in estimating the different measurements using the C_2 DBN fault model. As is expected, at the time the fault is introduced, the fault parameter value is unknown. Hence, it takes some time for the correct fault model estimates to converge to the observed measurements, as we

Table 15: Selected fault signatures for the electrical circuit

Fault	i_1	i_2	i_3	v_1	v_2	v_3	i_4	v_4	v_5	v_6
C_2^{-a}, C_2^{-i}, R_2^{+a}, R_2^{+i}	0−	0−	0−	0+	0+	0+	0+	0+	0+	0+
C_3^{-a}, R_4^{+a}	0−	0−	0+	0+	0+	0−	0+	+−	0+	0+
C_3^{-i}, R_4^{+i}	0−	0−	0+	0+	0+	0−	0+	0+	0+	0+
C_4^{-a}	0−	0−	0+	0+	0+	0−	0−	0+	+−	+−
C_4^{-i}, R_6^{+a}, R_6^{+i}	0−	0−	0+	0+	0+	0−	0−	0+	0+	0+
L_2^{-a}	0+	0−	0−	−+	−+	0−	0−	0−	0−	0−
L_2^{-i}	0+	0−	0−	0−	0−	0−	0−	0−	0−	0−
L_3^{-a}	0+	+−	+−	−∗	−+	0+	0+	0+	0+	0+
L_3^{-i}	0+	0+	0+	0−	0−	0+	0+	0+	0+	0+
L_4^{-a}	0+	0+	−+	0−	0−	0−	0−	0−	0−	0−
L_4^{-i}	0+	0+	0−	0−	0−	0−	0−	0−	0−	0−
L_7^{-a}	0+	0+	0−	0−	0−	0+	−+	0−	0−	−∗
L_7^{-i}	0+	0+	0−	0−	0−	0+	0−	0−	0−	0−
R_7^{+a}	0−	0−	0+	0+	0+	0−	0−	0+	0−	+−
R_7^{+i}	0−	0−	0+	0+	0+	0−	0−	0+	0−	0+

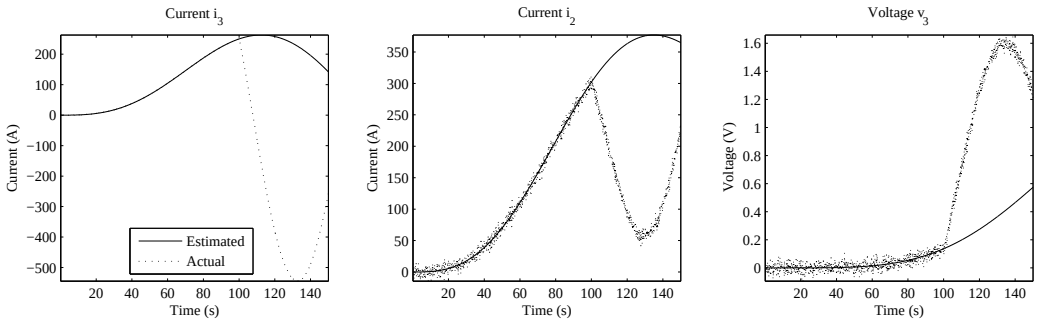


Figure 51: Detection of C_2^{-a} fault through tracking system behavior using nominal DBN model.

can see for the plots of measurements, v_2, i_2, i_3, and v_3 in Fig. 52. Estimates made using the incorrect R_2 DBN fault model, on the other hand, do not converge to the observed faulty behaviors, as the estimation errors in Fig. 53 show. A statistical test is employed on the sum of estimation errors across all the measurements to detect statistically significant sum of mean squared estimation errors. Fig. 54 shows the sum of mean squared estimation errors obtained using the two different fault models. The Z-test detects a statistically significant sum of mean squared estimation error obtained by both the DBN fault models at times $t = 101.2$ s and $t = 101.3$ s, respectively. However, the sum of mean squared estimation errors from the R_2 DBN fault model do not converge even after $s_d = 150$ s, whereas, the sum of mean squared estimation errors from the C_2 DBN fault model converges to the observed measurements from $t = 103.7$ s. Hence the true fault is isolated to be $C_2^{\pm a}/C_2^{\pm i}$ fault at $t = 251.3$ s. In order to isolate whether the fault is an abrupt or incipient fault in C_2, we run a window-based Z-test on the difference between the known nominal parameter value and the estimated state variable. This approach is similar to the one used by TRANSCEND for symbol

generation. At $t = 171.7$ s, the statistical test shows that the estimated parameter evolves in a -0 manner, implying it is an abrupt fault, and that it converges . By taking a mean of the values for 20 time steps after the abrupt fault is isolated, we obtain $\Delta^a_{C_2} = -0.897$. The actual value of $\Delta^a_{C_2}$ is -0.900. The estimate for the faulty parameter is shown in Fig. 55.

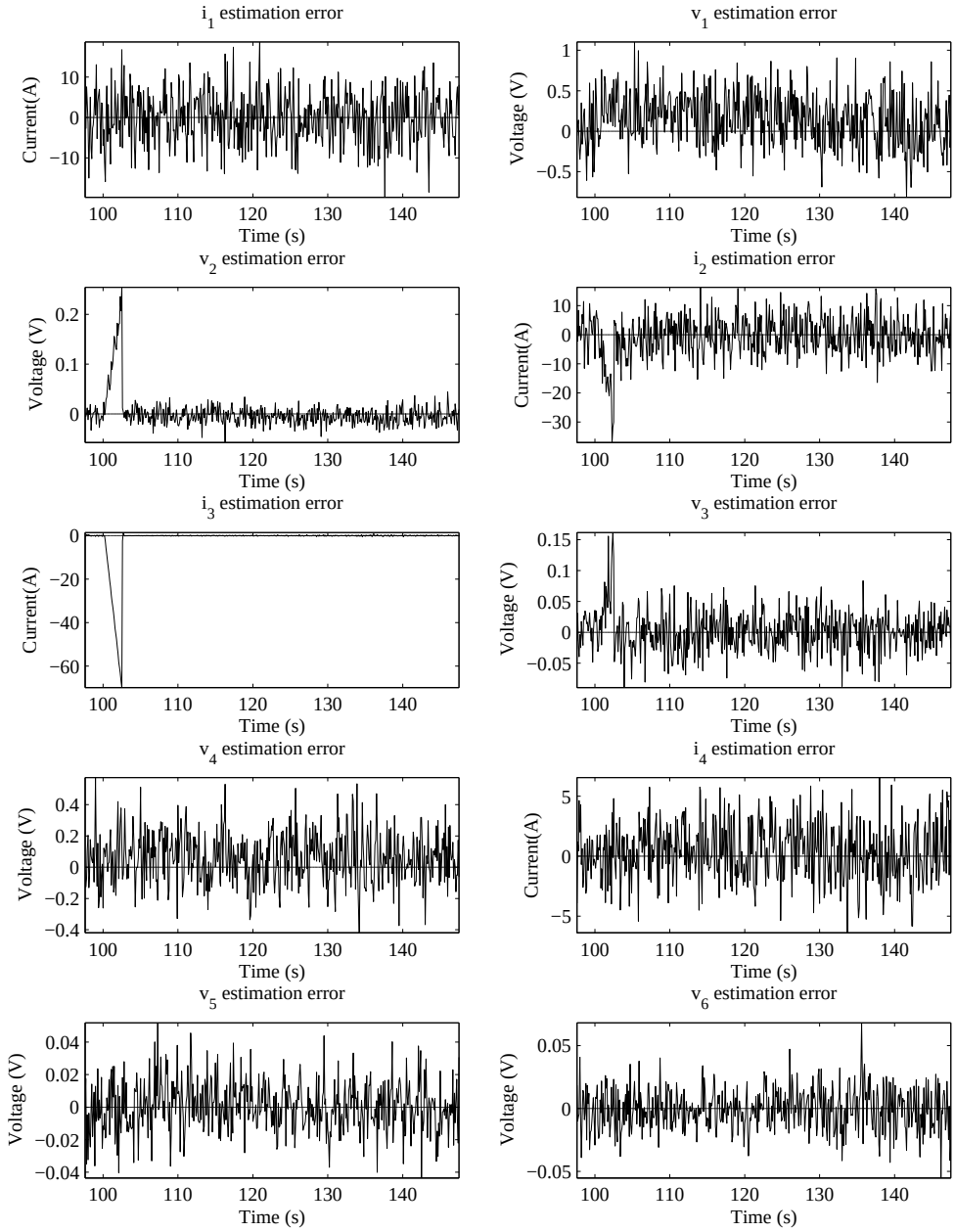


Figure 52: Tracking faulty system behavior using the $C_2^{\pm i}/C_2^{\pm a}$ DBN fault model.

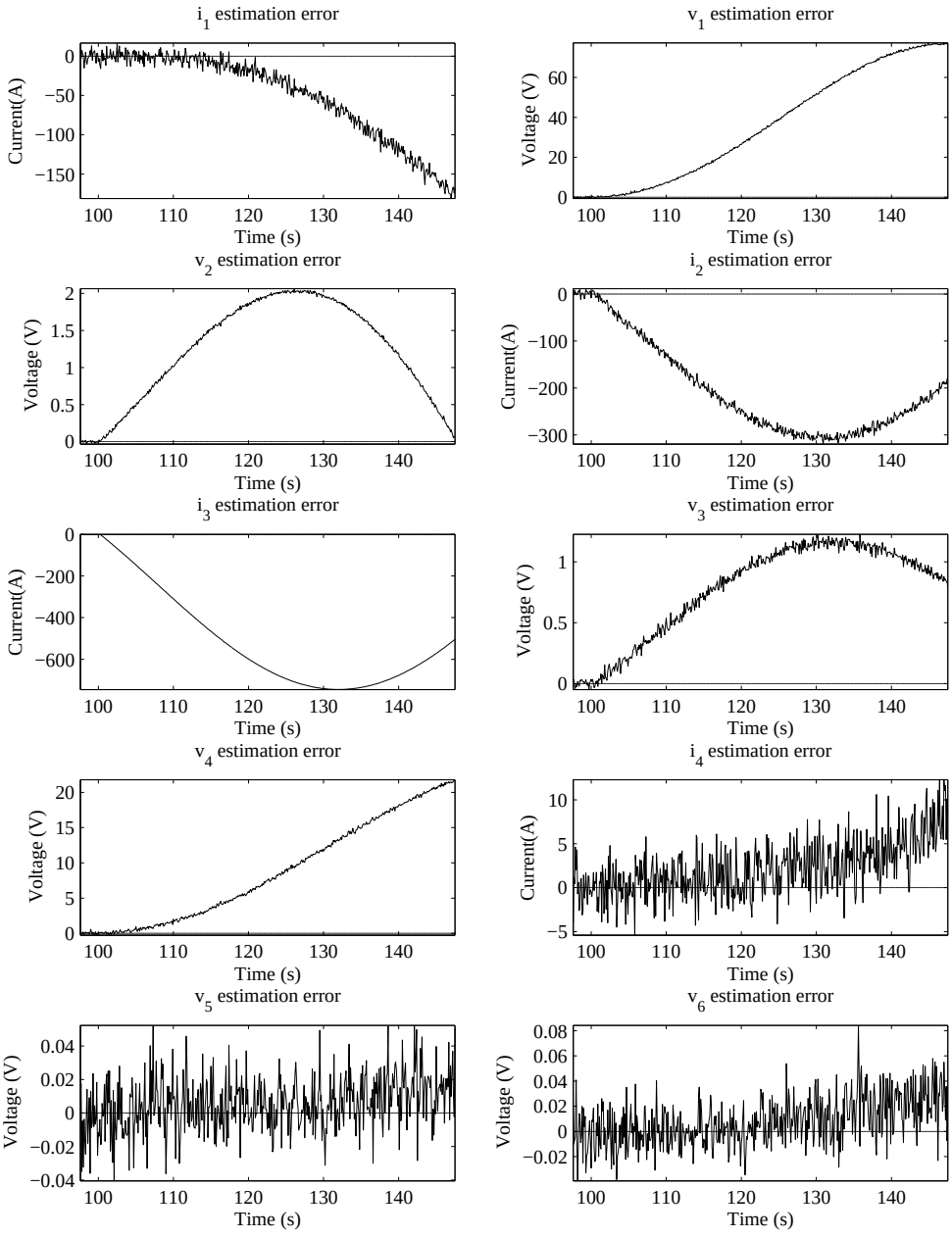
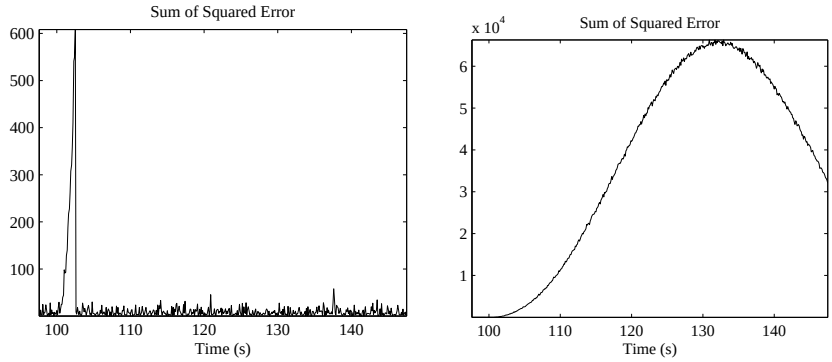


Figure 53: Tracking faulty system behavior using the $R_2^{\pm i}/R_2^{\pm a}$ DBN fault model.



(a) Sum of mean squared estimation errors obtained by C_2 DBN fault models.

(b) Sum of mean squared estimation errors obtained by R_2 DBN fault models.

Figure 54: Sum of mean squared estimation errors obtained by C_2 and R_2 DBN fault models.

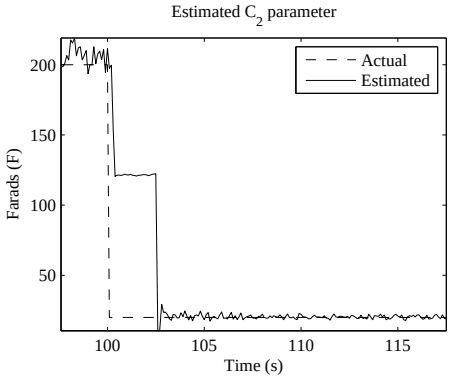


Figure 55: Estimate of C_2 obtained using the C_2 DBN fault model.

In another scenario, we diagnose an incipient fault in L_3, denoted by L_3^{-i}, with $\Delta^i_{L_3} = -0.050$, injected at time $t = 20$ s. At time $t = 69.4$ s, a positive deviation from nominal is observed in measurement i_3 (see Fig. 56). This results in the generation of fault hypotheses $\{C_3^{-a}, C_3^{-i}, R_4^{+a}, R_4^{+i}, C_4^{-i}, C_4^{-a}, R_6^{+a}, R_6^{+i}, L_3^{-a}, L_3^{-i}, R_7^{+a}, R_7^{+i}\}$. At $t = 92.3$ s, a positive deviation is observed in measurement i_2, as a result of which, the fault hypotheses set is refined by the Qual-FI scheme to $\{L_3^{-a}, L_3^{-i}\}$, based on the fault signatures shown in Table 15. At $t = 100.6$ s, the deviation in i_3 is confirmed to be 0+, which results in the fault hypothesis L_3^{-a} being dropped from consideration and the fault hypotheses set to be refined to the single fault hypothesis L_3^{-i}. Since the fault hypotheses set is refined to a single fault by Qual-FI, we invoke the Quant-FHRI to perform fault identification only. We start tracking the observed measurements using a L_3 DBN fault model from $t = 17.5$ s. After $s_d = 150$ s from the time the faulty observations are tracked, a Z-test based scheme is initiated, and this statistical test confirms that the fault is indeed an incipient L_3^{-i} fault at $t = 172.3$ s. The

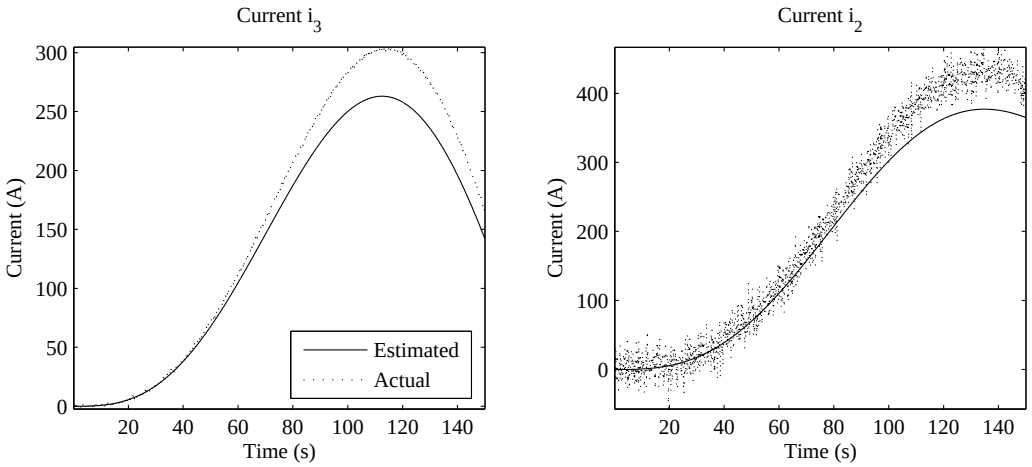


Figure 56: Detection of L_3^{-i} fault through tracking system behavior using nominal DBN model.

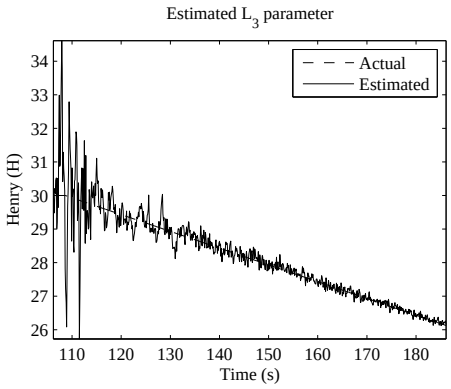


Figure 57: Estimate of L_3 obtained using the L_3 DBN fault model.

estimates of L_3 obtained from our Quant-FHRI approach is presented in Fig. 57. The Quant-FHRI estimates $\Delta^i_{L_3} = -0.047$, while the true value of is $\Delta^i_{L_3} = -0.050$.

Table 16 summarizes the results of the different diagnosis experiments we ran on the electrical circuit example. For each experiment, we conducted 5 runs, and took the average of the time to fault detection, time to single fault isolation (either by Qual-FI alone, or, by both Qual-FI and Quant-FHRI), time for the estimated parameter value to convergence to the true value, and the percentage estimation error in the estimates of the true fault parameter. The estimates of the faulty parameters made using the correct fault models for each respective fault for one of these experimental runs is shown in Fig. 58. When compared to the TRANSCEND diagnosis scheme, this combined diagnosis approach allows more precise diagnosis approach. For example, using the qualitative isolation scheme alone, we could not have isolated fault C_2^{-a} and C_2^{-i} from amongst $\{C_2^{-a}, C_2^{-i}, R_2^{+a}, R_2^{+i}\}$. Similarly, TRANSCEND would not have been able to discriminate between C_3^{-i} and R_4^{+i}.

Table 16: Results of Centralized Diagnosis Experiments on the Twelfth-Order Electrical Circuit

Experiment	Fault	Magnitude	Detection Time (s)	Isolation Time (s)	Convergence Time (s)	% Mean Absolute Error
1	C_2^{-a}	-0.90	1.26	53.80	4.76	0.27
2	C_2^{-i}	-0.55	32.82	53.80	11.52	6.05
3	L_3^{-a}	-0.90	0.50	3.98	5.08	0.49
4	L_3^{-i}	-0.05	49.62	82.9	12.40	5.50
5	C_3^{-a}	-0.90	0.2	2.8	3.26	0.12
6	R_7^{+a}	$+5.00$	196.8	377.4	115.6	0.48

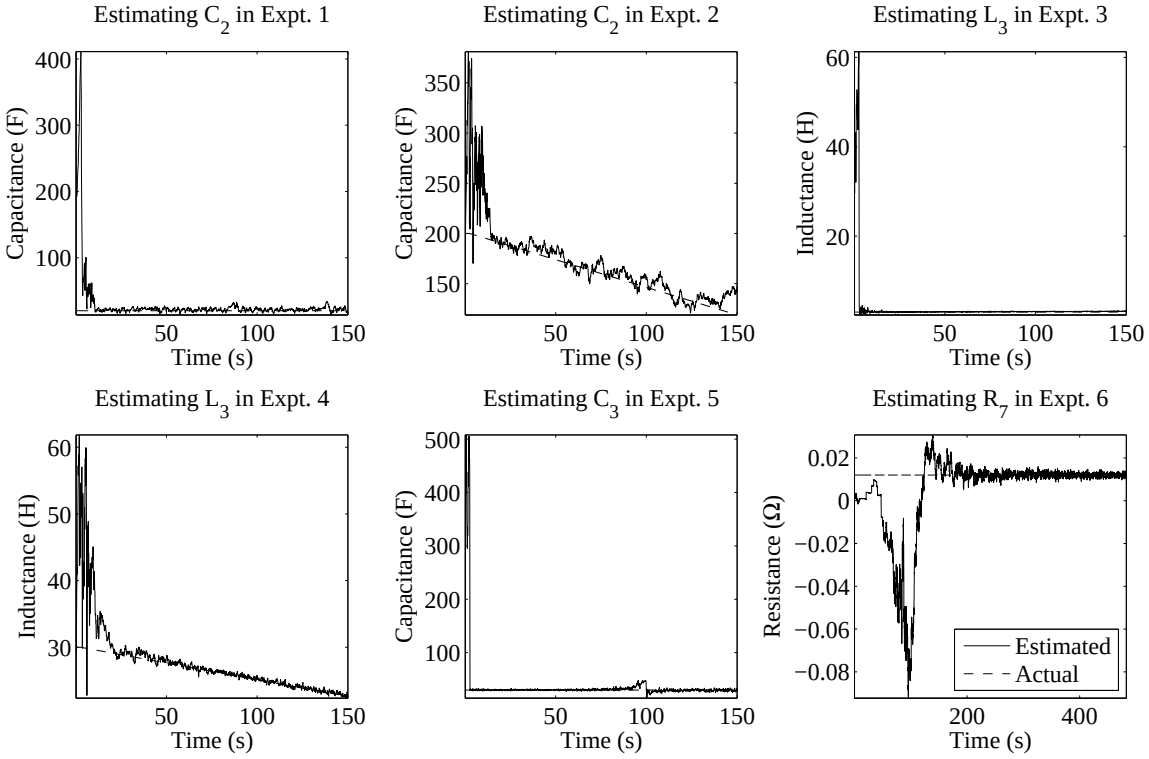


Figure 58: Parameter identification results for electrical circuit example.

Discussion and Summary

Particle filter estimators have been used extensively for system health monitoring and diagnosis of continuous and hybrid systems [13,58]. The general approach involves the system to include discrete nominal and fault modes, with the evolution of the system in each discrete mode being defined using differential equations. The process of diagnosis then involves tracking the observed measurements using a PF that runs on the comprehensive system model till the particles eventually converge to a discrete fault mode. PFs have also been used to diagnose parametric incipient and abrupt faults [77]. The usual approach for using PFs for diagnosis, however, cannot alleviate the problem of *sample impoverishment*, wherein particles in faulty state (with typically very low probability, and hence low

weights) are dropped during the re-sampling process. Even though several solutions to this problem have been proposed [4], the diagnosis scheme still has to rank the different fault hypothesis based on their likelihoods, and report the most likely fault mode that justifies the observations best. Our single fault assumption allows us to reduce the effects of the sample impoverishment problem to some extent by having a separate fault model for each fault hypothesis. Also, we do not rank the different fault hypotheses, and drop candidates based on their inability to track the observed faulty measurements. However, the issue of sample impoverishment remains when we need to estimate the unknown faulty parameter correctly, as discussed above.

In [116], the authors propose a combined qualitative-quantitative diagnosis scheme, where they combine look-ahead Rao-Blackwellised PFs (RBPFs) with the consistency-based Livingstone 3 (L3) approach for diagnosing faults in hybrid systems. In this approach, the nominal RBPF-based observer tracks the system evolution till a fault is detected, after which L3 generates a set of fault candidates that are then tracked by the fault observer (another RBPF). All the fault hypotheses are included in the same model, and tracked by the fault observer. In contrast, our approach executes the qualitative and quantitative fault isolation schemes in parallel, and uses separate fault models for each fault candidate.

In [111], the authors propose an approach for efficient combined state and parameter estimation based on the auxiliary particle filtering algorithm [12, 112]. The approach presented in [111] avoids the problem of loss of information through kernel smoothing of parameters, and gradually decreasing the variance of the unknown parameter as time progresses. In this paper, the authors emphasize that assuming constant parameters to be time varying, and applying standard state estimations approaches for combined parameter and state estimation, results in a loss of information, and may not give accurate estimates of the unknown "constant" parameter. While this approach is applicable to estimating abrupt fault parameters, we cannot directly apply this approach if the parameter is time varying, as is the case for incipient faults. Therefore, we have adapted the approach of using decreasing variance to apply to both incipient and abrupt fault scenarios within a common framework.

In the next chapter, we improve upon this centralized Bayesian diagnosis scheme by distributing the diagnosis task amongst several distributed diagnosis. Just like in Chapter IV, our Bayesian diagnosis approach necessitates careful construction of the distributed diagnosers to guarantee that each distributed diagnoser generates globally correct diagnosis results through local analysis, without the need of a centralized coordinator, without the exchange of partial diagnosis solutions amongst the individual diagnosers, and through minimal communication of measurement values amongst

themselves. We achieve this by partitioning the global system DBN into DBN factors that are decoupled across time, such that the random variables in each factor is conditionally independent of the variables in all other factors, given some shared measurements; and invoking our Bayesian diagnosis approach on each DBN factor independently. Also to ensure accurate state estimation, we analyze the structural observability properties of system bond graph to guarantee that these factors not only possess the structural observability property individually, but together they retain the observability properties of the global system.

CHAPTER VI

DISTRIBUTED BAYESIAN DIAGNOSIS OF CONTINUOUS SYSTEMS

In this chapter, we present a distributed Bayesian scheme for the diagnosis of incipient and abrupt faults. This work extends the DBN-based approach presented in Chapter V to a distributed diagnosis scheme, where the qualitative diagnoser for each distributed diagnoser satisfies all of the properties of the distributed diagnosers of Chapter IV, and each of the corresponding DBN-based local quantitative diagnosers also operates independently of the other diagnosers. Together, the set of distributed diagnosers can diagnose all single faults of interest in the DBN. This distributed diagnosis approach addresses the drawbacks of our centralized Bayesian diagnosis approach, such as single points of failures and poor scalability as systems become larger and more complex. In addition, the distributed scheme is more computationally efficient as compared to the centralized scheme, but without compromising on accuracy of the diagnosis results. Fig. 59 shows the architecture of the distributed diagnosis scheme that we have developed and tested in this chapter.

The basis of our diagnoser design is a procedure for *factoring* the global system DBN into DBN factors (DBN-Fs), such that the random variables in each DBN-F are conditionally independent of the random variables in all other DBN-Fs, given some subset of communicated measurements. The factors are generated by expressing some of the state variables as algebraic functions of the subset of chosen measurements, and making these variables as "input" nodes to the DBN factors. As a result, a number of across time causal links that are primarily responsible for the exponential nature of the computations are eliminated. It is the removal of these across-time links that results in the conditional independence of variables in DBN-Fs given the set of original and newly created input variables. The conditional independence amongst the random variables between the DBN-Fs makes it possible to invoke an independent DBN-diagnoser on each factor, such that each diagnoser takes on the form of the centralized diagnoser discussed in Chapter V. Independence among the factors is achieved by converting some of the system measurements to input variables and replacing the across-time causal links to these state variables with intra-time causal links from the inputs to these state variables. In this chapter, we establish that the independent estimation algorithms for each DBN-F generates correct and accurate inference results by guaranteeing that each DBN-F, by construction, represents a structurally observable subsystem, and together, the DBN-Fs retain the structural observability property of the entire system. We determine if a subsystem represented

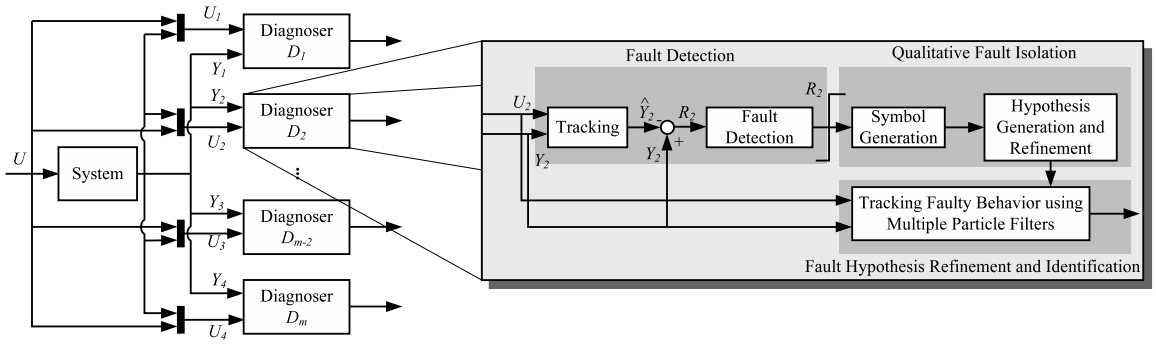


Figure 59: The distributed diagnosis architecture.

by a DBN-F is structurally observable by analyzing the bond graph factor that corresponds to the DBN-F.

This chapter is organized as follows. We start by formulating the diagnoser design problem by factoring the global system DBN, and then present the design algorithms for generating the distributed diagnosers. The next section presents how each distributed diagnoser is implemented based on a DBN-F, and establishes the proof that our distributed diagnosers generate globally correct diagnosis through local analysis. We then present some experimental results to demonstrate the effectiveness of our factored estimation and distributed diagnosis scheme. We conclude the chapter with a comparison of our distributed Bayesian diagnosis scheme to some related work, followed by a discussion of the results, and summary of accomplishments of this approach.

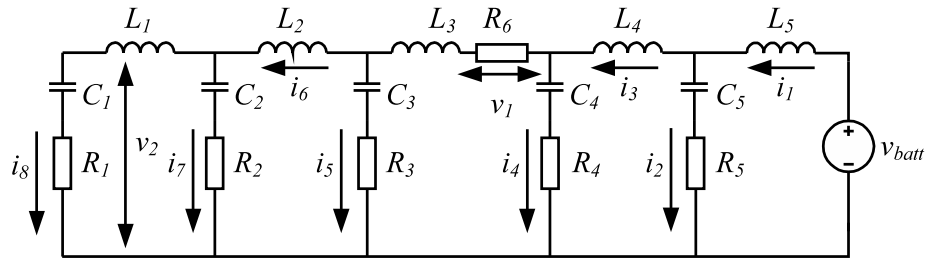
Formulating the Design Problem for Distributed Diagnosis

Formally, we define our diagnoser design problem as follows:

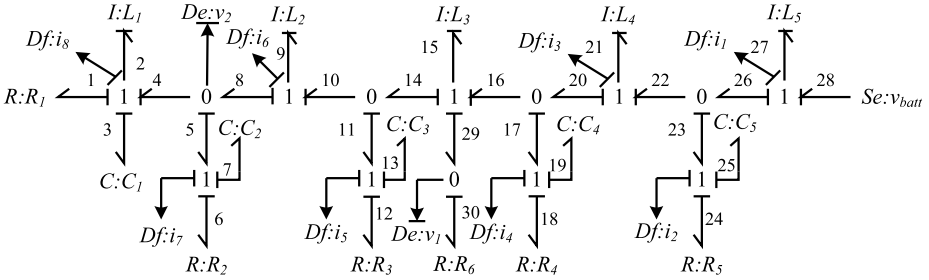
Problem 3. Given a system S modeled using a global DBN $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, partition D into the *maximal* number of *conditionally independent* DBN Factors (DBN-Fs), $D_i = (\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i)$, $i \in [1, m]$, such that each DBN-F is *observable*. Then, once generated, implement a combined qualitative-quantitative diagnoser for each DBN-F.

Observability and conditional independence of each DBN-F are necessary conditions for ensuring efficient and accurate state estimates when the estimation algorithm is applied to each DBN-F separately. If a system is not observable, there is no guarantee that the estimates obtained from the particle filtering-based estimation schemes are correct.

In the following, we formally define DBN-Fs and the notions of observability and conditional independence in the context of DBN-Fs. We illustrate these concepts using a tenth-order electrical



(a) Schematic.



(b) Bond graph.

Figure 60: Models of the tenth-order electrical system.

circuit shown in Fig. 60(a). The observed measurements in this electrical circuit are the currents $i_1, i_2, \ldots, i_8$, and the voltages v_1 and v_2. The battery v_{batt} drives the circuit.

Definition 12 (DBN Factor)**.** A *DBN Factor* (DBN-F), $D_i = (\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i)$, $i \in [1, m]$, of DBN $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$ is a subset of DBN, D, such that (i) $\bigcup \mathbf{X}_i \subset \mathbf{X}$, (ii) $\bigcup \mathbf{Y}_i \subset \mathbf{Y}$, (iii) $\bigcup \mathbf{U}_i = \mathbf{U} \bigcup (\mathbf{Y} - \cup \mathbf{Y}_i)$, and (iv) every $\mathbf{X}_i$ and $\mathbf{Y}_i$ in D_i is *conditionally independent* from other random variables in all other DBN-Fs, given the inputs, $\mathbf{U}_i$.

The measured variables, $\mathbf{Y} - \cup \mathbf{Y}_i$, denote the subset of measurements that are used as inputs, and used to compute the values of states $\mathbf{X} - \cup \mathbf{X}_i$. The method for deriving this subset is discussed later in the chapter. In order to be effective for distributed diagnosis, and ensure correct estimation of state variables, we assume that the measurement sensors do not become faulty.

Assumption 2. Given a DBN-F, D_i, we assume that the set of measurements that are treated as input variables, i.e., $(\mathbf{U} - \mathbf{U}_i) \cap (\mathbf{Y} - \cup \mathbf{Y}_i)$ have no faults. However, these sensors may be noisy.

Definition 13 (Conditionally Independent DBN-F)**.** Any DBN-F, $D_j = (\mathbf{X}_j, \mathbf{U}_j, \mathbf{Y}_j)$, of a global DBN, $D = (\mathbf{X}, \mathbf{U}, \mathbf{Y})$, is *conditionally independent* from all its other DBN-Fs $D_k = (\mathbf{X}_k, \mathbf{U}_k, \mathbf{Y}_k)$, s.t. $k \neq j, k \in [1, m]$ given $\mathbf{U}_j$ if (i) $P(\mathbf{X}_{j_{t+1}} | \mathbf{X}_{t-n:t}, \mathbf{U}_{t-n:t}) = P(\mathbf{X}_{j_{t+1}} | \mathbf{X}_{j_{t-n:t}}, \mathbf{U}_{j_{t-n:t}})$, and (ii) $P(\mathbf{Y}_{j_t} | \mathbf{X}_t, \mathbf{U}_t) = P(\mathbf{Y}_{j_t} | \mathbf{X}_{j_t}, \mathbf{U}_{j_t})$.

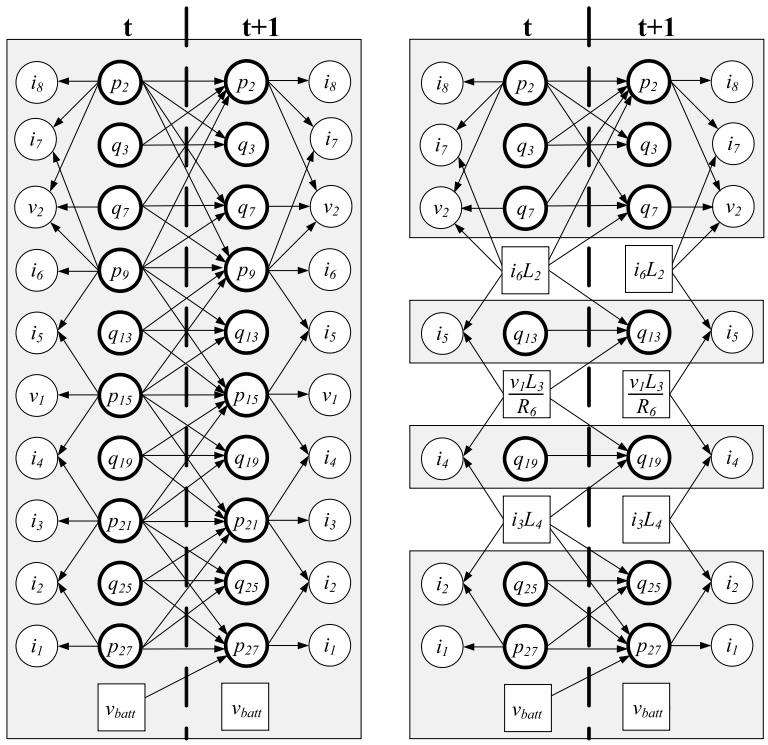
Example. Fig. 61(a) presents the global DBN, D, of the tenth-order electrical system. This DBN is generated systematically from the bond graph of the tenth-order electrical system shown in Fig. 60(b) using the approach presented in Chapter V. Fig. 61(b) shows four DBN-Fs, $D_1 = (\{p_2, q_3, q_7\}, \{i_6 L_2\}, \{i_8, i_7, v_2\})$, $D_2 = (\{q_{13}\}, \{i_6 L_2, \frac{v_1 L_3}{R_6}\}, \{i_5\})$, $D_3 = (\{q_{19}\}, \{\frac{v_1 L_3}{R_6}, i_3 L_4\}, \{i_4\})$, and $D_4 = (\{q_{25}, p_{27}\}, \{i_3 L_4, v_{batt}\}, \{i_2, i_1\})$, of the global DBN, D, shown in Fig. 61(a), such that the conditions of Definition 12 are fulfilled, i.e.,

$$\bigcup_{i\in[1,4]} \mathbf{X}_i \subset \mathbf{X}, \quad \bigcup_{i\in[1,4]} \mathbf{Y}_i \subset \mathbf{Y}, \quad \bigcup_{i\in[1,4]} \mathbf{U}_i = \mathbf{U} \bigcup (\mathbf{Y} - \cup \mathbf{Y}_i),$$

and, each of the four DBN-Fs shown in Fig. 61(b) is conditionally independent of all other DBN-Fs. For example, DBN-F D_2, shown in Fig. 61(b), is conditionally independent of all the other DBN-Fs, D_1, D_3 and D_4 given the input nodes $i_6 L_2$ and $\frac{v_1 L_3}{R_6}$ because the values of the single state variable in D_2, q_{13}, and measurement variable, i_5, at time t, do not depend on any variable external to D_2. Recall that we assume that there are no errors in the measurements i_6, v_1 and i_3, but they may be noisy. Note that the conditional independence is established as a result of converting measurement variables to inputs and the resultant factoring. The conditional independence relations did not exist in the unfactored global DBN shown in Fig. 61(a), and the value of variable q_{13} at time step $t+1$ depends on variables p_9, q_{13}, and p_{15} at time step t, and variables p_2 and p_7, among others, at time step $t-1$, and so on. The factoring of a DBN is not unique, and multiple factorings may exist for a DBN. Fig. 61(c) and Fig. 61(d) show other possible factorings of the full DBN.

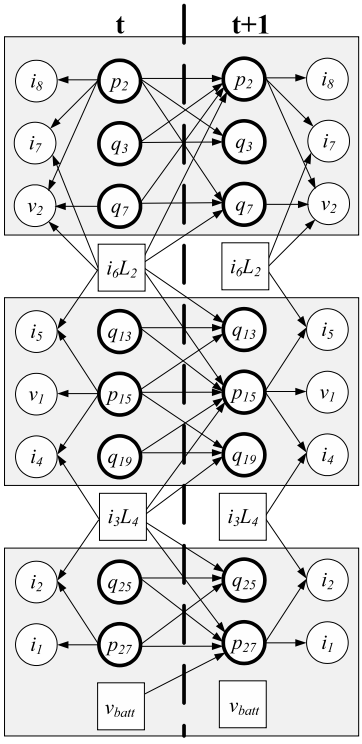
Definition 14 (Observable DBN-F)**.** A DBN-F, $D_j = (\mathbf{X}_j, \mathbf{U}_j, \mathbf{Y}_j)$ is *observable* if the subsystem it models is *structurally observable*.

Example. Consider the four DBN-Fs shown in Fig. 61(b). As we will show in the next section, each DBN-F has a one-to-one mapping with a bond graph factor (BG-F). If the BG-F corresponding to a DBN-F is analyzed to be structurally observable (as explained in the previous chapter), we say that the DBN-F is observable. The four DBN-Fs shown in Fig. 61(b) corresponds to the BG-Fs shown in Fig. 62. The two outer BG-Fs are structurally observable, as all their energy storage elements can be assigned a preferred derivative causality (albeit by dualizing an effort sensor into a flow sensor, indicated by the shaded background, in the first BG-F), and every state variable affects at least one sensor. The two BG-Fs in the middle, however, are not observable, since, in each of these two BG-Fs, the single state variable does not causally affect the flow sensor (whose value is determined by the two flow sources on the 0-junction).

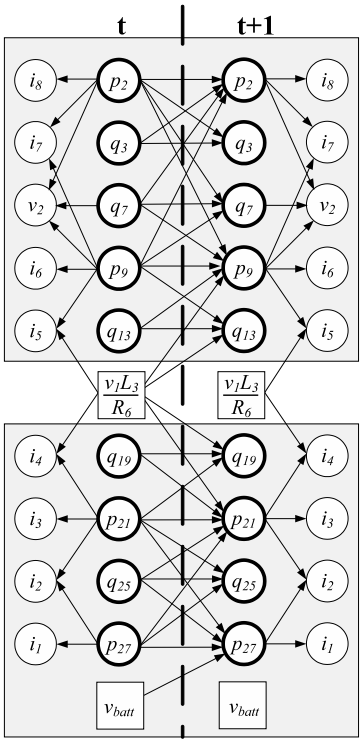


(a) Full DBN.

(b) 4-factored DBN.



(c) DBN with 3 factors.



(d) 2-factored DBN.

Figure 61: Factorings of the electrical system DBN.

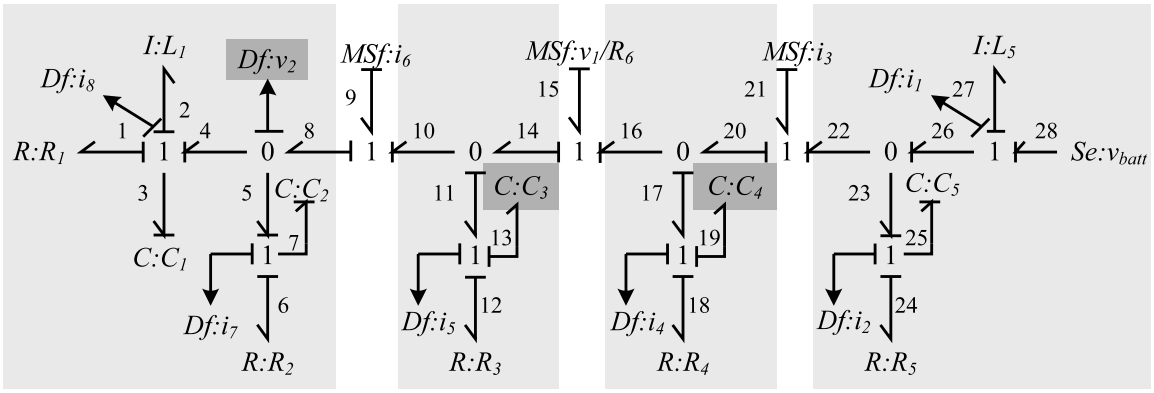


Figure 62: Four-factored bond graph of the electrical circuit with imposed derivative causality.

Designing the Distributed Diagnosers

Our distributed diagnoser design procedure consists of three main steps: (*i*) partitioning the system DBN into maximal number of conditionally independent DBN-Fs by output injection, i.e., replacing every state variable which can be determined as an algebraic function of at most r output measurements, where r is a user-specified parameter, (*ii*) mapping each generated DBN-F to a bond graph factor (BG-F) and analyzing the structure of this BG-F to determine if the DBN-F is observable, and (*iii*) merging every unobservable DBN-F with other DBN-Fs to possibly generate resultant observable DBN-Fs. Steps *ii* and *iii* are repeated till a factoring is obtained containing the most number of DBN-Fs that are *all* observable. These steps (shown in Algorithm 3) are presented in detail below. We assume that the system to be factored is globally observable, as otherwise, no factoring with only observable factors exist. Also, we assume that we have sufficient sensors to execute the factoring procedure.

Step 1 - Generating Maximal Factoring

Our procedure for factoring a DBN into maximal number of DBN-Fs involves replacing one or more of its state variables, each represented by X, by an algebraic function of at most r measured variables, $\mathbf{Y}_X^{\leq r}$, where r is a user-specified parameter. To identify state variables in the global DBN that can be replaced to generate DBN-Fs, we analyze the system DBN, and for every state variable, X, identify all single measurements, then pairs, triples, and so on, up to r measurements that this state X can be expressed in terms of. In other words, for every X, we determine

$$\mathbf{Y}_X^{\leq r} = \bigcup_{n=1}^{r} \mathbf{Y}_X^n,$$

Algorithm 3 Generating factors of a DBN.

Input: System DBN, D
Generate maximal $Factoring_1 = \{D_1, D_2, \ldots, D_n\}$
$SetOfFactorings = \{Factoring_1\}$
while true do
  $SetOfObsF = \varnothing$; $SetOfUnobsF = \varnothing$;
  for each $Factoring_i \in SetOfFactorings$ **do**
    if every DBN-F in $Factoring_i$ is observable **then**
      $SetOfObsF = SetOfObsF \cup Factoring_i$
    else
      $SetOfUnobsF = SetOfObsF \cup Factoring_i$
  if $SetOfObsF \neq \varnothing$ **then**
    $BestFactoring = Factoring_j \in SetOfObsF$ having the most number of balanced DBN-Fs
    exit
  else
    $NextBestFactoring = Factoring_j \in SetOfUnobs$ having the most number of unobservable DBN-Fs
  $SetOfFactorings =$ all possible pairwise mergings of the DBN-Fs of $NextBestFactoring$

where, $\mathbf{Y}_X^n$ denotes sets of exactly n measurements that can be used to compute the value of state variable X at every time step. Given n, and $\mathbf{Y}_X^n = h(\mathbf{X})$, we replace $\mathbf{X} = h^{-1}(\mathbf{Y}_X^n)$ if h is an invertible function. To compute X, we may need to solve multiple equations that involve more than one measurement variables.

Example. If $n = 1$, a state variable, X, can be replaced by an algebraic function of measurement Y, if and only if, X is the only state variable that has an intra-time slice link to Y. For example, in Fig. 61(b), $\mathbf{Y}_{p_2}^1 = \{i_8\}$, since p_2 is the only state variable causally affecting i_8 within a time step. Similarly, for $\mathbf{Y}_{p_9}^1 = \{i_6\}$, and so on. Note however, that $\mathbf{Y}_{q_7}^1 \neq \{v_2\}$ since v_2 depends on q_7, as well as, p_2 and p_9. The determination of $\mathbf{Y}_X^n$ for $n > 1$ involves solving a system of n simultaneous equations. For example, if $n = 2$, to determine $\mathbf{Y}_{p_{27}}^2$, we must solve the two equations

$$\begin{aligned} i_3 &= p_{21}/L_4 \\ i_2 &= p_{27}/L_5 - p_{21}/L_4, \end{aligned}$$

simultaneously to determine

$$p_{27} = (i_2 + i_3)L_5.$$

Hence, $\mathbf{Y}_{p_{27}}^2 = \{i_2, i_3\}$. For practical purposes, in this work, we mostly restrict the value of r to 1. If no observable factorings are possible with $r = 1$, we relax r to be 2. If r is assigned values 3 or above, the determination of $\mathbf{Y}_X^{\leq r}$ quickly becomes very involved.

For each state variable, X, for which set $\mathbf{Y}_X^{\leq r} \neq \varnothing$, we identify its smallest element $[\mathbf{Y}_X^{\leq r}]_{min}$, and express $X = h^{-1}([\mathbf{Y}_X^{\leq r}]_{min})$, considering the measurements in $[\mathbf{Y}_X^{\leq r}]_{min}$ to be inputs. Next, we delete every $X_t \rightarrow X_{t+1}$, $U_t \rightarrow X_{t+1}$, $X_t \rightarrow Y_t$ link, replace X with $h^{-1}([\mathbf{Y}_X^{\leq r}]_{min})$, and restore an intra-time slice link $h^{-1}([\mathbf{Y}_X^{\leq r}]_{min}) \rightarrow Y_t$ for every $X_t \rightarrow Y_t$, such that $Y_t \notin [\mathbf{Y}_X^{\leq r}]_{min}$. The across-time links into X_t are not restored, since $h^{-1}([\mathbf{Y}_X^{\leq r}]_{min})$ can be computed independently at each time step. If there are more than one possible choice for $[\mathbf{Y}_X^{\leq r}]_{min}$, we choose that value of $[\mathbf{Y}_X^{\leq r}]_{min}$, which has maximum overlap with other $[\mathbf{Y}_{X'}^{\leq r}]_{min}$, for states $X' \neq X$. We assume that the components associated with each algebraic function, h, do not become faulty during diagnostic analysis.

Assumption 3. We assume that any component parameter associated with the algebraic function, h, whose inverse is used to determine the state X in terms of $[\mathbf{Y}_X^{\leq r}]_{min}$, i.e., $X = h^{-1}([\mathbf{Y}_X^{\leq r}]_{min})$, cannot be faulty.

Once every state variable X with a nonempty $\mathbf{Y}_X^{\leq r}$ is replaced, the system DBN is *maximally factored* into DBN-Fs, if the removal of the across-time links involving the replaced state variables results in the random variables in each DBN-F being conditionally independent of variables in all other DBN-F, i.e., no random variable in a DBN-F has a causal link to another variable in any other DBN-F, and vice versa. To test for the conditional independence, we check if the removal of the newly introduced "input" nodes decomposes the global DBN into disconnected subgraphs. Each disconnected subgraph, along with its corresponding input nodes result in a DBN-F.

Example. Consider the DBN shown in Fig. 61(b). If we assume $r = 1$, since voltage $v_1 = h(p_{15}) = p_{15}R_6/L_3$, we can determine the value of the state variable p_{15} as an algebraic function of voltage v_1, i.e., $p_{15} = h^{-1}(v_1) = v_1L_3/R_6$. Therefore, we can replace p_{15} with v_1L_3/R_6. Since, we no longer need the variables p_9, q_{13}, p_{15}, q_{19}, and p_{21} to compute p_{15}, the across-time links to p_{15} can be removed. Similarly, we can replace state variables p_9 and p_{21} with algebraic functions of i_6 and i_3, respectively, since $p_9 = i_6L_2$ and $p_{21} = i_3L_4$. These replacements result in the maximally factored DBN with four DBN-Fs, where, each DBN-F is conditionally independent of all other DBN-Fs, given measurements i_6, v_1, and i_3, as defined in Definition 13.

Note that there are two situations in which a state variable is not removed from a global DBN: (i) if the removal of this state variable does not generate any new factors, e.g., the state variables p_2 and p_{27} are not replaced by functions of i_8 and i_1, respectively, as that would not generate any more factors in Fig. 61(b), and (ii) any parameter involved in the computation of the state variable

is a potential fault hypothesis, e.g., we could not have replaced the state variable p_{15} with $v_1 L_3/R_6$ if either L_3 or R_6 represent components that could fail during system operation.

Step 2 - Testing Observability of DBN-Fs

Given a DBN-F, D_i, we can test whether or not it is observable by mapping D_i to its corresponding BG-F, B_i, and analyzing B_i for structural observability. Before mapping a D_i to a B_i, we identify the state variables in the global DBN that were removed to generate D_i, and the measurement variables these state variables were replaced with. Given this information, the first step of mapping a D_i to a B_i is to replace the I or C element (in the global bond graph) corresponding to every state variable that was removed from the global DBN to generate D_i by a MSf or MSe element, respectively, and the value of its outgoing flow or effort is computed from at most r measurements.

Definition 15 (Bond graph factor)**.** We define a *bond graph factor* (BG-F), B_i, as a subgraph of connected bond graph elements such that a subset of its state variables, i.e., the corresponding energy-storage elements, are replaced by modulated sources of effort or flow.

Proposition 1. A bond graph may be factored into *independent* bond graph fragments $B_1, B_2, \ldots, B_n$ by replacing an I or C element with a MSf or MSe element, respectively.

Proof: A capacitor C_1's constituent equation is $\dot{q}_{C_1} = \int f_{C_1} dt$. In the state-space formulation, f_{C_1} can be expressed in terms of other state variables. Therefore, any measurement or state variable that depends on q_{C_1} would, in turn, be dependent on f_{C_1}, and hence, possibly every other state variable. Now, if f_{C_1} can be measured, and we replace C_1 with modulated $Se_{C_1} = h^{-1}(f_{C_1})$, the dependence between q_{C_1} and all other state variables is broken, and the bond graph is factored into *independent* BG-Fs. The proof similarly follows for an I element replaced with a modulated Sf. ■

Example. The maximally factored DBN for the tenth-order electrical circuit has four DBN-Fs (see Fig. 61(b)), which correspond to the BG-Fs shown in Fig. 62. The two outer BG-Fs are structurally observable, as all their energy storage elements can be assigned preferred derivative causality (albeit by dualizing an effort sensor into a flow sensor, indicated by the shaded background, in the first BG-F), and every state variable affects at least one sensor. The two BG-Fs in the middle, however, are not observable, since, in each of these two BG-Fs, the single state variable does not causally affect the flow sensor (whose value is determined by the two flow sources on the 0-junction). Hence, in the maximally factored DBN shown in Fig. 61(b), the two outer DBN-Fs are observable, while the two middle DBN-Fs are not. Therefore, the middle DBN-Fs, if used for state estimation, may not generate accurate state estimates.

Step 3 - Merging Unobservable Factors

We propose *merging* of two or more unobservable DBN-Fs to generate an observable DBN-F. m DBN-Fs, $D_1, D_2, \ldots, D_m$, can be merged by restoring those state variables in the system DBN that were replaced to generate $D_1, D_2, \ldots D_m$, redrawing the across-time causal links involving these state variables, and reintroducing the measurements that were used to compute these state variables.

An unobservable DBN-F can be merged with another DBN-F (observable or otherwise) to generate a resultant DBN-F that is observable. A DBN-F is unobservable if its corresponding BG-F does not satisfy the conditions of structural observability (described in Chapter V). An unobservable DBN-F, D_1, when merged with another DBN-F, D_2, generates the resultant DBN-F, $D_{1,2}$, which maps to the BG-F, $B_{1,2}$. The merging of D_1 and D_2 maps to the replacement of at least one source element in B_1 and B_2 with a I or C element, and reintroduction of at least one sensor element in the resultant $B_{1,2}$. The measurement sensors that are reintroduced are directly connected to the reinstated I or C elements in $B_{1,2}$. Hence, condition 1 of structural observability is satisfied for these reintroduced energy storage elements. Moreover, the new sensor can be causally linked to other I or C elements that are not linked to any sensor element, further aiding the satisfaction of condition 1 for $B_{1,2}$. Moreover, the greater are the number of sensors in $B_{1,2}$, the greater is the flexibility for dualizing these sensors to satisfy condition 2.

As shown in Algorithm 3, the merging procedure is invoked if any DBN-F in the maximally factored DBN is not observable. At every iteration of merging, we create new factorings through all possible pairwise mergings of unobservable DBN-Fs into other DBN-Fs, with the goal of creating at least one new partition where all the DBN-Fs are observable. If multiple such factorings get created, we use a heuristic to choose that factoring which has the most number of *balanced* DBN-Fs, determined by comparing how close the number of state variables in each of its DBN-F is to the average number of state variables per DBN-F. If the merging step does not generate any factorings where all DBN-Fs are observable, we select the maximal factoring with the largest number of factors and highest number of unobservable DBN-Fs, and generate the next set of factorings by pairwise merging of unobservable DBN-Fs. This procedure is repeated till we obtain at least one factoring with all its DBN-Fs being observable. Since the system was initially observable, continued merging will eventually result in a factoring in which all DBN-Fs are observable, in the worst case we will end up with the original DBN. Therefore, our factoring algorithm terminates. However, when it is possible, our algorithm uses additional heuristics to produce multiple balanced DBN-Fs.

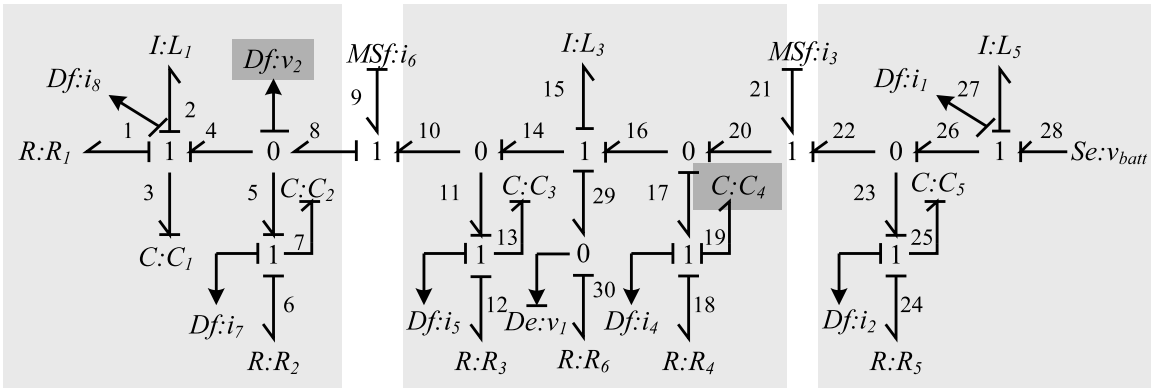


Figure 63: Three-factored bond graph of the electrical circuit with imposed derivative causality.

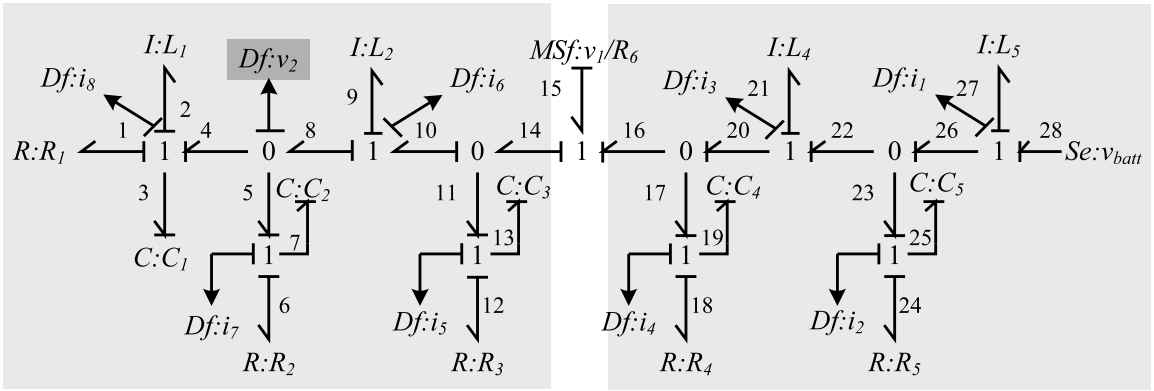


Figure 64: Two-factored bond graph of the electrical circuit with imposed derivative causality.

Example. The BG-Fs corresponding to the DBN-Fs in the maximally factored DBN are shown in Fig. 62. The two unobservable BGs in the middle can be merged in two possible ways, as shown in Figs. 63 and 64. The factoring shown in Fig. 63 corresponds to a DBN-F generated by merging the two central DBN-Fs (see Fig. 61(c)), and is not observable (since a consistent causal assignment is not possible in the middle BG-F after assigning capacitor C_4 a derivative causality) However, the two BG-Fs shown in Fig. 64 correspond to the DBN-Fs shown in Fig. 61(d), are observable, and hence, we select this as our desired factoring.

Once a system DBN is factored into m DBN-Fs, D_1, D_2, ... D_m, we construct a distributed diagnoser, D_i, based on every DBN-F D_i. A diagnoser D_i is responsible for diagnosing faults F_i based on its observations $\mathbf{U}_i$.

Implementing the Individual Distributed Diagnosers

Recall the computational architecture of our distributed Bayesian fault diagnosis approach shown in Fig. 59. Each distributed diagnoser, D_i, essentially implements the combined qualitative-quantitative scheme for diagnosing incipient and abrupt faults. However, instead of using a single diagnoser that operates on the complete system, in our distributed approach, each diagnoser operates with a DBN-F. Hence, each D_i receives the input signals $\mathbf{U}_i$, and the observed measurements $\mathbf{Y}_i$ from the system. Note that a diagnoser D_i's inputs $\mathbf{U}_i$ may include some of the inputs to the global system, i.e., $\mathbf{U}_i \cap \mathbf{U} \neq \varnothing$, as well as some measurements now considered inputs, i.e., $\mathbf{U}_i \cap \mathbf{Y} \neq \varnothing$. Two diagnosers D_j, D_k *communicate* a measurement $Y \in \mathbf{Y}$ if $Y \in \mathbf{U}_j \wedge Y \in \mathbf{U}_k$, i.e., measurement Y is an input to both D_j and D_k.

Each diagnoser D_i implements an independent particle filter-based observer for DBN-F D_i. Each of these particle filters takes as inputs, $\mathbf{U}_i$, and estimates $\mathbf{X}_i$ based on $\mathbf{Y}_i$. Only measurements $(\cup_i \mathbf{U}_i) - \mathbf{U}$ are shared between the particle filter-based observers for each D_i. Further, the particle filter for the DBN-F D_i is designed to use $a\frac{|\mathbf{X_i}|}{|\mathbf{X}|}$ particles, where a is a user-specified parameter. Given m DBN-Fs, we know that $\sum_i |\mathbf{X_i}| < |\mathbf{X}|$, where $\mathbf{X}$ is the total number of state states in the complete system. Therefore, the complexity of tracking using each DBN-F is less that that of tracking using the global DBN. Also, since the inference algorithms on the different factors are executed simultaneously, the total complexity of inference reduces to the complexity of inference of the particle filter with the maximum number of particles. The reduction of complexity is based on Assumptions 2 and 3, i.e., the sensors associated with measurements converted to inputs will not be faulty, and the components whose parameters are used in the algebraic functions are assumed not to fail. Therefore, there is a trade-off for robustness to gain efficiency.

As explained earlier in this chapter, each of these distributed particle filters corresponding to a DBN-F D_i can track nominal system behavior, and be linked to a local fault detector. Qual-FI is performed using the subset of measurements for each D_i. Quant-FHRI applies the same approach as centralized DBN diagnosis by including fault variables as extra state variables. The particle filters for each factor can be run in parallel on separate processors thus providing significant speed up. The observers are independent of one another, and this independence is guaranteed by construction, as discussed earlier. This independent execution of the observers in each diagnoser results in the following property.

Lemma 1. *The failure of one of the observers will not affect the quality of state estimates at other observers.*

Once a fault in F_j is detected in any one diagnoser D_j, first the Qual-FI is initiated, followed by Quant-FHRI till the true fault is diagnosed. Given the way the DBN-Fs are constructed, we can argue that our distributed diagnosers fulfill the following property.

Lemma 2. *A fault $\phi \in F_j$ is only detected by diagnoser D_j, and all other diagnosers, D_k, $k \neq j$, will not detect the fault. Hence, they are not activated, even though the effect of fault ϕ may propagate to all other factors.*

Proof: Every DBN-F D_i has a one-to-one mapping to a BG-F B_i. A diagnoser D_i is activated when it detects a fault. In general, let us assume that the observer in diagnoser D_i uses the state space equations $\hat{\mathbf{X}}_{i_{t+1}} = G_i(\mathbf{X}_{i_t}, \mathbf{U}_{i_t})$, and $\hat{\mathbf{Y}}_{i_t} = H_i(\mathbf{X}_{i_t}, \mathbf{U}_{i_t})$. Let us now assume that there is a fault in BG-F B_k. This means that functions G_k and H_k do not correctly represent the actual system any more. As a result, $\hat{\mathbf{Y}}_k \not\approx \mathbf{Y}_k$, and a fault is eventually detected by D_k. The effects of a fault in B_k can propagate to another BG-F B_j, $j \neq k$, through their shared inputs, $(\mathbf{U}_j \cap \mathbf{U}_k) - \mathbf{U}$, iff B_k and B_j communicate at least one measurement, i.e., $(\mathbf{U}_k \cap \mathbf{U}_j) - \mathbf{U} \neq \varnothing$. But, since we adopt the single-fault assumption, and since by construction, two BG-Fs can never share any parameters, the state space representations G_j and H_j of all other BG-Fs, B_j, $j \neq k$, will correctly represent the actual system dynamics of each BG-F. Hence, $\hat{\mathbf{Y}}_j \approx \mathbf{Y}_j$, i.e., the observers in other diagnosers will correctly track the faulty measurement, and hence no fault will be detected. Consequently, if a fault is not detected, the diagnoser will not be activated. ■

In the next section, we present two sets of experiments. The first set of experiments are implemented on the tenth-order electrical system we have been using as a running example throughout this chapter. These experiments focus on comparing the centralized versus distributed scheme in terms of computational efficiency and accuracy of estimates generated. We do not perform any diagnosis experiments on the tenth-order electrical system. Instead, we perform distributed diagnosis experiments on the twelfth-order electrical system that we used as a case study in Chapter V. The results of our distributed fault diagnosis experiments on the twelfth-order electrical system demonstrate that the distributed Bayesian diagnosers generate globally correct results through local analysis, and in many cases, result in faster isolation times and lower estimation errors, when compared to the results of applying our centralized Bayesian diagnosis scheme on the same twelfth-order electrical system in Chapter V.

Experimental Results for Distributed State Estimation Using DBN-Fs

This section presents experimental results that evaluate our distributed Bayesian state estimation scheme using DBN-Fs in comparison to the centralized estimation scheme that uses the global DBN. Our objective is to determine if the factored scheme observable DBN-Fs results in improved computational efficiency in state estimation without compromising the accuracy of estimation.

For this experiment, we tracked the state variables using the DBN factorings shown in Fig. 61(d) for 10 runs. Given m DBN-Fs, $D_i = \{\mathbf{X}_i, \mathbf{U}_i, \mathbf{Y}_i\}$, $i = 1, 2, \ldots, m$, such that $\mathbf{X} = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \ldots \mathbf{X}_m$, for each run we computed the estimation error: $E = \frac{1}{|\mathbf{X}|} \sum_{X \in \mathbf{X}} \left(\frac{1}{T} \sum_{t=0}^{T} (X_t - X_t^{model})^2 \right)$, where T is the total simulation time, X_t denotes the estimated value of state X at time t, and X_t^{model} denotes the actual value of state X at time t obtained from the simulation model. Table 17 reports the mean and standard deviation of errors obtained from each factoring over all runs. In this experiment, all prior and conditional probabilities are assumed to be Gaussian, and all sensors have associated white Gaussian noise with 0 mean and 1 W variance.

To demonstrate that the factoring scheme preserves the system dynamics, we hypothesized the accuracy differences measured by the error in estimation for the 2-factor and unfactored DBN would not be statistically significant, and the error for the 4-factor DBN (which is observable) would be significantly larger than the unfactored DBN. Further the difference in error for the 2-factor and 4-factor DBNs would also be statistically significant. We ran t-tests to establish significance of the differences in the error in estimating the state. The tests for significance indicated that the errors obtained using the 2-factor DBN did not significantly differ from that obtained using the unfactored DBN ($p < 0.05$), while those obtained using the 4-factor DBN was significantly greater ($p < 0.05$). The test of significance between the 2- and 4-factor DBN showed that the error in the 4-factor DBN was significantly larger ($p < 0.05$). Therefore, we conclude that the 2-factor DBN preserves dynamics of the unfactored DBN, whereas the 4-factor DBN, which has unobservable factors, does not.

Table 18 shows the average time taken by the slowest particle filter for each factoring to track system behavior for 1500 time steps. The time taken by a particle filter depends on the number of particles it uses. In our experiments, the number of particles used by a particle filter was proportional to the number of states in the DBN factor the particle filter was associated with. Hence, the particle filter for unfactored DBN (with 1000 particles) took the most time, followed by the particle filter on the larger DBN-F of the 2-factor DBN (with 500 particles). The least amount of time was taken by the particle filters applied to the 4-factor DBN, since its largest DBN-F has 3 state variables, and

Table 17: Average mean squared error and standard deviation over all state variables across 10 runs

No. of Factors →	1	2	4
Mean	0.1143	0.1381	0.1968
(Standard Deviation)	(0.0360)	(0.0470)	(0.0314)

Table 18: Time taken for particle filter to complete estimation

No. of Factors →	1	2	3	4
Time (s)	137.03	37.74	18.79	18.97

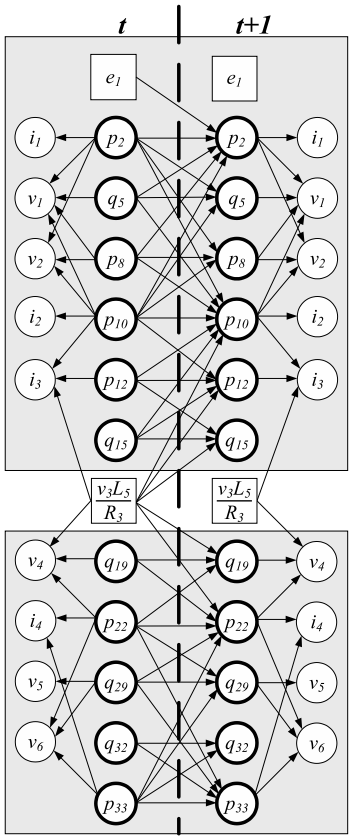


Figure 65: Two-factored DBN for the twelfth-order electrical circuit.

hence, its particle filter used 300 particles. Thus, running Bayesian estimation schemes on factored DBN improves computational efficiency.

Case Study 2: Diagnosis Experiments on the Twelfth-order Electrical Circuit

In this section, we use the twelfth-order electrical circuit shown in Fig. 49(a) as a case study to demonstrate the distributed diagnosis scheme we have presented in this chapter. Recall that this circuit was used as a case study for the centralized diagnosis scheme presented in Chapter V. In this

Table 19: Fault Signatures for Diagnoser D_1

Fault	i_1	i_2	i_3	v_1	v_2
C_2^{-a}, C_2^{-i}, R_2^{+a}, R_2^{+i}	0−	0−	0−	0+	0+
L_2^{-a}	0+	0−	0−	−+	−+
L_2^{-i}	0+	0−	0−	0−	0−
L_3^{-a}	0+	+−	+−	−∗	−+
L_3^{-i}	0+	0+	0+	0−	0−
L_4^{-a}	0+	0+	−+	0−	0−
L_4^{-i}	0+	0+	0−	0−	0−

chapter, we first describe the result of our distributed diagnoser design algorithm, when applied to this electrical circuit, and then compare how our distributed Bayesian diagnosis scheme compares to the our centralized Bayesian diagnosis scheme in terms of fault detection and isolation times, and error in estimating the magnitude of the true fault parameter.

Distributed Diagnoser Design

The bond graph model of this electrical circuit is shown in Fig. 49(b). Fig. 50 shows global DBN of this circuit. The available measurements in this circuit are current values, $i_1, i_2, \ldots, i_4$ and voltages $v_1, v_2, \ldots, v_6$. Fig. 65 shows the DBN of the electrical circuit factored into two DBN-Fs. We assume $r = 1$ in the following. It is evident from Fig. 49(a) that the current through the inductor L_5 is equal to $v_3 L_5/R_3$. Hence, we can replace p_{35} in Fig. 50 with $v_3 L_5/R_3$, as shown in Fig. 65. Since, $v_3 L_5/R_3$ can be *measured* at every time step, all causal links into this node is removed. As a result, given $v_3 L_5/R_3$, every variable in one factor is conditionally independent of the variables in the other factor. Thus, two conditionally independent factors are generated. In the factored DBN, we do not replace state variables, such as, p_2 with $i_1 L_1$, since this replacement does not yield any additional factors in Fig. 65. Moreover, we do not replace state variables p_{10} and q_{29} with $i_2 L_3$ and $v_5 C_4$, respectively, since we assume that inductor L_3, and capacitor C_4 can become faulty. We can see that the DBN-Fs shown in Fig. 65 map to the BG-Fs shown in Fig. 66. Both the BG-Fs are structurally observable as they fulfill both the conditions necessary for structural observability mentioned in Chapter V. Note that the current sensor i_1 had to be dualized to assign derivative causality to the BG-F on the left in Fig. 66. Since the two BG-Fs shown in Fig. 66 are structurally observable, we do not require any further merging in our particular example.

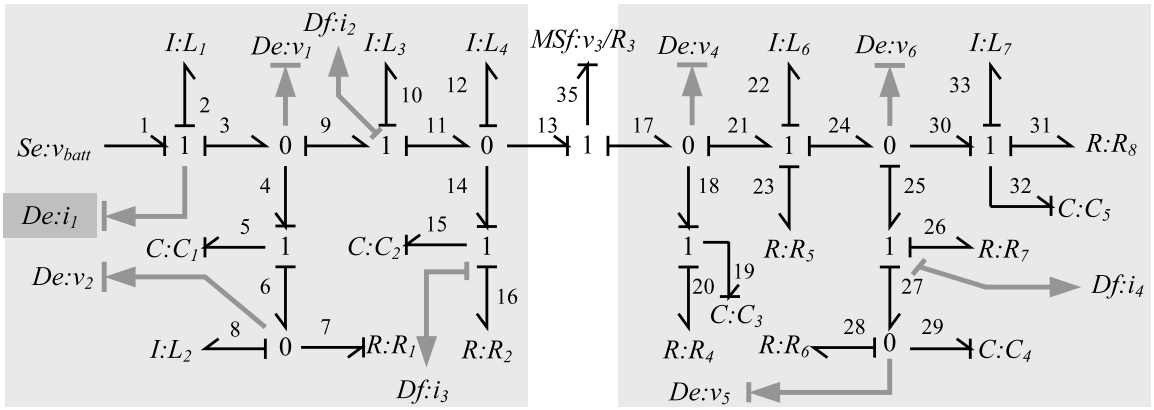


Figure 66: Two-Factored bond graph of the twelfth-order electrical circuit with imposed derivative causality.

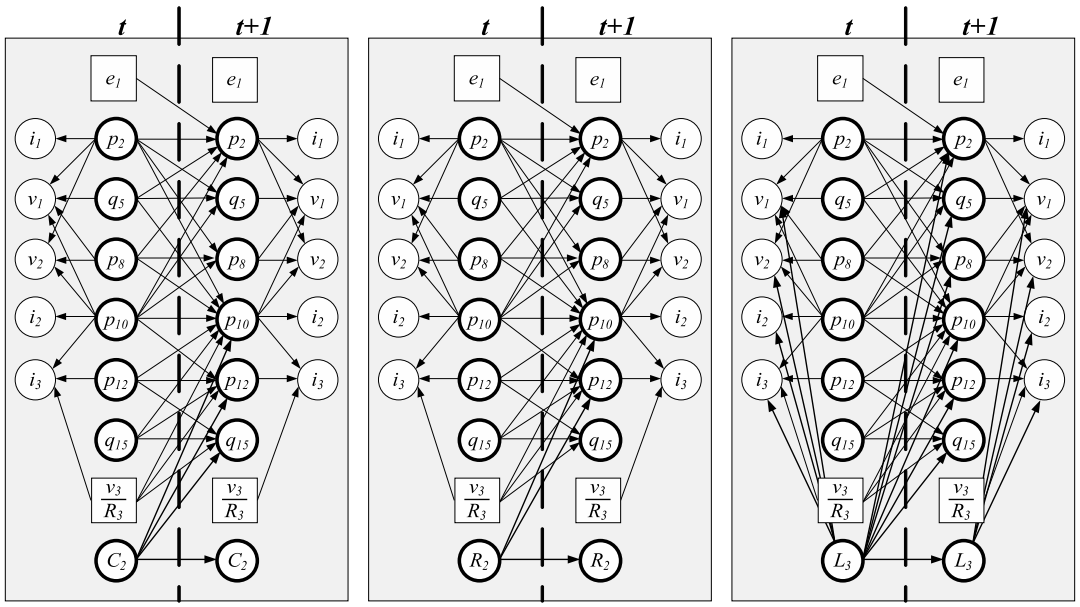
Table 20: Fault Signatures for Diagnoser D_2

Fault	i_4	v_4	v_5	v_6
C_3^{-a}, R_4^{+a}	0+	+−	0+	0+
C_3^{-i}, R_4^{+i}	0+	0+	0+	0+
C_4^{-a}	0−	0+	+−	+−
C_4^{-i}, R_6^{+a}, R_6^{+i}	0−	0+	0+	0+
L_7^{-a}	−+	0−	0−	−∗
L_7^{-i}	0−	0−	0−	0−
R_7^{+a}	0−	0+	0−	+−
R_7^{+i}	0−	0+	0−	0+

Distributed Diagnosis Experimental Results

In this section, we present experimental results obtained by applying the proposed distributed diagnosis approach to the electrical circuit shown in Fig. 49(a). Two distributed diagnosers, D_1 and D_2 are designed for this electrical circuit, for the top and bottom DBN-Fs shown in Fig. 65, respectively. The two diagnosers communicate measurement v_3 between each other. Tables 19 and 20 show the possible faults that must be diagnosed by each of the two diagnosers, and the fault signatures for each fault, given the measurements available to each diagnoser.

For our distributed diagnosis experiments, we applied the distributed Bayesian diagnosis scheme for the same fault scenarios described in Chapter V. The first experiment we performed was to diagnose an abrupt fault in C_2, C_2^{-a}, with $\Delta^a_{C_2} = -0.9$, that was introduced at time step, $t = 100$ s. As shown in Fig. 68, a negative deviation is noticed in measurement i_3 at $t = 101.4$ s. Based on this deviation, the fault hypotheses set, $\{C_2^{-i}, C_2^{-a}, R_2^{+a}, R_2^{+i}, L_2^{-i}, L_2^{-a}, L_4^{-i}, L_4^{-a}\}$, is generated. At $t = 102.9$ s, a negative deviation is observed in measurement i_2. Based on the fault signatures shown in Table 19, Qual-FI refines the fault hypotheses set by dropping fault hypotheses L_4^{-a} and



(a) DBN-F Fault model for C_2^{-a}/C_2^{-i}.

(b) DBN-F Fault model for R_2^{+a}/R_2^{+i}.

(c) DBN-F Fault model for L_3^{-a}/L_3^{-i}.

Figure 67: DBN-F Fault models for distributed diagnosis experiments using the Quant-FHRI approach 1.

L_4^{-i}. Finally, at $t = 102.9$ s, the deviation in i_3 is confirmed to be a gradual decrease. Hence, fault hypothesis L_4^{-a} is dropped as a fault hypothesis. At $t = 106.7$ s, measurement v_2 shows a positive deviation. As a result, based on the fault signatures shown in Table 19, the fault hypotheses are refined to $\{C_2^{-i}, C_2^{-a}, R_2^{+a}, R_2^{+i}\}$. Since these fault hypotheses cannot be further refined through Qual-FI alone, Quant-FHRI[1] is initiated. As shown in Fig. 69, the second diagnoser does not detect any fault. We start tracking the observed measurements from time $t = 97.5$ s, and instantiate two particle filters, one using a DBN-F model for fault C_2^{-i}/C_2^{-a}, and the other using a DBN-F model for fault R_2^{+i}/R_2^{+a}, with parameters C_2 and R_2 introduced as additional state variables in the nominal system DBN-Fs shown in Fig. 65. The DBN-F fault models for C_2^{-i}/C_2^{-a} and R_2^{+i}/R_2^{+a} are shown in Figs. 67(a) and 67(b), respectively. Fig. 70 shows the error in estimating the different measurements using the C_2 DBN-F fault model. Just like in the centralized diagnosis approach, for abrupt faults, at the time the fault is introduced, the abrupt fault parameter value is unknown. Hence, it takes some time for the correct fault model estimates to converge to the observed measurements, as we can see for the plots of measurements, v_1, v_2, i_2, and i_3 in Fig. 70. Estimates made using the incorrect R_2 DBN-F fault model, on the other hand, do not converge to the observed faulty behaviors, as the estimation errors in Fig. 71 show. A statistical test is employed on the sum of estimation

[1]For reasons explained in Chapter V, we adopt Quant-FHRI approach 1 for our distributed diagnosis experiments as well.

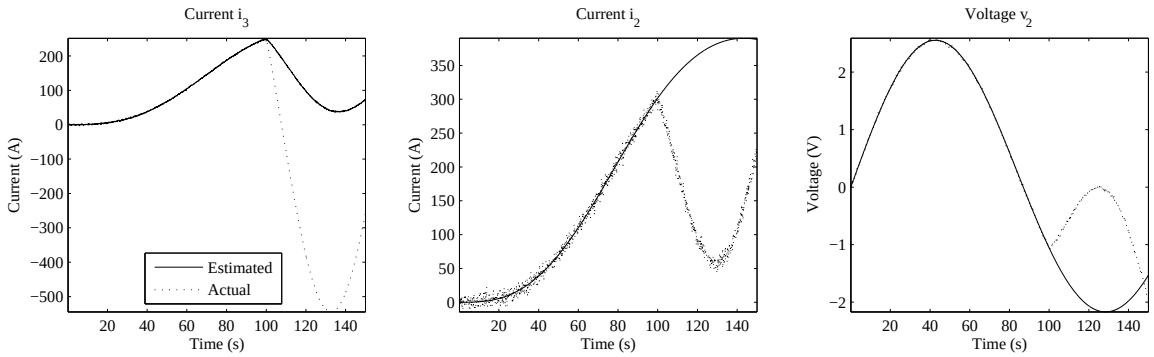


Figure 68: Detection of C_2^{-a} fault by diagnoser D_1.

errors across all the measurements to detect statistically significant sum of mean squared estimation errors. Fig. 72 shows the sum of mean squared estimation errors obtained using the two different fault models. The statistical test detects a statistically significant sum of mean squared estimation error obtained by both the DBN-F fault models at times $t = 101.3$ s and $t = 101.9$ s, respectively. However, the sum of mean squared estimation errors from the R_2 DBN fault model do not converge even after $s_d = 150$ s, whereas, the sum of mean squared estimation errors from the C_2 DBN fault model converges to the observed measurements from $t = 105.0$ s. Hence the true fault is isolated to be $C_2^{\pm a}/C_2^{\pm i}$ fault at $t = 251.3$ s. In order to determine whether the fault is an abrupt or incipient fault in C_2, we run a window-based Z-test on the difference between the known nominal parameter value and the estimated state variable. This approach is similar to the one used by TRANSCEND for symbol generation. At $t = 171.7$ s, the statistical test shows that the estimated parameter evolves in a -0 manner, implying it is an abrupt fault, and that it converges. By taking a mean of the values for 20 time steps after the abrupt fault is isolated, we obtain $\Delta^a_{C_2} = -0.897$. The actual value of $\Delta^a_{C_2}$ is -0.900. Thus, there is a 0.33% error in estimating $\Delta^a_{C_2}$. The estimate for the faulty parameter is shown in Fig. 73. Notice how the initial set of fault hypotheses generated in the distributed scheme is smaller than that generated in the centralized approach for the same fault experiment.

In the second scenario, we diagnose the incipient fault in L_3, denoted by L_3^{-i}, injected at time $t = 20$ s, with $\Delta^i_{L_3} = -0.05$. At time $t = 70.0$ s, a positive deviation from nominal is observed in measurement i_3 (see Fig. 74). This results in the generation of fault hypotheses set $\{L_3^{-a}, L_3^{-i}\}$. At $t = 84.2$ s, a positive deviation is observed in measurement i_2. Both the fault hypotheses are consistent with this symbol, and hence not dropped by the Qual-FI scheme. At $t = 106.1$ s, the deviation in i_3 is confirmed to be gradual, and fault hypothesis L_3^{-a} is dropped from consideration as its fault signature for i_3 is discontinuous $+-$. As shown in Fig. 75, the second distributed diagnoser

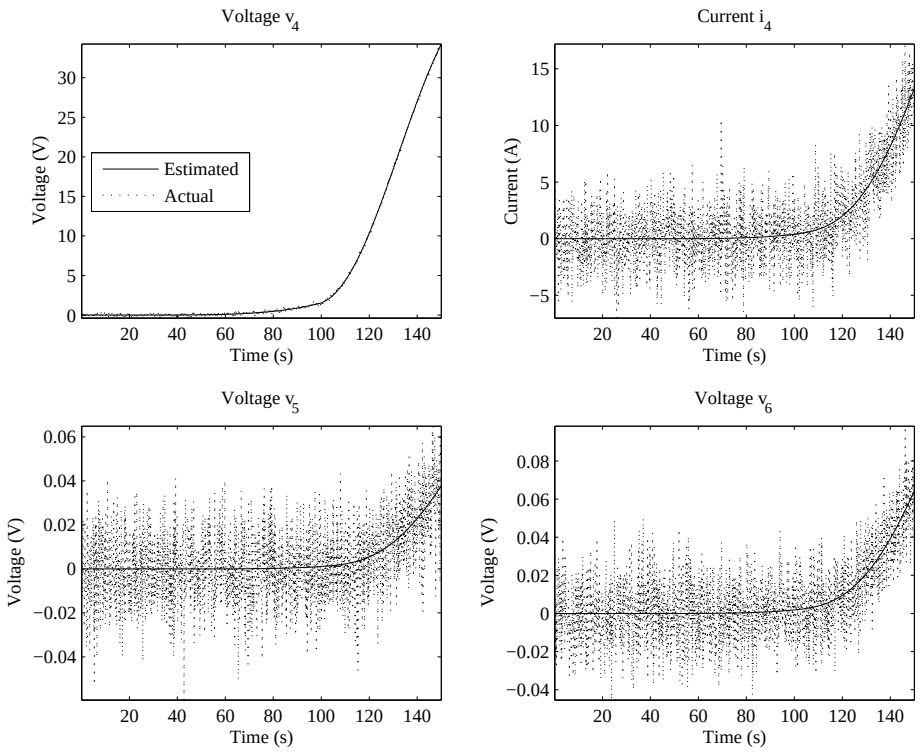


Figure 69: Tracking observations in the presence of C_2^{-a} fault by diagnoser D_2.

does not detect a fault. Since the Qual-FI isolates the true single fault, L_3^{-i}, the Quant-FHRI scheme is invoked to perform fault identification only. The DBN-F fault model for L_3^{-i}/L_3^{-a} is shown in Fig. 67(c). Quant-FHRI estimates $\Delta^i_{L_3} = -0.56$. Notice how the initial set of fault hypotheses generated in the distributed scheme is much smaller than that generated in the centralized approach. In the centralized scheme, C_3^{-a}, C_3^{-i}, C_4^{-a}, C_4^{-i}, R_4^{+a}, R_4^{+i}, R_7^{+a}, R_6^{+i}, R_7^{+a} and R_7^{+i} hypotheses were also generated when a positive deviation in i_3 was observed. However, in the distributed case, these hypotheses were not generated, because the task of diagnosis of these fault hypotheses is delegated to diagnoser D_2. If the true fault was indeed one of these fault hypotheses, the second distributed diagnoser, D_2 would have detected a fault. Since the fault hypotheses set is refined to a single fault by Qual-FI, we invoke the Quant-FHRI to perform fault identification only. We start tracking the observed measurements using a L_3 DBN-F fault model from $t = 17.5$ s. After $s_d = 150$ s from the time the faulty observations are tracked, a Z-test based scheme is initiated, and this statistical test confirms that the fault is indeed an incipient L_3^{-i} fault at $t = 134.8$ s. The estimates of L_3 obtained from our Quant-FHRI approach is presented in Fig. 76. The Quant-FHRI estimates $\Delta^i_{L_3} = -0.054$, while the true value of is $\Delta^i_{L_3} = -0.050$. Thus, there is a 8.00% error in estimating $\Delta^i_{L_3}$.

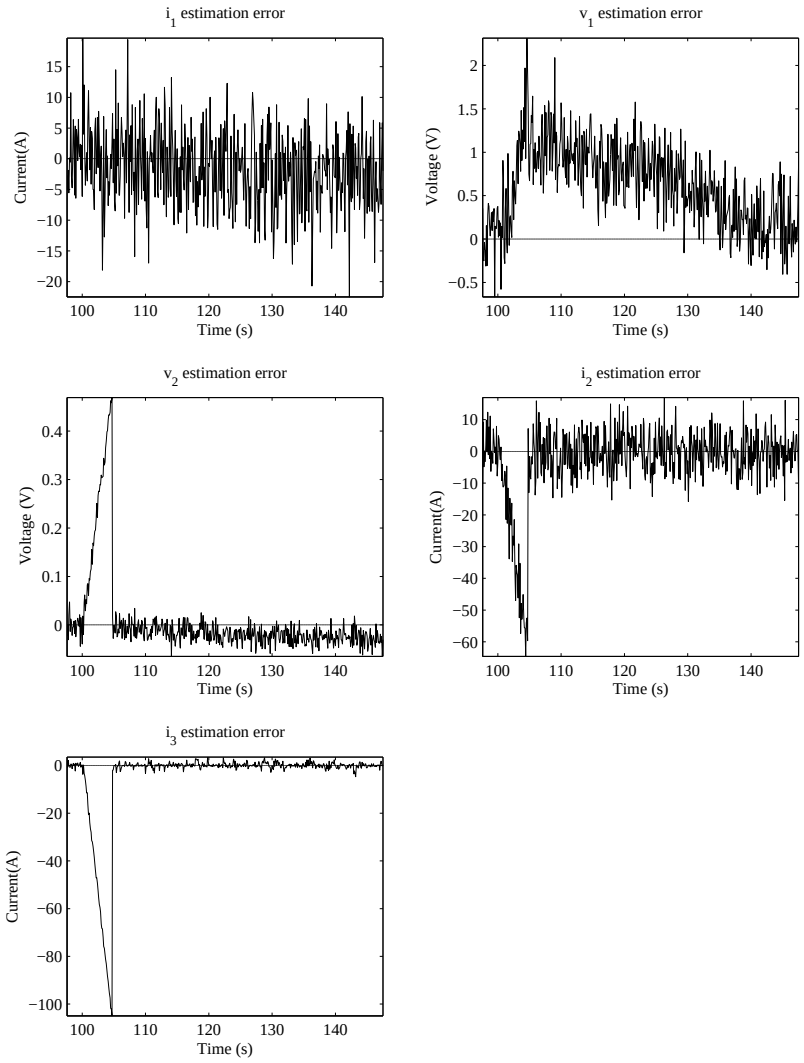


Figure 70: Tracking faulty system behavior using the $C_2^{\pm i}/C_2^{\pm a}$ DBN-F fault model.

Table 21 summarizes the results of the different distributed diagnosis experiments we ran on the electrical circuit example. For each experiment, we conducted 5 runs, and took the average of the time to fault detection, time to single fault isolation, time for the estimated parameter value to convergence to the true value, and the percentage error in the estimates of the true fault parameter. The estimates of the faulty parameters made using the correct fault models for each respective fault is shown in Fig. 77. By comparing the diagnosis results obtained by the centralized and distributed Bayesian diagnosis approaches, presented in Tables 16 and 21, respectively, we can draw the following conclusions.

First, the computational expense of our distributed diagnosis scheme was less than the centralized diagnosis scheme, since, the distributed diagnosis approach makes use of smaller DBN-F fault models,

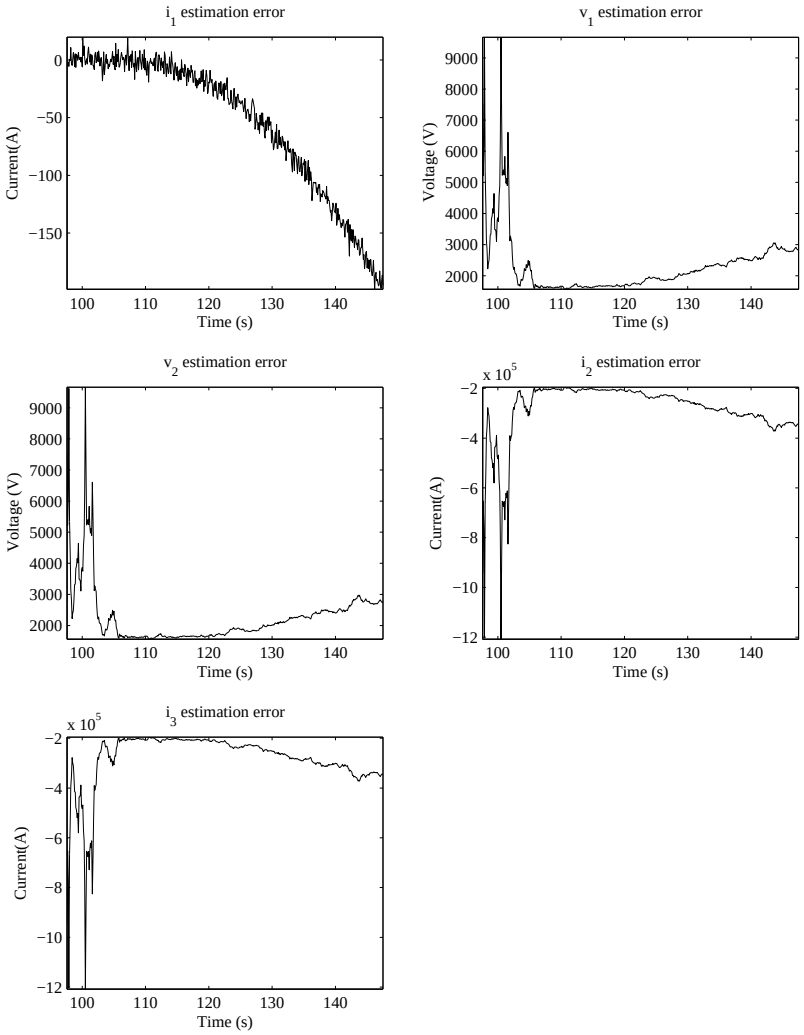
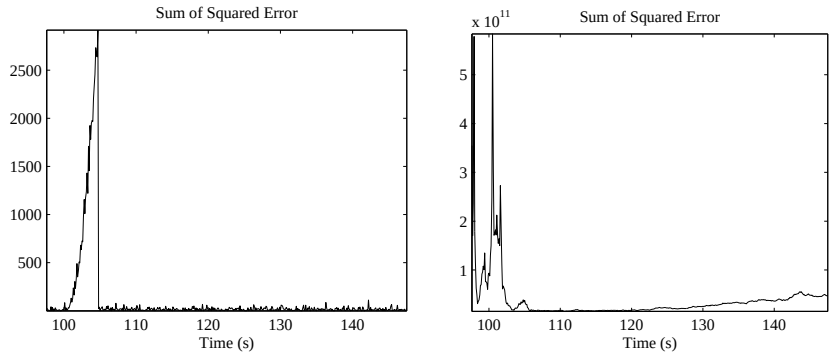


Figure 71: Tracking faulty system behavior using the $R_2^{\pm i}/R_2^{\pm a}$ DBN-F fault model.

compared to the DBN fault models used by the centralized scheme. If each distributed diagnoser is implemented on a separate process, the worst case efficiency of our distributed diagnosis scheme is determined by the largest DBN-F fault model used for tracking faulty measurements. Since, by construction, the largest DBN-F will still be smaller than the global DBN-F fault model, our distributed diagnosis is computationally more efficient than the centralized diagnosis scheme, given the same number of particles as the centralized scheme.

Compared to the centralized diagnosis approach, the distributed diagnosis approach resulted in larger estimation errors for four out of the six experimental runs, and comparable to the centralized diagnosis approach for the remainder of the two. Also, the parameter estimates made by the distributed approach took longer to converge in terms of the number of measurement points required than the centralized scheme. We anticipate the reason for this being the proportional distribution



(a) Sum of mean squared estimation errors obtained by C_2 DBN fault models.

(b) Sum of mean squared estimation errors obtained by R_2 DBN fault models.

Figure 72: Sum of mean squared estimation errors obtained by C_2 and R_2 DBN-F fault models.

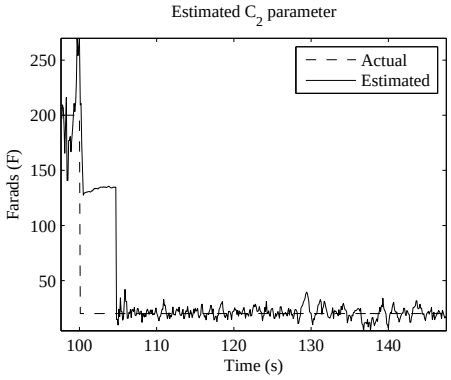


Figure 73: Estimate of C_2 obtained using the C_2 DBN-F fault model.

of particles based on the size of each factor, keeping the sum total of particles used by all the particle filters the same. Moreover, the use of a noisy sensor to compute the value of a state variable also contributed to this degraded accuracy. The centralized diagnosis scheme has access to more sensors and the state-estimates are not as noisy as those computed in terms of measurements in the distributed scheme. If the individual distributed diagnosers are executed on different processors, then we can increase the number of particles for each diagnoser, and our intuition is that this will improve the estimation accuracy and identification time of the distributed diagnosers. Thus, our experimental results on the twelfth-order electrical circuit illustrates the accuracy versus efficiency trade-off due to the factoring of the DBN into DBN-Fs.

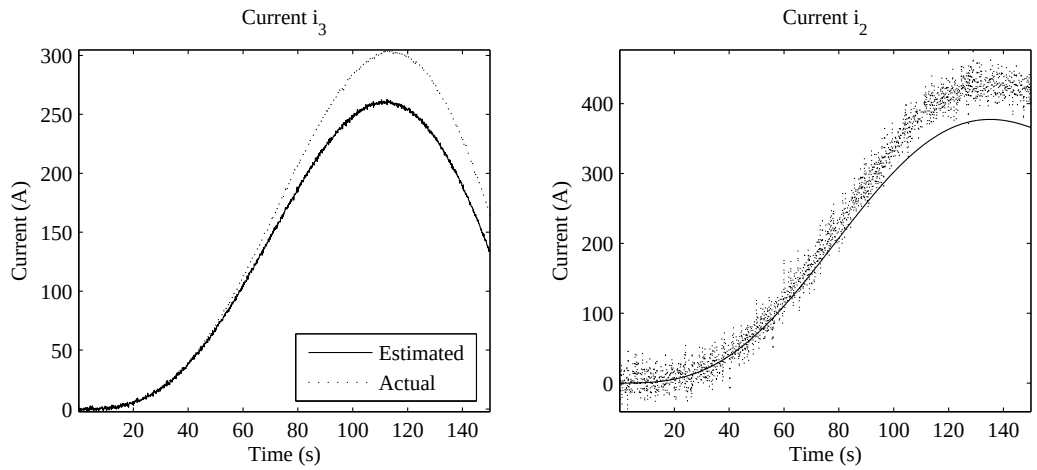


Figure 74: Detection of L_3^{-i} fault by diagnoser D_1.

Table 21: Results of Distributed Diagnosis Experiments on the Twelfth-Order Electrical Circuit with Particles Used Proportional to The Total Number of States per Factor

Experiment	Fault	Magnitude	Detection Time (s)	Isolation Time (s)	Convergence Time (s)	% Mean Absolute Error
1	C_2^{-a}	-0.90	1.04	55.06	5.88	0.64
2	C_2^{-i}	-0.55	37.3	53.90	19.80	39.74
3	L_3^{-a}	-0.90	0.50	4.32	6.56	1.11
4	L_3^{-i}	-0.05	50.40	82.50	23.76	21.14
5	C_3^{-a}	-0.90	0.20	3.02	3.64	0.13
6	R_7^{+a}	$+5.00$	118.30	163.30	128.64	0.66

Table 22: Results of Distributed Diagnosis Experiments on the Twelfth-Order Electrical Circuit with 500 Particles Used per Factor

Experiment	Fault	Magnitude	Detection Time (s)	Isolation Time (s)	Convergence Time (s)	% Mean Absolute Error
1	C_2^{-a}	-0.90	1.04	55.06	3.52	0.57
2	C_2^{-i}	-0.55	37.3	53.90	15.26	13.99
3	L_3^{-a}	-0.90	0.50	4.32	5.76	0.67
4	L_3^{-i}	-0.05	50.40	82.50	22.86	23.09
5	C_3^{-a}	-0.90	0.20	3.02	3.32	0.06
6	R_7^{+a}	$+5.00$	118.30	163.30	123.20	0.70

Table 23: Results of Distributed Diagnosis Experiments on the Twelfth-Order Electrical Circuit with 750 Particles Used per Factor

Experiment	Fault	Magnitude	Detection Time (s)	Isolation Time (s)	Convergence Time (s)	% Mean Absolute Error
1	C_2^{-a}	-0.90	1.04	55.06	3.50	0.68
2	C_2^{-i}	-0.55	37.3	53.90	13.30	14.63
3	L_3^{-a}	-0.90	0.50	4.32	5.06	0.45
4	L_3^{-i}	-0.05	50.40	82.50	19.84	16.55
5	C_3^{-a}	-0.90	0.20	3.02	3.36	0.08
6	R_7^{+a}	$+5.00$	118.30	163.30	117.2	0.52

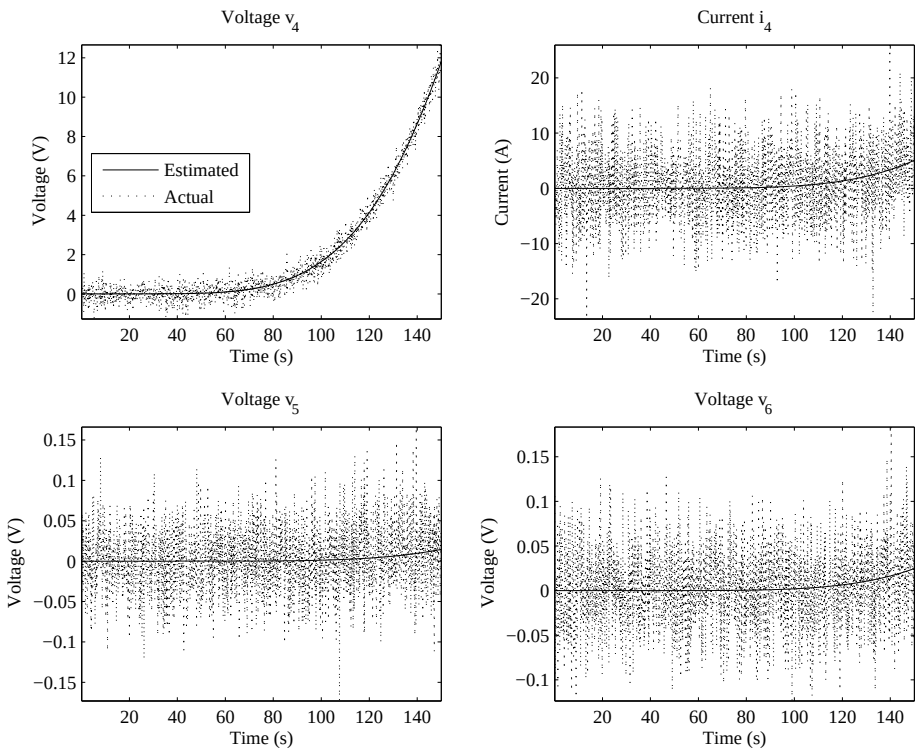


Figure 75: Tracking observations in the presence of L_3^{-i} fault by diagnoser D_2.

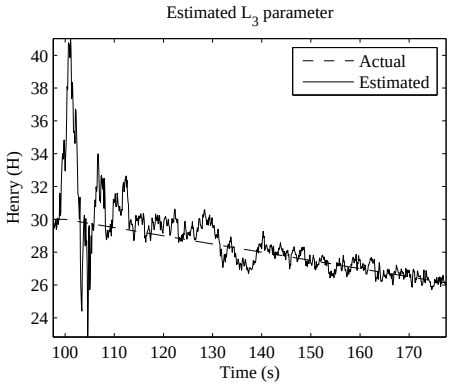


Figure 76: Estimate of L_3 obtained using the L_3 DBN-F fault model.

Table 24: Results of Distributed Diagnosis Experiments on the Twelfth-Order Electrical Circuit with 1000 Particles Used per Factor

Experiment	Fault	Magnitude	Detection Time (s)	Isolation Time (s)	Convergence Time (s)	% Mean Absolute Error
1	C_2^{-a}	-0.90	1.04	55.06	3.28	0.61
2	C_2^{-i}	-0.55	37.3	53.90	10.11	25.36
3	L_3^{-a}	-0.90	0.50	4.32	4.92	0.45
4	L_3^{-i}	-0.05	50.40	82.50	17.84	16.59
5	C_3^{-a}	-0.90	0.20	3.02	3.22	0.07
6	R_7^{+a}	$+5.00$	118.30	163.30	114.8	0.31

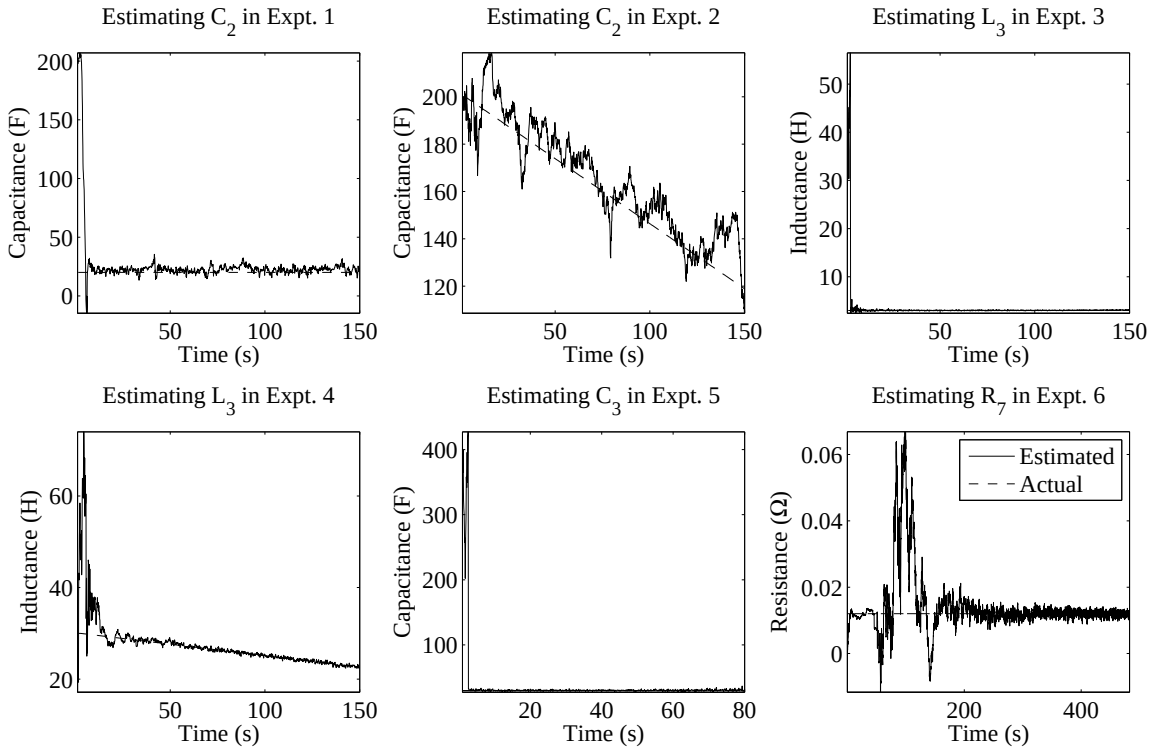


Figure 77: Parameter identification results for electrical circuit example.

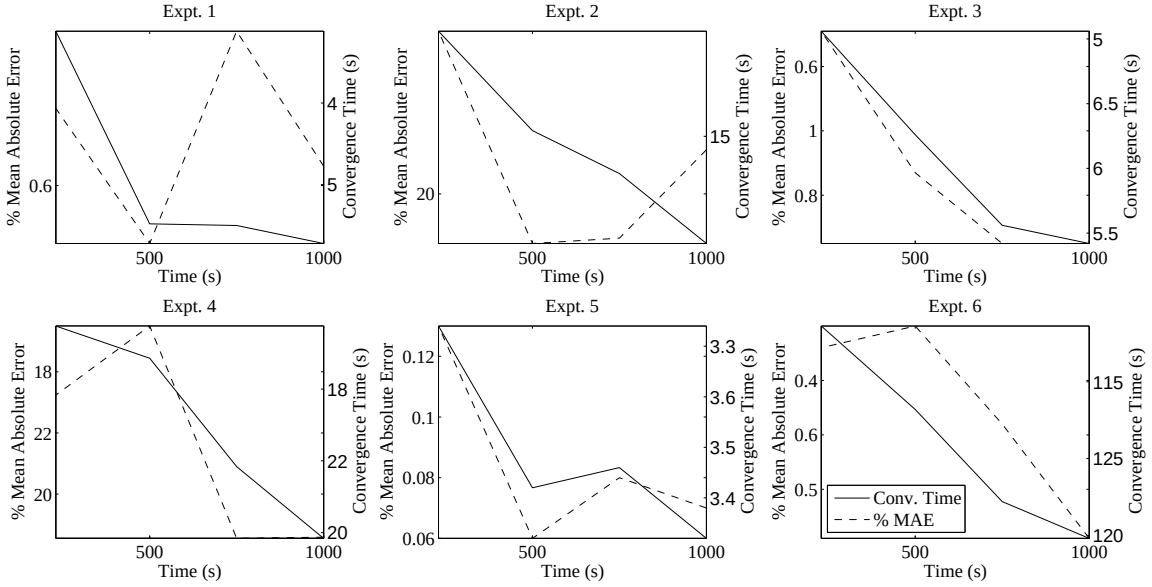


Figure 78: % MAE and convergence time of distributed Bayesian experiments for different numbers of particles.

Fig. 78 plots the % mean absolute error and convergence times of the six distributed experiments, but run using 500, 750, and 1000 particles. Tables 22, 23, and 24 show the results of the different distributed diagnosis experiments averaged over multiple runs using 500, 750, and 1000 particles, respectively. As was our intuition, both the % mean absolute error and the convergence time to the true parameter value show a decreasing trend with increase in the number of particles. However, compared to the centralized diagnosis experiments with 500 particles for the global DBN, the distributed diagnosis experiments with 1000 particles has less % mean absolute error for three out of the six fault cases, and faster convergence time for all six experiments. Experiments with 500 and 750 particles result in slower convergence times and more % mean absolute error than the experiments run using 1000 particles. Thus, increasing the number of particles used per factor for distributed diagnosis can improve estimation accuracy and convergence time.

Discussion and Summary

In this chapter, we established how the distributed diagnosers truly generate globally correct results without any centralized coordinator, and through communicating the minimal number of measurements alone, and not individual diagnoses. The requirement for communicating partial diagnosis results is avoided by systematic partitioning of the system in terms of structural observability. The requirement for communicating partial diagnoses is avoided because unlike other approaches, we have the knowledge of the global system model that is analyzed carefully for designing the diagnosers.

Several related approaches for distributed estimation of states exist in literature. Distributed decentralized extended Kalman Filters (DDEKF) [69] represent an approach for subdividing the estimation problem into smaller subproblems. However, in DDEKFs, each local component requires both measurements and estimates of state variables from other components to correctly estimate its states. As a result, the inaccuracies in one component can affect the estimation in other components. In our estimation approach, the variables in a factor are conditionally independent of those in all other factors, given some measurements. Hence, failures in individual factors do not affect the estimates made using other factors as long as the required measurements are available.

The BK algorithm, presented in [16], creates the individual factors by eliminating causal links between weakly interacting subsystems. Therefore, the belief state derived from the individual factors is an approximation of the true belief state. The error in this approximation is bounded, but these bounds may not be sufficiently precise for online diagnosis, since they may result in missed alarms and less precise diagnoses in the best case scenario, and false alarms and wrong diagnosis in

the worst case scenario. Heuristic techniques for automatically decomposing a DBN into factors are presented in [117]. This approach results in lower estimation errors, but the computed factored belief state is still an approximation. The Factored Particle Filtering (FPF) scheme [118] further reduces estimation errors by applying the particle filtering scheme to the BK factored inference approach. Our estimation approach uses the particle filtering scheme for inference using DBNs and preserves the overall system dynamics in the factored form, and *does not* approximate the belief state. Hence, we produce accurate state estimates efficiently.

In [119,120], Houfbaur and Williams present an approach of decomposing the overall model of a hybrid system into a set of concurrently operating automata that interact via shared measurements. These concurrently operating automata are termed concurrent probabilistic hybrid automata (cHPA) models, and like our DBN-Fs, are generated by converting observed variables as "virtual inputs". Once the system is decomposed, structural observability analysis is performed on each cPHA to exclude unobservable parts before implementing a Kalman filtering scheme on each cPHA for mode estimation. Although the cPHA generation approach is very similar in concept to our DBN factoring scheme, there are some notable differences between the two approaches. First, the procedure for generating the cPHAs requires the presence of a causal graph, where there are explicit causal links between the output variable of one component and a dependent variable of another. cPHAs are generated by slicing these causal graph at every vertex that has outgoing links to dependent variables, inserting a new vertex to represent a virtual input corresponding to the output node, and remapping the causal edge to originate from this new virtual input. The causal links are explicitly defined in the system descriptions. The strength of our factoring approach is that in our DBN models, the causal links between outputs and state variables are implicit and, must be determined based on the system's observation model. Hence, our scheme has more flexibility with respect to generating factors, compared to the cPHA generation scheme. Second, in the cPHA approach, dependent variables are not removed, and the order of the system, over all cPHAs, remains unchanged from the complete system. In our factoring scheme, the goal is to generate factors that are conditionally independent of other DBN-Fs over time, and this conditional independence is obtained only through the replacement of state variables by algebraic functions of output measurements, and removal of some across-time links. Application of cPHA-generation approach on DBNs might not yield conditionally independent DBN-Fs at all if all state variables have to be retained. Finally, both cPHA and DBN-F generation schemes ensure that each component satisfies the structural observability properties to allow correct state estimation. However, our definition of structural observability is stronger than the one applied in the cPHA generation scheme, since in the latter, the notion of

structural observability requires ensuring that every dependent variable is causally connected to an output variable. Recall that this is just the first of two conditions that need to be satisfied for our bond graph based procedure to determine a system to be structurally observable.

The DBN-F-based distributed diagnosis approach addresses the drawbacks of the centralized Bayesian diagnosis approach presented in V. The conditional independence of random variables in each DBN-F ensures that there is no requirement for a centralized coordination mechanism to generate globally correct diagnosis results. The reduced order state-space of the individual DBN-Fs also results in a reduction of computational complexity. Since we guarantee each DBN-F represents a structurally observable subsystem, we ensure that our distributed particle filter-based observers will generate accurate results. However, the design of DBN-Fs is based on Assumptions 2 and 3, i.e., the sensors associated with measurements converted to inputs will not be faulty, and the components whose parameters are used in the algebraic functions are assumed not to fail. Therefore, there is a trade-off for robustness to gain efficiency.

CHAPTER VII

CONCLUSIONS

In this dissertation, we presented two distributed diagnosis approaches: (*i*) a qualitative fault signature-based scheme that extends the TRANSCEND centralized diagnosis approach, and (*ii*) a Dynamic Bayesian Network (DBN)-based scheme for efficient and accurate diagnosis of abrupt and incipient faults in uncertain environments. The qualitative and quantitative approaches are combined into a unified framework for fault detection, isolation, and identification. The basis of our distributed diagnosis schemes is the careful design of the distributed diagnosers through the offline analysis of system properties, e.g., diagnosability and structural observability. By careful construction of these diagnosers, we guarantee that each distributed diagnoser generates globally correct diagnosis results by local analysis without a centralized coordinator, with minimal exchange of measurement information with each other, and without exchanging any partial diagnosis results amongst themselves.

The first step towards developing the TRANSCEND-based distributed diagnosis scheme was extending TRANSCEND to generate incipient fault hypothesis, and modifying TRANSCEND's existing fault generation scheme to derive the fault signatures for incipient fault hypotheses. We then analyzed the diagnosability properties of the global system to come up with an approach for designing our distributed qualitative fault diagnosers. We proposed two approaches for designing the distributed qualitative diagnosers. In the first diagnoser design approach, we assumed knowledge of subsystem structure, especially the measurements and faults that belong to each subsystem, and based on this information, we designed a local diagnoser for each subsystem such that it required minimal number of additional external measurements to globally diagnose all the faults assigned to that subsystem. In the second approach, we assumed no prior partitioning information. Instead, we generated the maximal number of distributed diagnosers, such that, each local diagnoser could operate independently without sharing measurements to generate globally correct diagnosis results.

In the Bayesian distributed diagnosis scheme, the DBN framework is designed to handle uncertainties in the form of process and measurement noise. To accommodate arbitrary distributions and non-linearities in the system, a particle filtering scheme is employed. We proposed three different approaches to address the sample impoverishment problem inherent in particle filtering-based fault diagnosis schemes. For a Bayesian state estimation scheme to be able to generate correct state

estimates, the system must be observable. Classical notions of analyzing the observability property requires the knowledge of system parameters and are mostly defined for linear systems. Since, our systems can be nonlinear, we explored the structural observability property of systems and ensured that the system is structurally observable before applying our Bayesian diagnosis approach to the system. Bayesian diagnosis schemes can be computationally expensive for large systems, so we proposed a novel scheme for combining the qualitative TRANSCEND fault isolation scheme with the Bayesian diagnosis approach to improve the overall computational efficiency and accuracy of the diagnosis of incipient and abrupt faults in the presence of uncertainties.

The efficiency and scalability of our Bayesian diagnosis scheme was further improved by distributing the diagnosis task amongst distributed diagnosers that generate globally correct diagnosis results locally, without a centralized coordinator, with minimal exchange of measurement information with each other, and without exchanging any partial diagnosis results amongst themselves. The basis of our diagnoser design is a procedure for factoring the global system DBN into DBN factors (DBN-Fs), such that the random variables in each DBN-F are conditionally independent of the random variables in all other DBN-Fs, given some subset of communicated measurements. The conditional independence amongst random variables of the DBN-Fs makes it possible to invoke the centralized, combined quantitative-qualitative Bayesian diagnosis on each individual DBN-F independently. Thus, in each distributed diagnoser, the fault detector makes use of a particle filtering scheme applied to a DBN-F to estimate the values of the state variables in that DBN-F. We design our distributed Bayesian diagnosers by analyzing the structural observability of the system and its component parts. This is in contrast to the design procedure for TRANSCEND-based distributed diagnosers, where we analyzed the diagnosability properties of a system to design the distributed diagnosers. The application of structural observability for the design and analysis for Bayesian diagnosers is, to the best of our knowledge, novel.

The experimental results of applying our distributed qualitative diagnosis to the Advanced Water Recovery System, demonstrated that our distributed qualitative diagnosis scheme is computationally more efficient than the centralized TRANSCEND diagnosis approach. Our centralized and distributed Bayesian diagnosis schemes were applied to a complex, twelfth-order electrical circuit, with highly oscillatory behavior. The centralized Bayesian diagnosis schemes resulted in correct and more precise diagnosis results in the presence of noisy sensors, when compared to the purely qualitative TRANSCEND diagnosis scheme. The distributed Bayesian diagnosis scheme resulted in improvement in computational efficiency over the centralized Bayesian diagnosis approach while still generating precise diagnosis results.

Note that our approach of analyzing the diagnosability and observability properties of a system offline to design distributed diagnosers that generate globally correct online diagnosis results through local analysis, without a centralized coordinator, with minimal exchange of measurement information, and no exchange of partial diagnosis results is not restricted to any one particular diagnosis scheme. In fact, our distributed diagnosis philosophy can be applied to other diagnosis approaches, given that we analyze diagnosability and/or observability properties pertinent to the particular diagnosis scheme. For example, we can apply our distributed diagnoser design scheme to any general ARR approach, if we are given the global incidence matrix which has information about which residuals are affected by which faults. One must note that the qualitative TRANSCEND diagnosis approach is itself a form of ARR, but TRANSCEND uses more information about its residuals (such as magnitude and slope symbol) rather than a binary 0 or 1 value to determine if the fault merely affects a residual or not, as is the case in other general ARR-based schemes [121].

Future Directions

Our distributed diagnosis approaches do have some limitations and there are open problems that need to be addressed in future work. In [1], the author presents a discrete event systems (DES) approach for diagnosis of continuous systems, derived from the qualitative TRANSCEND diagnostic framework. This approach automatically constructs a labeled transition system that describes the fault model, and also generates a computationally efficient event-based diagnoser. As part of future work, we would like to investigate how the algorithms described in this paper can be extended to develop distributed DES approaches for diagnosing continuous systems.

The scope of this research was restricted to the diagnosis of single, abrupt and incipient parametric faults. Although these faults cover a large set of real-world fault scenarios, there are cases when this assumption needs to be relaxed. In the future, we would like to extend the diagnosis approaches presented in this dissertation to the diagnosis of multiple abrupt and incipient parametric faults. Recall that once factored, Bayesian estimation algorithms can be applied to the different DBN-Fs independently. Also, the fault in a DBN-F will only cause the fault detector for that DBN-F to detect the fault, and the other distributed diagnosers will not detect this fault. Hence, our distributed Bayesian diagnosis approach can be easily extended to multiple faults, where the multiple faults are defined as a collection of faults, where only a single fault can occur in each DBN-F.

We would also like to extend our distributed Bayesian diagnosis scheme to the diagnosis of sensor faults. The basis of our distributed Bayesian diagnosis scheme is the computation of some

state variables in terms of some measurement values, that are now considered as system inputs. In this dissertation, we have assumed that these sensors that are used for computing state variables may be noisy, but not faulty. Although not discussed in this dissertation, the diagnosis of sensor faults is measurements that are not considered as system inputs requires a straightforward application of the sensor fault diagnosis scheme presented in [1]), where a sensor fault is isolated if a deviation is observed only in that sensor and no other measurements. However, the application of this sensor fault diagnosis scheme to sensors that are now considered as system inputs is not straightforward. One solution for diagnosing sensor faults is to have, in addition to the standard set of maximal number of observable DBN-Fs, a set of different DBN-Fs, where the sensors are not considered to be system inputs, and apply the approach presented in [1] on these additional DBN-Fs only to diagnose sensor faults.

At present, the distributed Bayesian diagnosers are designed by analyzing the structural observability of the system. Since the diagnosability property of the system is not considered during factoring, we are often left with large number of faults that cannot be uniquely isolated by applying qualitative TRANSCEND isolation scheme alone, and we rely on the fault hypothesis refinement and identification step to refine the fault hypotheses further, ideally to the single true fault. In the future, we would like to improve the efficiency of our distributed Bayesian diagnosis approach further by ensuring that the DBN-Fs are carefully designed to reduce the number of faults that cannot be isolated further by the qualitative fault isolation step to a minimal number.

Finally, as part of future work, we seek to investigate the important research problem of studying the diagnosability of the faulty DBN models, once the extra fault variables are introduced. It has been shown in [122] that the introduction of a fault in a system does not affect the structural observability of the system, since injection of fault in the bond graph parameters does not increase the number of extra states. However, for the sake of estimation, we include the faulty parameter as a state variable, and this extra state variable may affect the estimation problem, and the diagnosability of our Quant-FHRI scheme is related to the ability of estimating the faulty parameter correctly. The problem of identifying the correct set of measurements such that the faulty parameter can be estimated (and hence diagnosed) correctly, therefore, is also an important research task. We also wish to apply our diagnosis approach to a large real-world system to better analyze the scalability and efficiency of our methodology.

REFERENCES

[1] M. Daigle, "A qualitative event-based approach to fault diagnosis of hybrid systems," PhD Dissertation, Vanderbilt University, Nashville, TN, 2008.

[2] J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems.* New York, NY: Marcel Dekker, Inc., 1998.

[3] K. P. Murphy, "Dynamic Bayesian networks: Representation, inference, and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[4] V. Verma, G. Gordon, R. Simmons, and S. Thrun, "Real-time fault diagnosis," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 2, pp. 56–66, 2004.

[5] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[6] J. Sottile, F. C. Trutt, and J. L. Kohler, "Experimental investigation of on-line methods for incipient fault detection," *Conference Record of the 2000 IEEE Industry Applications Conference*, vol. 4, pp. 2682 – 2687, 2000.

[7] P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 29, no. 6, pp. 554–565, 1999.

[8] E.-J. Manders, S. Narasimhan, G. Biswas, and P. J. Mosterman, "A combined qualitative/quantitative approach for fault isolation in continuous dynamic systems," in *Proceedings of the* 4^{th} *Symposium on Fault Detection, Supervision, and Safety for Technical Processes*, Budapest, Hungary, 2000, pp. 1074–1079.

[9] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dynamic System: Theory and Applications*, vol. 10, no. 1/2, pp. 33–86, January 2000.

[10] Y. Pencole and M.-O. Cordier, "A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks," *Artificial Intelligence*, vol. 164, no. 1-2, pp. 121–170, 2005.

[11] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Designing distributed diagnosers for complex continuous systems," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 2, pp. 277–290, April 2009.

[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[13] R. Dearden and D. Clancy, "Particle filters for real-time fault detection in planetary rovers," in *Proceedings of the* 12^{th} *International Workshop on Principles of Diagnosis*, 2001, pp. 1–6.

[14] C. Sueur and G. Dauphin-Tanguy, "Bond graph approach for structural analysis of MIMO linear systems," *Journal of the Franklin Institute*, vol. 328, no. 1, pp. 55–70, 1991.

[15] ——, "Structural controllability/observability of linear systems represented by bond graphs," *Journal of The Franklin Institute*, vol. 326, no. 6, pp. 869–883, 1989.

[16] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," in *Proceedings of the* 14^{th} *Annual Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 33–42.

[17] K. D. Pickering, K. Wines, G. M. Pariani, L. A. Franks, J. Yeh, B. W. Finger, M. L. Campbell, C. E. Verostko, C. Carrier, J. C. Gandhi, and L. M. Vega, "Early results of an integrated water recovery system test," in *Proceedings of the* 29^{th} *International Conference on Environmental Systems*, Orlando, FL, 2001.

[18] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*, 3rd ed. New York, NY, USA: John Wile & Sons, Inc., 2000.

[19] R. Isermann, "Process fault detection based on modeling and estimation methods: A survey," *Automatica*, vol. 20, pp. 387–404, 1984.

[20] ——, "Supervision, fault-detection and fault-diagnosis methods – an introduction," *Control Engineering Practice*, vol. 5, no. 5, pp. 639 – 652, 1997.

[21] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, pp. 97–130, 1987.

[22] R. Davis and W. Hamscher, *Model-based reasoning: troubleshooting.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[23] R. Isermann and P. Balle, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709–719, 1997.

[24] M. Stumptner and F. Wotowa, "Guest-editorial speial issue on industrial applications of model-based reasoning," *AI Communications*, vol. 13, no. 2, 2000.

[25] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications." *Control Engineering Practice*, vol. 5, no. 5, pp. 671–682, 1997.

[26] 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.

[27] S. Abdelwahed, J. Wu, G. Biswas, J. Ramirez, and E.-J. Manders, "Online adaptive control for effective resource management in advanced life support systems," *Habitation*, vol. 10, no. 2, pp. 105–115, 2005.

[28] G. Vachtsevanos, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems.* Wiley, 2006.

[29] J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, vol. 8, no. 6, pp. 3 – 11, December 1988.

[30] B. Peischl and F. Wotawa, "Model-based diagnosis or reasoning from first principles," *IEEE Intelligent Systems*, vol. 18, no. 3, pp. 32 – 37, 2003.

[31] R. Reiter, "A theory of diagnosis from first principles," *Readings in Model-based Diagnosis*, pp. 29–48, 1992.

[32] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 3rd ed. Amsterdam: Kluwer-Nijhoff, 1996.

[33] E. Gelso, S. Castillo, and J. Armengol, "Fault diagnosis by refining the parameter uncertainty space of nonlinear dynamic systems," in *American Control Conference, 2008*, June 2008, pp. 4304–4308.

[34] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall Inc., 1995.

[35] J. Lunze, "Diagnosis of quantized systems based on a timed discrete-event model," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 3, pp. 322–336, 2000.

[36] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199 – 1212, 2003.

[37] T. Umeda, T. Kuriyama, E. O'Shima, and H. Matsuyama, "A graphical approach to cause and effect analysis of chemical processing systems," *Chemical Engineering Science*, vol. 35, no. 12.

[38] M. Iri, K. Aoki, E. O'Shima, and H. Matsuyama, "An algorithm for diagnosis of system failures in the chemical process," *Computers and Chemical Engineering*, vol. 3, no. 1-4.

[39] D. L. Dvorak and B. Kuipers, "Model-based monitoring of dynamic systems," in *International Joint Conference on Artificial Intelligence*, Detroit, MI, USA, 1989, pp. 1238–1243.

[40] ——, "Process monitoring and diagnosis: A model based approach," *IEEE Expert*, vol. 6, no. 2, pp. 67–74, 1991.

[41] M. A. Djeziri, R. Merzouki, B. O. Bouamama, and G. Dauphin-Tanguy, "Robust fault diagnosis by using bond graph approach," *IEEE Transactions on Mechatronics*, vol. 12, no. 6, pp. 599–611, 2007.

[42] W. M. Wonham, *Notes on Control of Discrete-Event System: ECE 1636F/1637S 2003-2004*, 2003.

[43] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems.* Boston, MA: Kluwer Academic Publishers, 1999.

[44] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Transactions Control Systems Technology*, vol. 4, no. 2, pp. 105–124, 1996.

[45] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis Part II: Qualitative models and search strategies," *Computers and Chemical Engineering*, vol. 27, pp. 313 – 326, 2003.

[46] J. Shiozaki, H. Matsuyama, E. O'Shima, and M. Iri, "An improved algorithm for diagnosis of system failures in the chemical process," *Computers and Chemical Engineering*, vol. 9, no. 3, pp. 285–293, 1985.

[47] C. C. Chang and C. C. Yu, "Online fault diagnosis using signed directed graph," *Industrial and Engineering Chemistry Research*, vol. 29, no. 7, pp. 1290–1299, 1990.

[48] M. A. Kramer and B. L. Palowitch, "A rule-based approach to fault diagnosis using the signed directed graph," *AIChE Journal*, vol. 33, no. 7.

[49] P. Rose and M. A. Kramer, "Qualitative analysis of causal feedback," in *Proceedings of the AAAI*, 1991, pp. 817–823.

[50] B. Kuipers, "Qualitative simulation," *Artificial Intelligence*, vol. 29, no. 3, pp. 289 – 338, 1986.

[51] R. J. Patton, J. Chen, and H. Y. Zhang, "Design of unknown input observers and robust fault detection filters," *International Journal of Control*, vol. 63, pp. 85–105, 1996.

[52] J. Moreno, "Unknown input observers for SISO nonlinear systems," in *Proceedings of the 39^{th} IEEE Conference on Decision and Control*, 2000, pp. 790–801.

[53] J. Gertler, "Analytical redundancy methods in fault detection and isolation," in *Proceedings of IFAC Symposium on Fault Detection Supervision Safety Technical Processes*, 1991, pp. 9–21.

[54] J. Gertler, X. Fang, and Q. Luo, "Detection and diagnosis of plant failures: The orthogonal parity equation approach," *Control and Dynamic Systems*, vol. 37, pp. 159–216, 1990.

[55] P. Young, "Parameter estimation for continuous time models - a survey," *Automatica*, vol. 17, no. 1, pp. 23–29, 1991.

[56] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: robustness and applications," *Control Engineering Practice*, vol. 5, no. 5, pp. 671 – 682, 1997.

[57] M. Staroswiecki and G. Comtet-Varga, "Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems," *Automatica*, vol. 37, pp. 687–699, 2001.

[58] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proceedings of the* 17^{th} *National Conference on Artificial Intelligence*, 2000, pp. 531–537.

[59] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Efficient tracking for diagnosis using factored dynamic Bayesian networks," in *Proceedings of the* 7^{th} *IFAC Symposium on Fault Detection, Supervision, and Safety of Technical Processes (SAFEPROCESS 2009)*, 2009, pp. 300–305.

[60] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[61] M. Fiszman, W. W. Chapman, D. Aronsky, R. S. Evans, and P. J. Haug, "Automatic detection of acute bacterial pneumonia from chest X-ray reports," *Journal of the American Medical Informatics Association*, vol. 7, no. 6, pp. 593–604, 2000.

[62] D. Nikovski, "Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 509–516, 2000.

[63] J. F. Huard and A. A. Lazar, "Fault isolation based on decision-theoretic troubleshooting," Center for Telecommunications Research, Columbia University, Tech. Rep. CU/CTR/TR 442-96-08, 1996.

[64] C. S. Hood and C. Ji, "Proactive network-fault detection," *IEEE Transactions on Reliability*, vol. 46, no. 3, pp. 333–341, 1997.

[65] R. E. Neapolitan, *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. New York, NY: John Wiley & Sons, Inc., 1990.

[66] J. Pearl, "Evidential reasoning using stochastic simulation of causal models," *Artificial Intelligence*, vol. 32, pp. 245–257, 1987.

[67] R. D. Shacter and M. A. Peot, "Simulation approaches to general probabilistic inferences on belief networks," in *Proceedings of the* 5^{th} *Annual Conference on Uncertainty in Artificial Intelligence*, 1990, pp. 221–234.

[68] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optical decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967.

[69] A. G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. Boca Raton: CRC Press, 1998.

[70] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Sensor fault detection and identification in a mobile robot," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, 1998, pp. 1383–1388.

[71] A. J. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparitive study," *Journal of Engineering for Gas Turbines and Power*, vol. 125, pp. 917–924, 2003.

[72] A. Volponi and L. A. Urban, "Mathematica methods of relative engine performance diagnosis," *SAE Transactions*, vol. 101, 1992, *Journal of Aerospace*, Technical Paper 922048.

[73] S. Simani, C. Fantuzzi, and S. Beghelli, "Diagnosis techniques for sensor faults of industrial processes," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 848–855, 2000.

[74] N. de Freitas, "Rao-blackwellised particle filtering for fault diagnosis," in *Proceedings of the IEEE Aerospace Conference*, 2002, pp. 1767–1772.

[75] F. Hutter and R. Dearden, "The Gaussian particle filter for diagnosis of non-linear systems," in *Proceedings of the 5^{th} IFAC Symposium on Fault Detection, Supervision, and Safety of Technical Processes*, 2003.

[76] X. Koutsoukos, J. Kurien, and F. Zhao, "Monitoring and diagnosis of hybrid systems using particle filtering methods," in *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[77] D. Koller and U. Lerner, "Sampling in factored dynamic systems," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001.

[78] C. Rogo, R. Prasanth, R. K. Mehra, and R. Fortenbaugh, "Failure detection and identification and fault tolerant control using the IMM-KF with applications to the Eagle-Eye UAV," in *Proceedings of the 37^{th} IEEE International Conference on Decision and Control*, vol. 4, 1998, pp. 4208–4213.

[79] P. Goel, G. Dedeoglu, S. I. Roumeliotis, and G. S. Sukhatme, "Fault detection and identification in a mobile robot using multiple model estimation and neural network," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, 2000, pp. 2302–2309.

[80] A. Blake and M. Isard, "The CONDENSATION algorithm — conditional density propagation and applications to visual tracking," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, p. 361.

[81] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999. [Online]. Available: citeseer.ist.psu.edu/fox99markov.html

[82] S. Narasimhan, G. Biswas, G. Karsai, T. Pasternak, and F. Zhao, "Building observers to handle fault isolation and control problems in hybrid systems," in *Proceedings of the IEEE International Conference on Systems Man and Cybernetics*, vol. 4, Nashville, TN USA, 2000, pp. 2393–2398.

[83] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 37, no. 3, pp. 348–361, May 2007.

[84] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "A bayesian approach to efficient diagnosis of incipient faults," in *Proceedings of the 17^{th} International Workshop on Principles of Diagnosis*, Spain, 2006, pp. 243–250.

[85] ——, "Comprehensive diagnosis of continuous systems using dynamic bayes nets," in *Proceedings of the 19^{th} International Workshop on Principles of Diagnosis*, 2008, pp. 151–158.

[86] E. J. Manders, G. Biswas, J. Ramirez, N. Mahadevan, J. Wu, and S. Abdelwahed, "A model-integrated computing tool-suite for fault adaptive control," in *Proceedings of the 15^{th} International Workshop on Principles of Diagnosis*, 2004.

[87] P. Mosterman and G. Biswas, "A theory of discontinuities in physical system models," *Journal of the Franklin Institute*, vol. 335B, no. 3, pp. 401–439, 1998.

[88] B. O. Bouamama, K. Medjaher, M. Bayart, A. Samantaray, and B. Conrard, "Fault detection and isolation of smart actuators using bond graphs and external models," *Control Engineering Practice*, no. 2, pp. 159–175, February 2005.

[89] I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P. J. Mosterman, "A method for efficient simulation of hybrid bond graphs," in *Proceedings of the International Conference of Bond Graph Modeling*, San Diego, California, 2007, pp. 177 – 184.

[90] P. J. Mosterman, E.-J. Manders, and G. Biswas, "Qualitative dynamic behavior of physical system models with algebraic loops," in *Proceedings of the* 11^{th} *International Workshop on Principles of Diagnosis*, Moriela, Mexico, May 2000, pp. 155–162.

[91] G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, and G. Karsai, "A robust method for hybrid diagnosis of complex systems," in *Proceedings of the* 5^{th} *Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Washington, DC, June 2003, pp. 1125–1131.

[92] M. J. Chantler, S. Daus, T. Vikatos, and G. M. Coghill, "The use of qualitative dynamic models and dependency recording for diagnosis," in *Proceedings of the* 9^{th} *International Workshop on the Principles of Diagnosis*, Cape Cod, MA USA, 1998, pp. 59 – 68.

[93] M. Daigle, X. Koutsoukos, and G. Biswas, "Relative measurement orderings in diagnosis of distributed physical systems," in *Proceedings of the* 43^{rd} *Annual Allerton Conference on Communication, Control, and Computing*, September 2005, pp. 1707–1716.

[94] S. Narasimhan, P. J. Mosterman, and G. Biswas, "A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems," in *Proceedings of the* 9^{th} *International Workshop on Principles of Diagnosis*, Cape Cod, MA USA, 1998, pp. 94 – 101.

[95] I. Roychoudhury, G. Biswas, X. Koutsoukos, and S. Abdelwahed, "Designing distributed diagnosers for complex systems," in *Proceedings of the* 16^{th} *International Workshop on Principles of Diagnosis*, Monterey, California, June 2005, pp. 31–36.

[96] T. K. Yu and J. H. Seinfeld, "Observability and optimal measurement locations in linear distributed parameter systems," *International Journal of Control*, vol. 18, pp. 785–799, 1973.

[97] J. E. T. Penny, M. L. Friswell, and S. D. Garvey, "Automatic choice of measurement locations for dynamic settings," *AIAA Journal*, vol. 32, no. 2, pp. 407–414, 1994.

[98] S. Y. Chen and Y. F. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 1, pp. 393–408, 2004.

[99] V. T. Paschos, "A survey of approximately optimal solutions to some covering and packing problems," *ACM Computing Surveys*, vol. 29, no. 2, pp. 171–209, 1997.

[100] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 576 – 593, Apr. 2002.

[101] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, pp. 293 – 311, 2003.

[102] J. Kurien, X. Koutsoukos, and F. Zhao, "Distributed diagnosis of networked embedded systems," in *Proceedings of the* 13^{th} *International Workshop on Principles of Diagnosis*, Semmering, Austria, 2002, pp. 179–188.

[103] R. Su, W. Wohnam, J. Kurien, and X. Koutsoukos, "Distributed diagnosis of qualitative systems," in 6^{th} *International Workshop on Discrete Event Systems, Zaragoza (WODES-2002)*, Zaragosa, Spain, October 2002, pp. 169–174.

[104] E. Fabre, A. Benveniste, S. Haar, and C. Jard, "Distributed monitoring of concurrent and asynchronous systems," *Journal of Discrete Event Systems*, vol. 15, no. 1, pp. 33 – 84, 2005.

[105] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, "Diagnosis of large active systems," *Artificial Intelligence*, vol. 110, no. 1, pp. 135–183, 1999.

[106] M. Daigle, X. Koutsoukos, and G. Biswas, "Distributed diagnosis of coupled mobile robots," in *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, May 2006, to appear.

[107] T. Liu, E. Ko, and J. Lee, "Intelligent control of dynamic systems," *Journal of the Franklin Institute*, vol. 330, no. 3, pp. 491–503, 1993.

[108] S. Gentil and J. Montmain, "Hierarchical representation of complex systems for supporting human decision making," *Advanced Engineering Informatics*, vol. 18, pp. 143–159, 2004.

[109] M. Daigle, X. Koutsoukos, and G. Biswas, "Distributed diagnosis in formations of mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 353–369, 2007.

[110] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.

[111] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice. New York*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. Springer-Verlag, New York, 2000.

[112] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[113] A. K. Samantaray and B. O. Bouamama, *Model-based Process Supervision: A Bond Graph Approach*. Springer, 2008.

[114] C. Sueur and G. Dauphin-Tanguy, "Bond graph approach to multi-time scale systems analysis," *Journal of The Franklin Institute*, vol. 328, no. 5/6, pp. 1005–1026, 1991.

[115] H. Mayeda, "On structural controllability theorem," *IEEE Transactions on Automatic Control*, vol. 26, no. 3, pp. 795–798, 1981.

[116] S. Narasimhan, R. Dearden, and E. Benazera, "Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems," in *Proceedings of the* 15^{th} *International Workshop on Principles of Diagnosis*, 2004.

[117] C. Frogner and A. Pfeffer, "Discovering weakly-interacting factors in a complex stochastic process," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 481–488.

[118] B. Ng and L. Peshkin, "Factored particles for scalable monitoring," in *Proceedings of the* 18^{th} *Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2002, pp. 370–377.

[119] M. W. Hofbaur and B. C. Williams, "Hybrid estimation of complex systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 34, no. 5, pp. 2178–2191, 2004.

[120] ——, "Hybrid diagnosis of unknown behavioral modes," in *Proceedings of the* 13^{th} *International Workshop on Principles of Diagnosis*, 2002.

[121] G. Biswas, X. Koutsoukos, A. Bregon, and B. Pulido, "Analytic redundancy, possible conflicts and TCG-based fault signature diagnosis applied to nonlinear dynamic systems," in *Proceedings of the* 7^{th} *IFAC Symposium on Fault Detection, Supervision, and Safety of Technical Processes (SAFEPROCESS 2009)*, 2009, pp. 1486–1491.

[122] C. S. Kam and G. Dauphin-Tanguy, "Bond graph models of structured parameter uncertainties," *Journal of the Franklin Institute*, vol. 342, no. 4, pp. 379–399, 2005.