VANDERBILT UNIVERSITY

INSTITUTE FOR SOFTWARE
INTEGRATED SYSTEMS

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, 37235

System Diagnosis using Hybrid Failure Propagation Graphs

Sherif Abdelwahed Gabor Karsai Gautam Biswas

TECHNICAL REPORT

ISIS-02-302

System Diagnosis using Hybrid Failure Propagation Graphs[*]

Sherif Abdelwahed
sherif.abdelwahed@vaderbilt.edu

Gabor Karsai
gabor@vuse.vanderbilt.edu

Gautam Biswas
biswas@vuse.vanderbilt.edu

Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN, 37203

Abstract

This technical report introduces an approach for robust diagnosis of a general class of multi-mode systems. The proposed approach is based on a temporal mode-dependant failure propagation model referred to as hybrid failure propagation graph (HFPG). The HFPG model is a labeled graph that represents failure conditions and their propagation effect (causal consequences) as causal relations with timing and mode switching properties. The proposed approach targets a general class of systems with both time and event driven dynamics such as hybrid and discrete event systems.

1 Introduction

Large engineering systems such as manufacturing systems, power networks, and chemical plants are usually designed for autonomous or semi-autonomous operation. Automated diagnosis and control forms a necessary part of these systems. Accurate and speedy diagnosis of faults is vital to their health and efficiency. In general, diagnostic modules aim to detect (recognize the occurrence of fault), isolate (identify faulty components) and estimate (determine the parameter value of faulty components) system failures by observing signals and measurements from the system sensors and actuators, comparing it with a model representing nominal and/or faulty behavior, and explaining the observed behavior in terms of a set of hypotheses about possible changes to the parameters of the system components.

For diagnosis, two kinds of modeling paradigms have been commonly used to describe the behavior of engineering systems: *analytical models* and *fault models*. Analytical models such as difference and differential state equations, finite state machines, and hybrid automata are used to describe the nominal (correct) system behavior. The choice of the model depends on the physical characteristics of the system and scope of analysis. Modern diagnosis approaches infer fault occurrences by comparing the observed behavior of the system with the given analytical model. From the analytical (model-based) diagnosis viewpoint, the system model defines a consistency relationship between the system behavior and its parameter. A fault occurrence reflects a change in the consistency relationship in which the observed behavior no longer corresponds to the one defined by the nominal system model. Model-based diagnosis involves monitoring the set of measured variables, detecting inconsistencies between measured and nominal behavior, and linking these inconsistencies to changes in specific model parameters.

[*] Funded, in part, by DARPA's Software-Enabled Control Program under AFRL contract F33615-99-C-3611.

The analytical approach, however, depends on the availability of a precise mathematical model which is difficult to obtain for many practical real-life systems. Even when a precise model can be obtained the computational requirements of model-based diagnosis procedures are usually prohibitive. To address the complexity in most engineering systems, researchers have used abstraction techniques to reduce the complexity of the model while preserving relevant information regarding the system behavior. In multiprocess systems, distributed model-based reasoning techniques have been used also to reduce the complexity of diagnosis algorithms by limiting the analysis to individual system components.

On the other hand, associative models such as fault trees, cause-consequence diagrams, diagnosis dictionaries, and expert systems describe system behavior when faults are present [1, 6, 7]. Typically, association-based models emulate a human expert diagnosing faults and are used for diagnosis of complex systems which can not be modeled analytically. The underlying fault models usually describe qualitatively the causal relationship (dependency) between observed signals and failure sources. Sensors signals are used to reason about possible failure based on the given causal relationship.

Fault models help in diagnosis by reducing the diagnostic search space. Hypothesis generation is straight-forward - just consider all the failure modes that could have caused the discrepancies. Diagnosing with a single fault assumption is simple. Diagnosing with multiple faults and/or sensor failure assumption can possibly result in a large number of combinations of faults to be examined. In this case, some reasonable heuristics can be used which are derived from the fault characteristics. Using fault models is more common in practice due its simplicity and computational efficiency. Associative fault models can be enriched to handle temporal, probabilistic and dynamical specifications. Also, support for integrated diagnosis of hierarchical systems can be easily established.

In this paper, we present a qualitative approach to failure diagnosis based on a temporal fault model referred to as timed failure propagation graph. Timed failure propagation graphs (TFPG) [3, 4] are causal models that describe the system behavior in presence of faults. The TFPG model is closely related to the fault model presented in [5, 2] and used for an integrated fault diagnoses and process control system. We extend the basic TFPG model to handle mode-switching systems, in which the system model depends on a set of possible operation modes. The extended structure, referred to as hybrid failure propagation graph (HFPG) captures the effect of the switching dynamics and timing constraints on the propagation of failures in typical discrete event and hybrid systems. The HFPG model adds mode dependency constraints on the propagation links which can be used to handle failure scenarios in hybrid and switching systems. The HFPG model supports both AND and OR propagation semantics which can be used to build complex failure propagation dependency situations. The proposed extension also allows cyclic dependency between signals (discrepancies) in the fault model.

The paper is organized as follows. Section 2 introduces the notation and terminology that are used throughout the report. In Section 3, an informal description of temporal failure propagation graph models is given. The hybrid failure propagation graph model is introduced in Section 4. Section 5 presents the formal description of the diagnosis problem and the main elements of the diagnostic system based on the hybrid failure propagation graph settings. In section 6, the diagnosis reasoning algorithm is introduced together with complexity analysis of its main procedures.

2 Notation and Terminology

In this section some of the terms and concepts used in this report are defined and discussed. A *component* is part of the physical hardware assembly of the system. It may refer to a single component like a pipe or an assembly of components, e.g., a pump assembly. A system may have different modes of operations, referred to as *system modes.* The nominal and faulty behavior of the system depends on the current mode of operation and therefore different models may be needed for each mode. Typically a system has to stay in each mode for a finite non-zero amount of time before switching to another mode. It is assumed that the number of possible mode-switching in any finite time interval is finite.

A *failure mode* is a failure of a component. A component may have more than one failure mode, i.e., a component may fail in more than one way. When a component malfunctions, we say that a failure mode of the component has occurred. The occurrence of a failure mode is called a *fault.* A component which exhibits one or more failure modes is referred to as a faulty component. A component which is not malfunctioning (none of the failure modes have occurred) is called a *healthy* component.

A fault in a component will produce anomalies in system behavior. These anomalies are called *discrepancies.* A discrepancy may be immediately observable or it may go unobserved depending upon sensor allocation and fault detection algorithms used. The generic term for a failure mode, a fault and a discrepancy is failure. When the term failure is used, it should be clear from the context what is meant; otherwise it will be explicitly stated.

Fault *detection* means determining that there is something wrong with the system. Usually, faults are detected by observing the values of physical variables in the system and then deducing that one or more discrepancies exist, which implies that one or more faults in some components have occurred. Once a discrepancy is observed, i.e., a fault has been detected, it needs to be diagnosed. Fault *diagnosis* means identifying the faults, i.e., locating the physical components that are not functioning properly. The diagnostic result consists of a set of one or more components in the system that are believed to be faulty.

Correctness of the diagnostic results means that only those components that are actually faulty are identified as faulty and no healthy component is part of the diagnostic result. *Completeness* of results means that all the components that are faulty are indicated.

Sensors are those components in the system that are used to measure values of physical variables like temperature, pressure, etc. The signals generated by these sensors can be used for control and monitoring. The fault detection algorithms can use these signals to determine whether a discrepancy exists or not. Sensor failure means that the sensor is given a wrong value (within a given accuracy limits) for the associated variable or parameter.

An *alarm* is an indication that a discrepancy has occurred. The alarm is said to *signal* when a discrepancy is observed. A discrepancy which has an alarm assigned to it is called a *monitored* discrepancy, while the ones without alarms are called *non-monitored* discrepancies. An alarm may become *silent* after ringing for a while (the discrepancy may not exist after a while, possibly because of a repair action). The ringing and silencing of alarms introduce *events* that trigger the diagnostics. All the events have a time stamp associated with them, signifying the time that the status of the discrepancy changed. Alarms may fail by either being silent while it should be ringing or ringing while should be silent, in this case it is called *false alarm.*

Sensor and alarm failures can lead a diagnoser astray and the diagnostic results can be incorrect and/or incomplete. A *robust* diagnostic system should be able to handle observation errors. By a diagnostic system that can handle observation errors we mean a system that will, ideally, be able to interpret the (possibly erroneous) observations properly and come up with the correct and complete diagnostic result. At worst, the diagnostic system should degrade gracefully as the number of observation errors increases. An important issue of the diagnosis of large scale systems is that of efficiency of the diagnostic algorithm. An algorithm of exponential complexity is not scalable. Thus an efficient diagnostic algorithm should be able to handle observation errors and also be of polynomial complexity.

Diagnosability of a system, in its most general sense, means that property of the system which allows the faults in the system to be detected and diagnosed in a timely manner, that is, within a finite interval from the time at which the failure occur. In order to characterize the diagnosability of a system, one needs to develop some criteria or metrics, which express the property of diagnosability in a reasonable and coherent manner.

3 Temporal Failure Propagation Models

There are three main aspects to the failure propagation models: failure modes of physical components, temporal and mode switching dependencies, and discrepancies in functionalities and their associated sensors. In the following we will describe the intuitive physical meaning of each aspect and its rule in the overall failure propagation model.

3.1 Failure Propagation

The occurrence of a failure mode causes one or more discrepancies in the system. These discrepancies usually appear as out of range physical variables. For instance, an output valve of the pump assembly, when stuck closed, causes the output flow rate to drop and the internal pressure to build up. Because the physical variables are related to each other, an out of range physical variable may cause some more physical variables to go out of limits. For example, a rise in temperature inside a gas container will cause the internal pressure to build up. Further, an out of range physical variable can cause a fault in a physical component, e.g., high pressure may lead to a leak in a pipe.

This phenomenon of causation between failure modes and discrepancies is called failure propagation. We say that the antecedent failure (failure mode or discrepancy) propagates to the consequent discrepancies. Following the chain of antecedent and consequent failures, we can enumerate the failure propagation paths starting from any given failure. The failure propagation paths are the mechanisms by which a fault in one part of the system can cause discrepancies to occur in another "remote" part.

Due to the dynamics of the system, the failure propagations do not take place instantaneously; instead they take a finite amount of time. For example, a "heater broken high" will take some finite amount of time to cause the temperature to rise beyond acceptable limits. Further, in any real system, the time taken can not be specified exactly. However, the minimum and maximum time that a propagation takes can be known fairly accurately, allowing us to use a time interval to express the uncertainty. To incorporate the dynamics into fault models, each failure propagation is parameterized with a time interval $[t_{min}, t_{max}]$, called propagation interval, which gives the minimum and the maximum time that the antecedent failure will take to cause the consequent failure. In another words, assuming that the propagation link is active, the failure cannot reach the destination of the propagation link before t_{min} time from the time it reached its source and had to reach the destination before t_{max} time from the time it reached its source

Many real life systems have several operation modes. Each mode is characterized by a smooth and continuous energy flow between the physical components of the system. Mode changes corresponds to discontinues evolution of some of the system variables which typically results in a sudden changes in the flow. Consequently, and due the fact that failure take a finite amount of time to propagate, failure propagation may change direction as a result of mode switching in multi-mode systems. For example, depending on the state of the valve connecting two pipes, the failure effect on one pipe may or may not propagate to the other pipe. To incorporate mode switching effects into the fault model, propagation links are parameterized with the set of mode at which they are active on. Under the assumption that mode switching is totally observable, one can always determine if a given fault can propagate from one part of the system to another.

The interactions between failure modes and discrepancies can be represented pictorially, as shown in Figure 1 for a system with two three components C1, C2 and C3 and two modes of operations A and B. There are four failure modes in the systems FM1 to FM4. The discrepancies of first component, C1 are D1, D2 and D3; of C2, D4, D5, D6 and D7; of C3, D8, D9, D10. The system also contains a discrepancy, D11, that is not associated with any component. In the shown diagram rectangle boxes represent the failure modes while the circles represent the discrepancies. The arrows between the nodes represent failure propagation. All the propagations shown are parameterized with propagation interval $[t_{min}, t_{max}]$ as well as the activation modes. Here t_{min} is the minimum time for propagation of failure along the edge, and t_{max} which is the maximum time for propagation of failure along the edge. Activation modes are not shown for propagation that are always active irrespective of the current system mode.

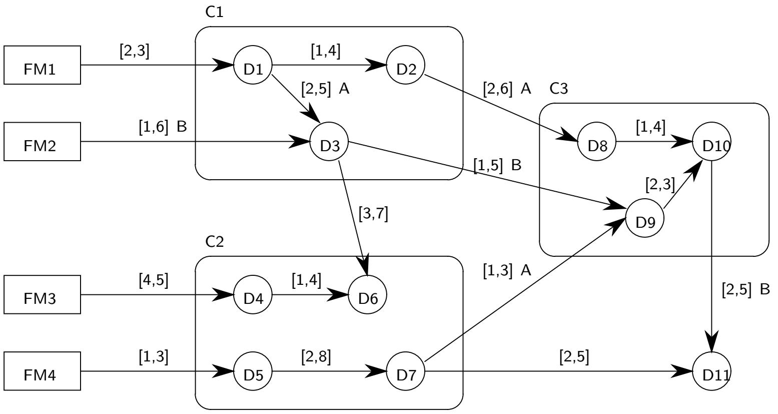
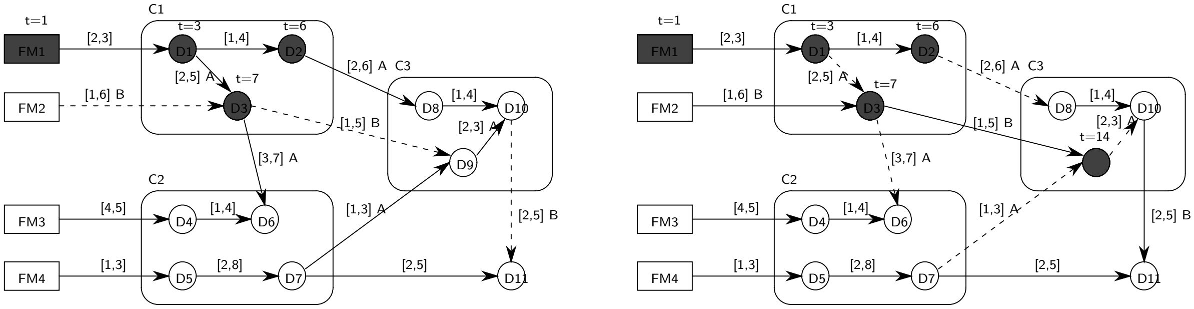


Figure 1: Pictorial representation of failure propagation

Figure 2 shows the propagation of failure mode FM1 at different time instances and with respect to several mode switches occurring at the boundaries of these time intervals. The Figures show the discrepancies (dark circles) that will occur at the given time instances (shown above the circles) as the faults propagate. The left diagram shows the discrepancies that will signal after 10 sec of operation given that the failure mode FM1 occur after 1 sec of operation. During the first 10 sec of operation the system was in mode A. Inactive propagation links, during this period, are shown as dashed lines. The right diagram shows the situation after 16 sec of operation where the system operates at mode B during the period $(10, 16]$.



At t =10, Mode= A for $t \in [0, 10]$

At t =16, Mode=B for $t \in]10, 16]$

Figure 2: Propagation of faults

The above scenario demonstrates the intuitive rule of fault propagation; a fault should propagate from node A to node B within the given time interval counted from the time when the failure reached node A (alarm at A is signaling) and the link between A and B becomes active. For instance, the failure effect that reached discrepancy D3 could not propagate to D9 during the first period $t \in [0, 10]$ as the corresponding link (D3,D9) is inactive for mode A. However, when the system switches to mode B, the failure effect takes 4 seconds to propagate from D3 to D9. That is, in this case, the failure effect starts to propagate at time $t = 10$ when the link is activated at mode B. Another interesting case happened when the failure effect reached D2. The failure effect starts to propagate at time $t = 6$ but could not reach D8 before the link becomes inactive due to mode switching at time $t = 10$.

For fault diagnosis, the spatial and temporal pattern of discrepancies can be used to isolate the faults. An inconsistency in the pattern can be used to detect sensor failures. For diagnosability studies, it should be possible to determine the relative importance of a discrepancy for detecting and/or diagnosing a failure mode and the time periods involved.

3.2 Failure Monitoring

The observation of anomalies in system behavior is provided by sensors. This section discusses their role in diagnostics and how they are modeled. Sensors measure the values of physical variables and provide signals. These signals can be used for control or they can be used for monitoring the health of the system. Following the dichotomy of physical and functional structure, we model sensors as physical components that monitor the functional failures.

From the diagnostic point of view, sensors provide evidence about the existence of discrepancies. These evidences are called alarms, which have been defined earlier in section 2. Sensor allocation describes which sensors are used for which alarms. However, this representation is not comprehensive, in that, it does not allow one to describe exactly how the sensor values relate to the alarms and discrepancies. This relationship can be modeled by using sensor states. A sensor in the system typically provides continuous valued readings over a wide range. The continuous range of sensor values can be divided into a set of ranges (not necessarily disjoint) that cover the sensor range which are called sensor states. When a sensor value is within one of these sub-ranges, the sensor is said to be in the state corresponding to that sub-range, or, that the particular sensor state has "occurred".

For example, a temperature sensor might have a continuous range from 20^oC to 90^oC. Typically, the sensor states for this sensor might be `Temp-Zero`, `Temp-Low`, `Temp-Nominal`, `Temp-High` and `Temp-Full`, representing ranges 20^oC-25^oC, 26^oC-40^oC, 41^oC-70^oC, and 71^oC-90^oC respectively. Many sensors, on the other hand, are used to display a binary condition in which the off state corresponds to the normal operation and the on state corresponds to the off-normal one. For example, a pressure sensor in a tank may indicate a normal pressure (specified as a range of possible values) or an abnormal pressure situation which can be above or below (or both) the normal range.

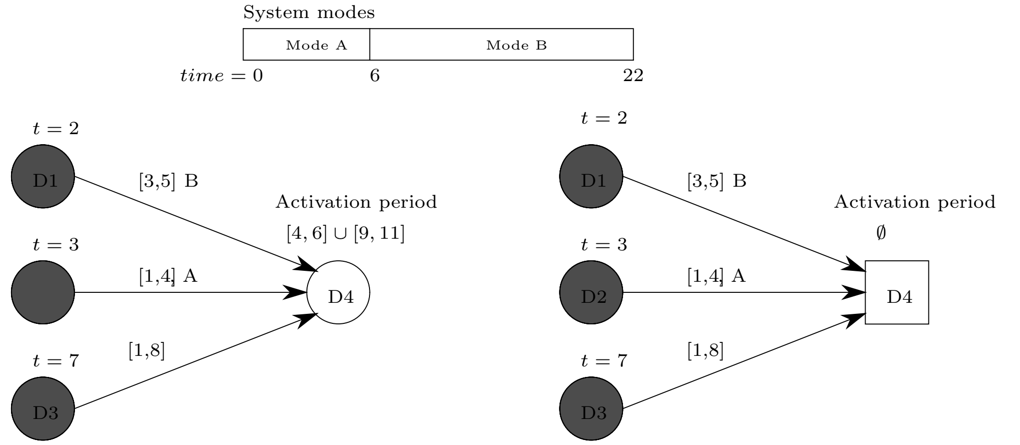
There is no prescription as to the number and kinds of states that a sensor can be in. There can be as many sensor states as are required to model the monitoring scheme. A sensor can be in only one of these states at any given time. Whenever the sensor is reading values that correspond to a particular state, the sensor state is said to be active. Since the sensor can be in only one state at a given time, all the other sensor states corresponding to the sensor are said to be inactive.

A discrepancy causes one or more sensors to be in particular state(s). Thus there is a causal relationship which goes from discrepancies to sensor states. During diagnosis, by examining the combinations of current states of the sensors, it can be ascertained if a discrepancy exists. Thus, the monitoring mechanism for a discrepancy can be modeled by specifying the alarm on that discrepancy or by listing the sensor states that the discrepancy impacts. The alarm representation does not give any information about how an alarm is generated from the sensors, while sensor state representation does.

Sensor states can be considered to be primitive discrepancies and the causality between discrepancies and sensor states can be modeled as failure propagations even though, strictly speaking, sensor states are not failures. In fact, sensor states can be converted to a set of binary conditions each represented by a discrepancy. For instance, the states `Temp-Zero`, `Temp-Low`, `Temp-Nominal`, `Temp-High` and `Temp-Full` in the above example can be converted to four discrepancies each corresponds to one of the above abnormal states and can take the values of either `ON` or `OFF` where `ON` indicates the condition of the underlying state is true while `OFF` means the condition is false[1]. This approach allows us to integrate the causality into fault models more coherently. To simplify the presentation, in the following we will assume that sensor states are incorporated into the failure propagation model as monitored or non-monitored discrepancies. However, the current model and diagnosis algorithm can be easily extended to handle sensor states directly.

[1] Note that, in general, the conditions defining the discrepancies are not necessarily disjoint.

In the fault model the state of a discrepancy is evaluated based on the condition of its parent nodes which could be either fault modes or other discrepancies. One can distinguish between two primitive forms at which the discrepancy can be affected by its parents; `AND`, and `OR`. A monitored discrepancy of type `AND` can only be triggered if the all its parents nodes are triggered and enough time has been elapsed so that the signals from the parent alarms could reach the child discrepancy node. The same applies to non-monitored discrepancies of type `AND` except that no alarm is triggered when the discrepancy condition becomes true (discrepancy state= `ON`). A discrepancy of type `OR` will be activated by any of its parent. In the failure propagation graph we AND type discrepancies will be depicted as squares.



Case1: Failure propgarion to an OR discrepancy

Case2: Failure propgarion to an AND discrepancy

Figure 3: The effect of the discrepancy type on the propagation of fault

The activation of both types of discrepancies is affected by the propagation delay and possible mode switchings. Consider for instance the situation shown in the Figure 3. The system has two modes of operation A and B. In this scenario the system remains in mode A for the first 6 sec and then switches to mode B. In the first case, the OR discrepancy D4 can be activated by D1 during the period $[9, 11]$, by D2 during $[4, 6]$, or by D3 during $[8, 15]$. Therefore, the earliest time D4 can be activated is $t = 4$ and the latest time is $t = 11$ and D4 cannot be activated in the period $[6, 8]$. In the second case, D4 cannot be activated during the given period $[0, 22]$ due to the fact that there is no common time at which the failure effect from D1,D2, and D3 can reach D4. D4 can be activated, however, if the propagation link between D2 and D4 is mode independent. In this case, the failure effect at D2 can propagate to D4 in the period $[4, 7]$ and therefore, D4 can be activated in the period $[9, 11]$.

The combined fault model with failure modes, discrepancies, alarms, etc. is called the Failure Propagation Model (FPM). The pictorial representation used for failure propagations can now be extended to include alarms. As before, the square boxes represent the failure modes of components and the circles represent the discrepancies. The bold line circles are discrepancies that are monitored with an alarm. The fine line circles are discrepancies that don't have any alarm explicitly associated them.

4 The Hybrid Failure Propagation Graph Model

In this section we will present a formal model for failure propagation, referred to as *hybrid failure propagation graph*, that represent the the failure aspects discussed in the previous section. The hybrid failure propagation graph (HFPG) is a labeled directed graph where the nodes represent either failure modes - which are fault causes - or discrepancies - which are off-nominal conditions that are the effects of failure modes. Discrepancies can either be monitored (attached to alarms) or non-monitored, and depending on the way it is triggered by the incoming signals it is further classified as either `AND` or `OR` discrepancy. Edges between nodes in the graph represent failure propagation. The interval attribute of a failure propagation edge specifies the upper and lower constraints on the time it will take for the failure to propagate from the source to the destination node.

The HFPG model allows the representation of failure propagation in multi-mode (switching) systems in which the failure propagation depends on the current mode of the system. To this ends, edges in the graph model can be constrained to a subset of the set of possible operation modes of the system. Formally, a hybrid failure propagation graph model is represented as a tuple $G = (F, D, E, M, \mathsf{ET}, \mathsf{EM}, \mathsf{DC}, \mathsf{DS})$, where:

- F is a nonempty set of failure nodes,
- D is a nonempty set of discrepancy nodes, with $F \cap D = \varnothing$,
- $E \subseteq V \times V$ is a set of edges connecting the set of all nodes $V = F \cup D$. We will write $src(e)$ and $dst(e)$ for the source and destination nodes of the edge e, respectively,
- M is a nonempty set of system modes. We assume that at each time instance t the system can be in only one mode.
- $\mathsf{ET} : E \to I$ is a map that associate every edge in E with a time interval in $I = \{[t_{min}, t_{max}] \mid t_{min} \in \mathbb{R}_+,\ t_{max} \in \mathbb{R}_+ \cup \{\infty\},\ t_{min} \leq t_{max}\}$ is the set of all finite time intervals,
- $\mathsf{EM} : E \to \mathcal{P}(M)$ is a map that associate every edge in E with a set of modes in M (we assume that $\mathsf{EM}(e) \neq \varnothing$ for any edge $e \in E$),
- $\mathsf{DC} : D \to \{\mathtt{AND}, \mathtt{OR}\}$ is a map defining the class of each discrepancy as either `AND` or an `OR` node,
- $\mathsf{DS} : D \to \{\mathtt{ON}, \mathtt{OFF}\}$ is a map defining the monitoring status of the discrepancy as either `ON` for the case when the discrepancy is monitored by an online alarm or `OFF` for the case when the discrepancy is not monitored.

The set V contains $n + m$ vertices, representing n failure modes and m discrepancies. Some of the discrepancies are monitored as defined by the map DS. The set of monitored discrepancies will be denoted D_a. An edge $e = (v, v') \in E$ iff the failure effect represented by the node v can propagate and participate in causing the effect represented by the node v'. The map ET associates each edge $e \in E$ with the minimum and maximum time (given as interval) for propagation of failure along the edge. We will write $t_{min}(e)$ and $t_{max}(e)$ for the minimum and maximum time for failure propagation along the edge e, respectively, so that $\mathsf{ET}(e) =$

$[t_{min}(e), t_{max}(e)]$. That is, given that a propagation edge is enabled (active), it will take at least (most) t_{min} (t_{max}) time for the fault to propagate from the source node to the destination node. The map EM associates each edge $e \in E$ with a subset of the system modes at which the failure can propagate along the edge. Consequently, the propagation link e is enabled (active) in a mode $m \in M$ if and only if $m \in \mathsf{EM}(e)$. The map DC defines the type of a given discrepancy as either `AND` or `OR`. An `OR` type discrepancy node will be activated when the failure propagate to the node form any of its parents. On the other hand, an `AND` discrepancy node can only be activated if the failure propagates to the node from all its parents. We assume the following assumptions hold for the graph structure (V, E):

- $(\forall v \in V) \quad (v, v) \notin E$
- $(\forall e \in E) \quad dst(e) \notin F$
- $(\forall d \in D)(\exists v \in V) \quad (v, d) \in E$

The first assumption states that the graph does not contain self loops as the current version of HFPG only deals with persistent faults. The second assumption states that a failure node cannot be a destination of any edge so in effect failure nodes are the initial nodes of the graph. Finally, we assume that every discrepancy must the destination of an edge, that is a discrepancy must be caused by either another discrepancy or failure mode.

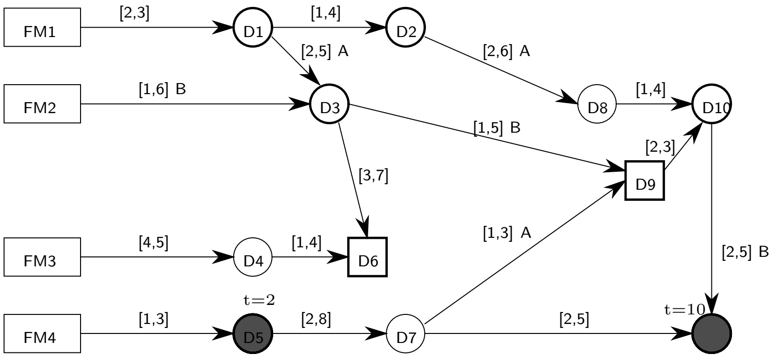


Figure 4: A hybrid failure propagation graph

Figure 4 shows a modified version of the failure propagation graph discussed in the previous section. As indicated earlier, rectangles in the hybrid failure propagation graph model represent the failure modes while circles and squares represent `OR` and `AND` type discrepancies, respectively. Monitored discrepancies are shown with bold lines. The arrows between the nodes represent failure propagation. Propagation edges are parameterized with the corresponding interval, $[t_{min}, t_{max}]$, and the set of modes at which the edge is active. The above figure shows also a sequence of alarm signals identified by shaded discrepancies. The time at which the alarm is observed is shown above the corresponding discrepancy.

5 The Diagnosis Problem

The diagnostic system operates on the HFPG model described in the previous section and characterizes the fault status (actual current state) of the system by hypothesizing about the faults in components and sensors based on the signals received from the sensors and the current mode of the system. The diagnoser uses the HFPG model and the timed sensor/mode-switching signals to generate a set of logically valid hypotheses of the current state of the system. The hypotheses are then ranked according to certain criteria that is generally based on the number of supporting alarms versus the number of inconsistent ones. A more advanced ranking that takes into account the relative significance/relaiablity of the sensor signals can also be established. The set of hypotheses with the highest rank will be selected as the most plausible estimations of the current state of the system. The diagnoser is implemented as a reactive module that is triggered by signals from the set of active sensors and mode-switching signals.

5.1 Diagnosis Strategy

The HFPG diagnoser (1) receives events, (2) generates hypotheses, and (3) selects some hypothesis(es) according to how consistently they explain the observations. For this it uses the principles of *parsimony* and *structural redundancy*. In the following, these two general principles are explained in the HFPG context.

Parsimony

A particular hypothesis is said to be consistent with the received events (observations) if the spatial and temporal constraints imposed by the propagation models are satisfied. If observation errors are possible, not all of the received events have to comply with the spatial and temporal constraints. Consequently, the number of hypotheses that are plausible under the given set of observed events will become larger. In the extreme case when the sensors and/or fault detection algorithms are completely unreliable, any fault hypotheses is plausible, since the observations do not carry information about the actual state of the system. Of course, in any realistic case, most of the observed events are directly related to underlying faults, therefore diagnosis is possible. The principle of parsimony suggests that the simplest explanation is the best. If a hypothesis can explain consistently all of the observed events, it should be considered more plausible than another one, which additionally requires the assumption of a sensor fault as well. Application of the principle of parsimony means that the set of plausible hypotheses should be minimal.

Structural redundancy

As we have discussed before, the physical interactions in dynamic systems impose spatial and temporal constraints on the observed events. In those parts of the system where failure propagation occurs, a single fault results in multiple manifestations. Obviously, these manifestations are not independent of each other. They provide a redundant observation of the fault. Because the failure propagation models primarily represent structural relationships in the systems, we call this redundancy structural redundancy. Due to the structural redundancy, events can confirm or contradict other events in a propagation model, therefore, a concept similar to that of the analytical redundancy approach can be developed. The idea is illustrated with the simple HFPG shown in Figure 5. The graph shown here includes only failure propagations. For the sake of simplicity, all of the discrepancies are of the OR type. Also, there is only one mode at which all edges are enabled and there is no temporal restriction on the failure propagation on any edge ($t_{min} = 0$ and $t_{max} = \infty$ for all edges). Discrepancies here are associated with a unique sensor, whose output signal is used by a monitoring algorithm to generate the alarm.

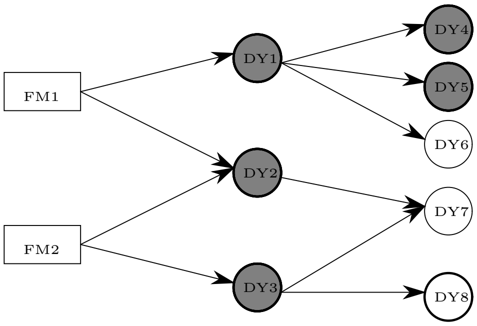


Figure 5: Use of structural redundancy for sensor fault detection

In the particular fault scenario in Figure 5, the alarms associated with DY1, DY2, DY3, DY4 and DY5 are ringing (shown as shaded in the graph), while the alarm assigned to DY8 is silent. The simplest explanation for the alarms at DY1, DY2, DY4 and DY5 is that FM1 is a fault source (parsimony). Possible explanation for alarm DY3 is that FM2 is also a fault source. However, if FM2 is a fault source, the alarm at DY8 should also ring (structural redundancy), therefore this hypothesis implies that the sensor at DY8 must be faulty as well. An alternative explanation is that the sensor associated with DY3 is faulty and is giving rise to a spurious alarm. This hypothesis is more plausible than the previous one, since it explains the alarm scenario with two fault sources (FM1 and the sensor associated with DY3) instead of three (FM1, FM2 and the sensor associated with DY8) (parsimony). A number of other explanations can be found for the alarm pattern that are all less plausible than the previous ones. Thus, structural redundancy means the use of the interdependence among the alarms in the reasoning algorithm. The actual reasoning method is considerably more complex than the illustrative example due to the temporal aspect of the HFPG model.

6 Elements of the Diagnostic System

The diagnosis system operates on the HFPG model of the system to detect and isolate faults by generating and selecting appropriate hypothesis to explain the incoming signals from the system. In this section we will discuss the structure of the failure hypothesis generated by the system to identify a possible state of the system. We also present the basic definition of system events and states that are used for hypothesis generation and reevaluation.

6.1 Failure Mode Hypotheses

The diagnoser responds to input signals by generating hypothesis. Each hypothesis is an evaluation of the status of a failure mode in the HFPG model together with the corresponding evidences. Formally, a hypothesis is a tuple $h_f = (f, t_e, t_l, r, SP, SS, I, M, P)$, where $f \in F$ is the failure mode for which the hypothesis stands, t_e and t_l are the estimated earliest and latest time of occurrence of the failure mode f. The static rank, r, of the hypothesis is number associated with a measure of belief in the hypothesis. The rank is set to 0 at the creation of the hypothesis and updated each time a new event is triggered. Hypotheses with negative ranks are not considered during the reasoning process. The elements SP, SS, CS, I, M, and P are sets of discrepancies with special relevance to the hypothesis h_f:

- $SP \subseteq D_a$ is the set of primary signalling discrepancies that support the hypothesis h. These are the active alarms that are triggered as an immediate consequence of f, or the ones that can only be explained based on the occurrence of f and does not require any other failure mode for explanation. These alarms are the main justification of the hypothesis h_f.
- $SS \subseteq D_a$ is the set of secondary signalling discrepancies that support the hypothesis h. These are the active alarms that are triggered as a consequence of alarms already explained as a consequence of f and are supporting the hypothesis h_f. Active alarms in the set SS can be explained based on the occurrence of f alone and do not require any other failure mode for explanation.
- $CS \subseteq D_a$ is the set of secondary signalling discrepancies that support the hypothesis h given that other hypothesis are valid. These are the active alarms that are triggered as either a consequence of a set of failure modes $F' \subseteq F$ where $f \in F'$ or as a consequence of alarms already explained as a consequence of F'.
- $I \subseteq D_a$ is the set of signalling monitored discrepancies that are inconsistent with the hypothesis h. These are the alarms that are connected to the failure mode f but cannot be explained based on the hypothesis h_f.
- $M \subseteq D_a$ is the set of silent monitored discrepancies that are inconsistent with the hypothesis h_f. These are the alarms that are connected to the failure mode f but should be singling according to the hypothesis h_f.
- $P \subseteq D_a$ is a set of pending discrepancies whose status cannot be identified at the current time. Pending discrepancies are silent monitored discrepancies that are expected to signal in the future according to the hypothesis h_f.

Note that the hypothesis h_f also implicitly provide an estimation of the status of the monitored alarms connected to the failure mode f. That is, h_f is also a hypothesis for monitored alarms connected to f. Under the hypothesis h_f, supported alarms are considered healthy (providing the correct signals) and inconsistent alarms are faulty. In addition to generating and updating hypotheses, the diagnoser also generates a list of false alarms, namely those alarms that could not be explained by any hypothesis based on the timing of events and the structure of the failure propagation graph.

Note that each hypothesis h_f considers only those discrepancies that are reachable from the underlying failure mode f. This allows the diagnoser system to deal with the sensors signals more efficiently by focusing on the nodes that are connected to the corresponding discrepancy. However, in general the TFPG structure may contain several failure modes that can propagate to certain common discrepancies. These mutual dependencies can lead to a conflict between the hypothesis of different failure modes, because as mentioned above h_f also implicity provides an estimation of the status of monitored discrepancies connected to it.

Consider for instance the HFPG graph in Figure 5, assume h_1 is a hypothesis about FM1 that considers DY2 as a primary supporting alarm and h_2 be a hypothesis about FM2 that considers DY2 as an inconsistent alarm. Then clearly both hypotheses cannot be part of a consistent set of hypotheses as one considers DY2 healthy while the other considers it faulty. Note that this situation is independent of the type of DY2, that is, the conflict between the two hypotheses remains if DY2 is either of type AND or OR. The diagnoser eliminates those sets of hypotheses that contain conflicting elements when generating the failure report. The exact definition of conflicting hypotheses will be discussed later in this technical report.

6.2 System Events

The current state the HFPG edges and nodes can change in reaction to system events. These changes can then trigger a set of hypotheses updates which include creating new hypothesis and/or reevaluating current hypothesis. There are two types of events that the can trigger a hypothesis update in the HFPG diagnosis system; physical and hypothetical events. A physical event correspond to observed signals from the system sensors, while a hypothetical event correspond to confirmed measurement inconsistencies according to a given hypothesis. The two event types are described below in more details.

There are two types of physical events that triggers a hypothesis change in the HFPG model: a signalling alarm event and a mode change event. Formally, a physical event is represented by the tuple $e = (x, t)$, where $x \in D_a \cup M$ is either a monitored alarm ($x \in D_a$) or a mode-switching signal ($x \in M$) and t is the time at which the signal is observed. We will write $\mathsf{Signal}(e)$ to identify the source of the event, and $\mathsf{Time}(e)$ to identify the time at which the event occur. Therefore, $e = (\mathsf{Signal}(e), \mathsf{Time}(e))$. An event e is triggered whenever the state of a discrepancy is changed or the system switches to a new mode. The diagnostic system keeps a record of the sequence of all timed events from the system initial start to the current time. As mentioned earlier, we write e_k to identify the kth event. The diagnostic system maintains a record of all physical events ordered by their time.

In contrast with physical event, a hypothetical event does not correspond to any new observation or measurement. A hypothetical event is generated based on the expectation of a given hypothesis regarding a future signal that should occur according to the hypothesis. Such event is referred to as a time-out event. Time-out events are internal event generated by the diagnostic reasoner based on the current hypothesis set. For a given hypothesis h_f, a time-out event will be issued at time t if a monitored discrepancy d_a was expected to signal by the time t according to h_f but it did not signal. In this case the time-out event is given by the tuple $o = (h_f, d_a, t)$.

6.3 Physical and Hypothetical States

The diagnostic reasoner generates hypotheses based on the notions of causality and temporal consistency. Both are defined over the current state of the HFPG nodes. Based on the way the state of a given node is evaluated we distinguish here between two types of states that are used to define the failure status of the system: physical states and hypothetical states. A physical state corresponds to the observed state of a monitored discrepancy, while a hypothetical state is the estimated state of a node in the HFPG model according to a given hypothesis. The relationship between there two types of states defines the consistency relationship between alarms and hypothesis. The two state types are described below in more details.

A physical state type can only be defined for monitored discrepancy, as it corresponds to the state of a node as either signalling or not. The physical state of the set of monitored discrepancies is given by a map $\mathsf{PState} : D_a \to \{\mathtt{ON}, \mathtt{OFF}\}$, which assigns to each monitored alarm $d \in D_a$ it current measured status which can be $\mathtt{ON}$ if the alarm is signalling, otherwise it is $\mathtt{OFF}$. We define another map $\mathsf{PTime} : D_a \to \mathbb{R}$ where $\mathsf{PTime}(d)$ is the time of the last change in the physical state of $d \in D_a$. The maps PState and PTime are time dependent and therefore we may write PState_k and PTime_k to denote the maps after the kth event, e_k. The script k will be removed if the evaluation time is clear from the context. Initially, the physical states of all alarms are set to $\mathtt{OFF}$ and the corresponding physical times are set to zero. That is,

$$(\forall d \in D_a) \quad \mathsf{PState}_o(d) = \mathtt{OFF}, \;\; \mathsf{PTime}_o(d) = 0$$

In general we will write $\mathsf{Time}(e_k)$ or simply $\mathsf{Time}(k)$ to denote the time at which the kth event e_k occur. Therefore, e_o is the event of starting the system.

A hypothetical state, on the other hand, is the state of an HFPG node according to a given hypothesis. This state type can be defined for any node in the HFPG model. Given a set of hypotheses H, the hypothetical state of a node with respect to a hypothesis $h_f \in H$ is given as a map $\mathsf{HState}^{h_f} : V \to \{\mathtt{ON}, \mathtt{OFF}, \mathtt{UDF}\}$, where $\mathtt{UDF}$ is used hereafter to denote undefined values. The map HState^{h_f} assigns to each node $v \in V$ it status according to the hypothesis h_f. The above map satisfies

$$(\forall v \in V) \quad v \notin \mathsf{Domain}(f) \to \mathsf{HTime}^{h_f}(v) = \mathtt{UDF}$$

Where $\mathsf{Domain}(f)$ denotes the domain of f, namely, the set of nodes that can be reached from the failure mode f at any system mode including the node f. Note that $\mathsf{Domain}(f)$ is a model property and does not depend on the current mode or time. The status of a node in the domain of f can be $\mathtt{ON}$ if the node should be active according to the hypothesis h_f, otherwise it is $\mathtt{OFF}$. Here the word "active" has different interpretation depending on the type of the node. If the node is a failure mode then active means that the associated failure must have happened according to h_f. Otherwise if the node is a discrepancy then active means "should be signalling" according to the hypothesis h_f.

We define another map $\mathsf{HTime}^{h_f} : V \to I \cup \{\mathtt{UDF}\}$ that assigns to each node in V the time interval in which the node must have been activated within according to the hypothesis h_f. Recall that I denotes the set of all finite time intervals. The map HTime^{h_f} also satisfies,

$$(\forall v \in V) \quad v \notin \mathsf{Domain}(f) \to \mathsf{HTime}^{h_f}(v) = \mathtt{UDF}$$

That is, the map HTime^{h_f} is only defined for nodes that are within the domain of the failure mode f. Note that the map HTime^{h_f} is defined for silent discrepancy in the domain of f,

$\mathsf{Domain}(f)$. In this case, $\mathsf{HTime}^{h_f}(d)$ shows the time interval at which the discrepancy d should be active according to h_f. This interval or a subset of it may occur in the future.

In general, a hypothetical state may be dependant on more than one hypothesis. Such situation is attributed to the existence of AND type discrepancies in the HFPG graph, particularly due to the fact that the state of an AND-type discrepancy depends on the combined states of all its parent nodes. There are several way to represent such multiple dependency. In this report this is modeled using the dependency set map $\mathsf{DSet}^{h_f} : V \to \mathcal{P}(H)$ which assigns to each node $v \in V$ the set of hypothesis that the hypothetical state $\mathsf{HState}^{h_f}(v)$ depends on. That is, the evaluation $\mathsf{HState}^{h_f}(v)$ is valid only in conjunction with the validity of the set $\mathsf{DSet}^{h_f}(v)$. If $\mathsf{DSet}^{h_f}(v) = \varnothing$ then the hypothetical state $\mathsf{HState}^{h_f}(v)$ is independent of any other hypothesis. Note that DSet^{h_f} always evaluate to empty set when there is only OR-type nodes in the HFPG model.

The maps HState^{h_f}, HTime^{h_f} and DSet^{h_f} are time dependent and therefore we may write $\mathsf{HState}_k^{h_f}$ and $\mathsf{HTime}_k^{h_f}$ to denote the maps after the kth event, e_k. We may remove the scripts k or h_f if the map evaluation is clear from the context. Note that there are no initial hypotheses for a HFPG, that is, $H = \varnothing$ when the system starts and therefore there are no hypothetical states initially.

In the HFPG model propagation edges can be disabled and enabled based on the current mode of the system. The diagnoser maintains a record of the status of the propagation edges. The physical state of the an edge $e \in E$ is given by a map $\mathsf{EState} : E \to \{\mathtt{ON}, \mathtt{OFF}\}$, which assign to each edge $e \in E$ it current status which can be ON if the edge is enabled, that is, failure can propagate through it, otherwise it is set to OFF. We define another map $\mathsf{ETime} : E \to \mathbb{R}$ where $\mathsf{ETime}(e)$ is the time of the last change in the physical state of the edge $e \in E$. The maps EState and ETime are also time dependent and therefore we may write EState_k and ETime_k to denote the maps after the kth event, e_k. The script k will be removed if the evaluation time is clear from the context. The initial state of the edges depends on the initial mode of the system. Note that the state of an edge is a physical (observable) state and is not subject to failure conditions.

7 The Diagnostic Reasoning Approach

Given a HFPG representing the failure propagation in the system and a sequence of sensors signals corresponding to monitored discrepancies and possible mode switches, the diagnosis problem is to generate a failure report which consist of a set of hypothesis that explains all the current signaling discrepancies. At the occurrence of every event, the diagnoser updates the set of hypothesis and the faulty components will be identified. The reasoning algorithm uses two basic data structures, the system current fault status and the HFPG model. The system fault status consists of the current set of fault hypotheses and the corresponding evidences. The diagnoser updates the set of possible hypotheses about the system state based on the causal and timing consistency between the discrepancies. Consistency between discrepancy nodes is calculated based on the type of the node and the current mode of the system.

7.1 Causality Relationship

The diagnoser reasoner generates hypotheses based on the notion of causality. Causality is a relation between the states of the nodes in the HFPG mode. There are two types of causalities depending on the type of the node `AND` or `OR`. Causality relationships are time dependent and therefore will be scripted by the current event index. The OR-causality at the kth event is denoted as OC_k. OR-causality is a relationship between the state of a discrepancy and the hypothetical state of one of its parent nodes, while AND-causality is a relationship between the state of a node and the states of all its parents.

Causality relationship are independent on the state type of the node. Therefore it will be described for a general state. To this end, we will write $\mathsf{State}(v)$, $\mathsf{Tmin}(v)$, $\mathsf{Tmax}(v)$ to denote the current state of the node and the limits of the time interval at which this state was last changed, respectively. In case of physical state, we have $\mathsf{Tmin}(v) = \mathsf{Tmax}(v) = \mathsf{PTime}(v)$, that is the interval reduces to a single time instance. Let $v', v \in V$ be two nodes in the HFPG model such that, $DC(v) = \mathtt{OR}$ and $(v', v) \in E$, that is v is a child node of v'. Assume that at time index k, v' hold a state $\mathsf{State}_k(v')$ and that v changed its state to a new state $\mathsf{State}_k(v)$ such that $\mathsf{Tmin}(v) \leq \mathsf{Time}(k) \leq \mathsf{Tmax}(v)$. Then $(\mathsf{State}_k(v'), \mathsf{State}_k(v)) \in \mathsf{OC}_k$ if all the following hold.

- $\mathsf{State}_k(v') = \mathtt{ON}$,
- $\mathsf{State}_k(v)) = \mathtt{ON}$,
- $\mathsf{EState}_k((v', v)) = \mathtt{ON}$,
- $t_{min}((v', v)) \leq \big(\mathsf{Tmin}(v) - \max\big(\mathsf{Tmax}(v'), \mathsf{ETime}((v', v))\big)\big)$,
- $t_{max}((v', v)) \geq \big(\mathsf{Tmax}(v) - \max\big(\mathsf{Tmin}(v'), \mathsf{ETime}((v', v))\big)\big)$

The above simple says that $\mathsf{State}_k(v')$ (possibly) caused the current change of the $\mathsf{State}_k(v)$ to `ON` if v' is also `ON` and the link between v' and v is currently enabled and the time it takes for the fault to propagate for v' to v is consistent with the timing attributes of the propagation link and the time this link is enabled. OR-causality is illustrated in the following graph.

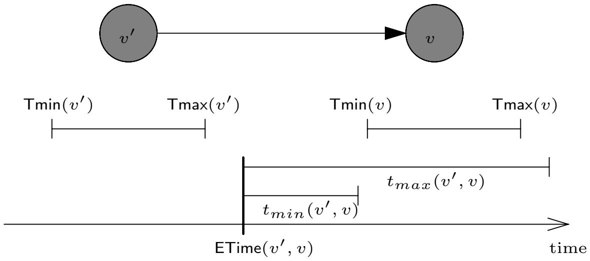


Figure 6: OR-Causality relationship

AND-causality can be defined similarly. Let $v \in V$ be a node in the HFPG model such that $\mathsf{DC}(v) = \mathtt{AND}$. Assume that at time index k each parent of v, v' hold a state $\mathsf{State}_k(v')$

and that v changed its state to a new state $\mathsf{State}_k(v)$ such that $\mathsf{Tmin}(v) \leq \mathsf{Time}(k) \leq \mathsf{Tmax}(v)$. Write $\mathsf{State}_k(V')$ to denote the conjunction of the individual states of each $v' \in V$. That is, $\mathsf{State}_k(V') = \{\mathsf{State}_k(v') \mid v' \in V'\}$. Then $(\mathsf{State}_k(V'), \mathsf{State}_k(v)) \in \mathsf{AC}_k$ if all the following hold

- $(\forall v' \in V')\ \ \mathsf{State}_k(v') = \mathtt{ON}$
- $\mathsf{State}_k(v)) = \mathtt{ON}$,
- $(\forall v' \in V')\ \ \mathsf{EState}_k((v', v)) = \mathtt{ON}$,
- $(\forall v' \in V')\ \ t_{min}((v', v)) \leq \big(\mathsf{Tmin}(v) - \max\big(\mathsf{Tmax}(v'), \mathsf{ETime}((v', v))\big)\big)$,
- $(\exists v' \in V')\ \ t_{max}((v', v)) \geq \big(\mathsf{Tmax}(v) - \max\big(\mathsf{Tmin}(v'), \mathsf{ETime}((v', v))\big)\big)$

The above conditions says that the $\mathsf{State}_k(V')$ (possibly) caused the current change of the $\mathsf{State}_k(v)$ to $\mathtt{ON}$ if state of every $v' \in V'$ is also $\mathtt{ON}$ and the link between every v' and v is currently enabled and the time it takes for the fault to propagate for v' to v is consistent with the timing attributes of the propagation link. AND-causality is illustrated in Figure 7.

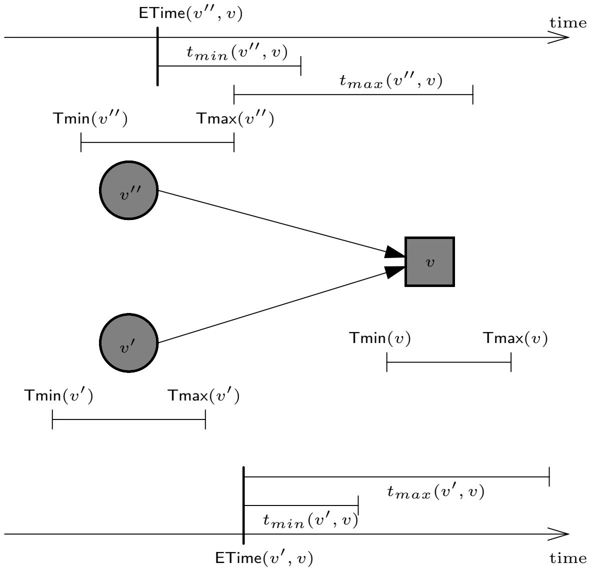


Figure 7: AND-Causality relationship

Note that the conditions for AND-causality require that the minimum time it takes for the fault to propagate from any parent node $v' \in V'$ to the node v is greater than the minimum propagation time of the corresponding link. However, there is no "similar" restriction regarding the maximum propagation time. It is only required that the maximum time it takes for the fault to propagate from one of the parents node $v' \in V'$ to the node v is less than the maximum propagation time of the corresponding link. Intuitively, this condition ensures that the node v' will not be activated until it receives all the failure effects from its parents. Note that it is required that all parent links between v and its parent nodes are enabled. That is, an `AND`-type node cannot be activated in a mode that disables a propagation link from a parent node.

7.2 Temporal Consistency and Hypothesis Update

The diagnostic reasoner receives inputs from the alarm and system-mode monitors and update the physical state of the system accordingly. This is achieved by changing the map PState or EState to reflect the new measurement. Based on the change of the system state, the reasoner tries to explain the new state by updating the set of hypothesis to explain the new state. This update includes creating new hypothesis and/or reevaluating the current set of hypotheses. Based on the principle of parsimony, the reasoner would not create a new hypothesis unless the new state cannot be explained using the current set of hypothesis. In addition, the reasoner will try to limit hypotheses reevaluation to a minimum.

Consider the event $e_k = (d, t)$ indicating that a monitored discrepancy d has changed its state from $\mathtt{OFF}$ to $\mathtt{ON}$ at time $t = \mathsf{Time}(e_k)$. Assume that d is of OR-type. That is, d is an OR-type monitored discrepancy that is triggered at time index k. Then the signalling discrepancy d is said to be *temporally consistent* with the hypothesis h if

$$(\exists v' \in \mathsf{Parents}(d)) \;\; (\mathsf{HState}_k^h(v'), \mathsf{PState}_k(d)) \in \mathsf{OC}_k$$

Where $\mathsf{Parents}(d)$ denotes the set of parents of the node d in the HFPG model. That is one of the parents of v has a hypothetical state with respect to h that is OR-causal to the current physical state of d.

Consistency with respect to AND-type nodes can be defined similarly. However, for AND-type nodes consistency depends on the AND-causality relationship. Let d be an AND-type monitored discrepancy that is triggered at time index k. Then the signalling discrepancy d is said to be *temporally consistent* with the hypothesis h if

$$(\mathsf{HState}_k^h(\mathsf{Parents}(d)), \mathsf{PState}_k(d)) \in \mathsf{AC}_k$$

The above condition requires that all the parents of d has a hypothetical state with respect to h_f that is AND-causal to the current physical state of d. Note that in the above definition, all the parents of d must have a hypothetical state with respect to the same hypothesis h. In above conditions, it is assumed that the hypothetical state of the parent(s) does not depends on any other hypothesis, that is, $\mathsf{DSet}^h(v') = \varnothing$, where v' is the corresponding parent of d.

When d is explained by h the correspond state and time maps is then updated by setting $\mathsf{HState}_k^h(d) = \mathsf{PState}_k(d) = \mathtt{ON}$, and $\mathsf{HTime}_k^h(d) = [t, t]$. Consequently, the monitored discrepancy d is added to the set of secondary supporting alarms of h and the rank of h will be incremented accordingly. In following we will refer to temporal consistency between nodes simply as consistency. The consistency relationship at time index k will be represented by the predicate $\mathsf{Consis}_k \subseteq H_k \times V$, where H_k denotes the set of hypothesis at time index k. That is, $\mathsf{Consis}_k(h, v)$ is true when the node v is consistent with the hypothesis h.

Consistency between nodes as defined above for a given hypothesis h_f is absolute in the sense that it does not depend on any other hypothesis. However, it is possible that consistency between a node state and parent node state(s) depends on more than one hypotheses. Such dependency originates from the nature of AND-type alarms and can propagate to OR-type alarms. A consistency relation that depends on more that one alarm is referred to as *conditional consistency*. The conditional consistency relationship at time index k will be represented by the predicate $\mathsf{DConsis}_k \subseteq H_k \times V \times \mathcal{P}(H_k)$. That is, $\mathsf{DConsis}_k(h, v, H')$ is true when the node v is conditionally consistent with the hypothesis h given the set of hypothesis $H' \subset H_k$.

Conditional consistency is defined formally as follows. Let d be an AND-type monitored discrepancy that is triggered at time index k. Then the signalling discrepancy d is said to be *conditionally consistent* with the hypothesis h given the set of hypothesis $H' \subset H_k$ if all the following holds

- $(\forall v_i \in \mathsf{Parents}(d))(\exists h_i \in H_k) \;\; (\{\mathsf{HState}_k^{h_i}(v_i) | i \in I\}, \mathsf{PState}_k(d)) \in \mathsf{AC}_k$
- $(\exists i \in I) \;\; h_i = h$
- $H' = \left(\bigcup_{i \in I} \{h_i\} \cup \mathsf{DSet}^{h_i}(v_i)\right) - \{h\}$

In the above conditions, I denotes the index set of the parents of the node d. The above condition requires that all the parents of d have hypothetical states. This set of hypothetical states of the parent nodes is AND-causal to the current physical state of d. The second condition requires this set of hypothetical states of the parent nodes contains a hypothetical state with respect to h. The third condition states that consistency is conditional on the set of all hypotheses that the hypothetical state of the parents of d depends on.

Conditional consistency for OR-type alarms can be defined similarly. Let d be an OR-type monitored discrepancy that is triggered at time index k. Then the signalling discrepancy d is said to be *conditionally consistent* with the hypothesis h given the set of hypothesis $H' \subset H_k$ if

$$(\exists v' \in \mathsf{Parents}(d)) \;\; (\mathsf{HState}_k^h(v'), \mathsf{PState}_k(d)) \in \mathsf{OC}_k \text{ and } \mathsf{DSet}_k^h(v') = H'$$

That is, one of the parents of d, v' has a hypothetical state with respect to h that is OR-causal to the current physical state of d and in addition the hypothetical state of v' is conditionally dependent on H'.

When the node d is explained by h_f conditional on the set of hypothesis H' the correspond state and time maps is then updated by setting $\mathsf{HState}_k^h(d) = \mathtt{ON}$, $\mathsf{HTime}_k^h(d) = [t, t]$, and $\mathsf{DSet}_k^h(d) = H'$. Consequently, the monitored discrepancy d is added to the set CS of conditionally supporting alarms of h_f. However, the rank of h will not be incremented in this case. The rank will only be incremented if h is provided in conjunction with the set H'.

In the above setting, (absolute) consistencies takes precedence over conditional ones. That is, if an alarm can be explained based on the occurrence of a single failure mode f, then any other explanation that requires the occurrence of several failure modes including f will not be considered. However, explanations the requires the occurrence of several failure modes non of which can be used as a single explanation of the alarm will be considered. This setting is a direct consequence of the parsimony principle which suggests the preference of the simplest explanation when possible.

For every event $e_k = (d, t)$, $d \in D_a$, the reasoner will try to explain e_k based on the current available hypothesis. If the event cannot be explained based on any of the current hypothesis, a new set of hypothesis will generated to explain d. In the case of OR-type discrepancies the reasoner will generate a hypothesis for each failure mode directly connected to d. Formally, let f be a failure mode directly connected to the OR-type discrepancy d. Then a new hypothesis for f is created if all the following holds.

- $\mathsf{EState}_k((f, d)) = \mathtt{ON}$,

- $t_{min}((f,d)) \leq \big(\mathsf{Time}(k) - \mathsf{ETime}((f,d))\big) \leq t_{max}((f,d))$

In the new hypothesis h_f the hypothetical state of f, $\mathsf{HState}_k^{h_f}(f)$, is set to `ON` and the corresponding interval is set to $\mathsf{HTime}_k^{h}(d) = [t_1, t_2]$, where

$$t_1 = \mathsf{Time}(k) - t_{min}(f,d), \quad \text{and} \quad t_2 = \min(\mathsf{ETime}((f,d)), \mathsf{Time}(k) - t_{max}(f,d))$$

Note that condition for generating h_f ensures that $t_1 \leq t_2$. Also, in the new hypothesis h_f, the correspond state and time maps is then updated by setting $\mathsf{HState}_k^{h_f}(d) = \mathtt{ON}$, and $\mathsf{HTime}_k^{h_f}(d) = [t,t]$. The monitored discrepancy d is added to the set of primary supporting alarms of h_f and the rank of h_f will be set to 1.

The case when d is an AND-type discrepancy is treated similarly. In this case a new hypothesis for each failure mode f directly connected to d is generated similar to the OR-type case. The hypothetical state and interval of f is also set in the same way. However, the hypothesis generated in this case are temporary hypothesis. Once all possible new hypothesis are generated, the reasoner will try to explain the alarm at d using the conditional consistency definition for AND-nodes as discussed earlier in this section. If the node cannot be explained using the new generated hypothesis. Then this set of temporary hypothesis will be deleted, and the node will be declared inexplainable.

Finally, if the new event (alarm) (d,t) cannot be explained by either consistency relationship with existing hypothesis or by generating new hypothesis, then the alarm is declared an absolute false alarm. In this case, the type of corresponding discrepancy will be changed to un-monitored discrepancy, that is, $\mathsf{DS}(d)$ will be set to `OFF` and any physical event correspond to the discrepancy d will be ignored. Algorithm 1 below shows the procedure for updating hypotheses for alarm events.

7.3 Evaluating Hypothesis and Generating Failure Report

In this stage, the current set of hypotheses are examined for possible conflicts due to common paths and nodes in the HFPG model. The ranks of consistent hypotheses sets are updated counting into effect mutual dependencies. The hypotheses set with the highest ranking is used to generate the failure report which contains an estimation of the current failure modes, their time of occurrence, and any possible sensor failures.

7.4 Complexity Analysis

The total number of nodes in the HFPG model is $(n+m)$ where n is the number of failure modes and n is the number of discrepancies. The graph is implemented as an adjacency matrix, and therefore both BFS and DFS searching algorithms are $O(n+m)^2$. The worst case number of hypotheses is $O(nm)$. However, the number of hypotheses in typical practical situations is more likely to be within $O(n)$. Updating the hypotheses set is done by updating the consistency relation between nodes in the graph which is by search the graph recursively until the set reach a settling point (for the given hypothesis). This part is of polynomial complexity on the size of the graph and the current number of hypothesis. Resolving the conflict between hypothesis is done by generating all possible combinations of hypothesis and therefore is of exponential complexity with respect to the number of hypothesis.

```
// This procedure is called every time a new alarm is activated.
// It terminates by explaining new alarms w.r.t the current hypotheses
// or by generating a new set of hypothesis or by declaring the alarm false
Input: $e_k = (d, t)$
Explained := false
for all $h$ in $H_k$ do
  if $\mathsf{Consis}_k(h, d)$ then
    Explained := true
    $h.SS$.add($d$)
    $\mathsf{Rank}(h) := \mathsf{Rank}(h) + 1$
    $\mathsf{HState}_k^h(d) :=$ ON; $\mathsf{HTime}_k^h(d) := [t, t]$
  else if $\mathsf{DConsis}_k(h, d, H')$ then
    Explained := true
    $h.CS$.add($d$)
    $\mathsf{HState}_k^h(d) :=$ ON; $\mathsf{HTime}_k^h(d) := [t, t]$; $\mathsf{DSet}_k^h(d) := H'$
  end if
end for
if not Explained then
  for all $f$ in $\mathsf{Parents}(d) \cap F$ do
    if $\mathsf{EState}_k(f, d) =$ ON and $t_{min}((f, d)) \leq \big(t - \mathsf{ETime}((f, d))\big) \leq t_{max}((f, d))$ then
      if $\mathsf{DC}(d) :=$ OR then
        $h_f = H_k$.AddNewHypothesis($f$)
        Explained := true
      else
        $h_f = H_k$.AddNewTempHypothesis($f$)
      end if
      $\mathsf{HState}_k^{h_f}(f) :=$ ON; $\mathsf{Rank}(h_f) := 1$
      $\mathsf{HTime}_k^{h_f}(f) := [(\mathsf{Time}(k) - t_{min}(f, d)), \min(\mathsf{ETime}((f, d)), \mathsf{Time}(k) - t_{max}(f, d))]$
      $\mathsf{HState}_k^{h_f}(d) :=$ ON; $\mathsf{HTime}_k^{h_f}(d) := [t, t]$
    end if
  end for
  if $\mathsf{DC}(d) =$ AND then
    for all $h$ in $H_k$ do
      if $\mathsf{DConsis}_k(h, d, H')$ then
        Explained := true
        $h.CS$.add($d$)
        $\mathsf{HState}_k^h(d) :=$ ON; $\mathsf{HTime}_k^h(d) := [t, t]$; $\mathsf{DSet}_k^h(d) := H'$
      end if
    end for
  end if
end if
if not Explained then
  $H_k$.RemoveTempHypotheses
  FalseAlarms.add($d$)
  $\mathsf{DS}(d) =$ OFF
end if
```

Algorithm 1: The Update Hypothesis algorithm

8 Diagnoser Implementation

Given a HFPG representing the failure propagation in the system and a sequence of sensors signals corresponding to monitored discrepancies and possible mode switches, the diagnosis problem is to generate a failure report which consist of a set of hypothesis that explains all the current signaling discrepancies. Figure 8 shows a simplified UML diagram of the basic elements of the HFPG diagnosis system and the relation between them.

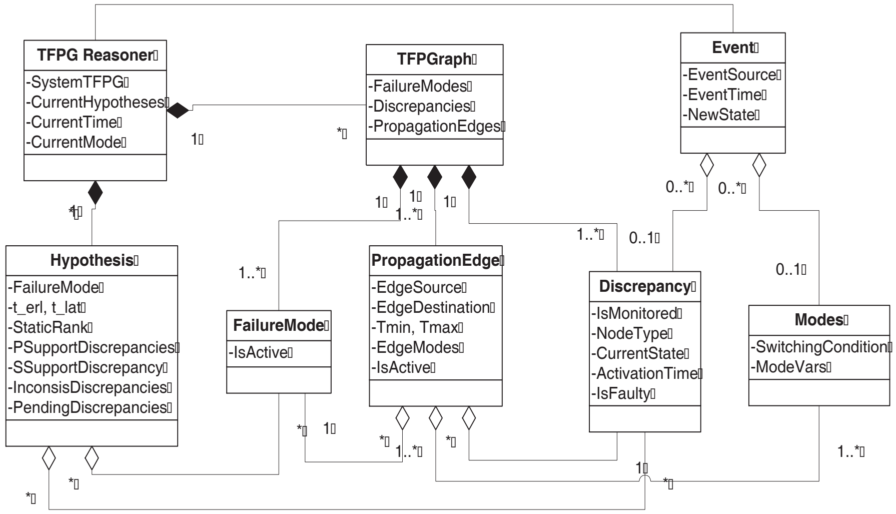
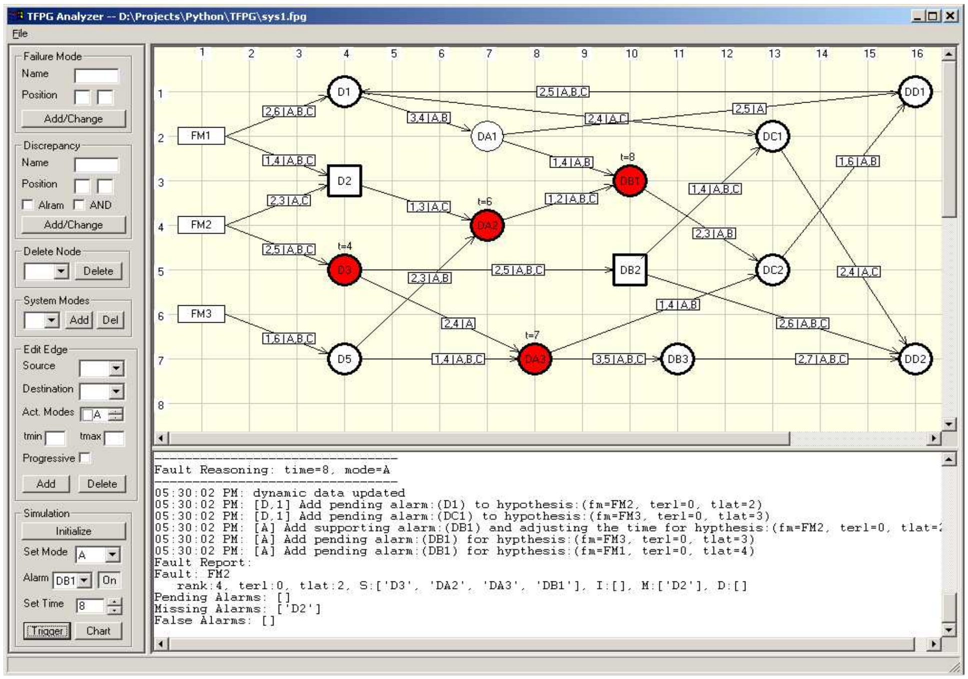


Figure 8: A simplified diagram of the HFPG diagnosis system

We have developed and tested a real-time diagnosis tool based on the hybrid failure propagation graph. The HFPG diagnosis tool is shown above in Figure 9. The reasoner engine in the HFPG tool is based on a robust incremental diagnostics algorithm described above. The tool can handle observation errors (sensor failure) and multiple fault scenarios. The tool also provides a simulation interface to test and evaluate the HFPG model for fault scenarios. The HFPG diagnosis algorithm is currently used as a part of the diagnostic module in a fault adaptive control structure aimed to support integrated fault diagnostics and control reconfiguration for large-scale and heterogeneous systems.

9 Conclusion

In this paper we introduced an approach for robust diagnosis of switching systems based on an extended version of the hybrid failure propagation graph model. The model can be used for diagnosis a general class of systems with mode switching conditions. We presented the main elements of the diagnostic system based on the hybrid failure propagation graph settings, and described the main parts of the diagnosis reasoning algorithm. In future work, we plan to enhance the efficiency of the diagnosis algorithm by incrementally identifying conflicting hypotheses at each time the set of hypotheses is updated.



Act. Modes A

Set Mode A

Alarm DB1 On

Set Time 8

```
--------------------------------
Fault Reasoning: time=8, mode=A
--------------------------------
05:30:02 PM: dynamic data updated
05:30:02 PM: [D,1] Add pending alarm:(D1) to hypothesis:(fm=FM2, terl=0, tlat=2)
05:30:02 PM: [D,1] Add pending alarm:(DC1) to hypothesis:(fm=FM3, terl=0, tlat=3)
05:30:02 PM: [A] Add supporting alarm:(DB1) and adjusting the time for hypthesis:(fm=FM2, terl=0, tlat=2
05:30:02 PM: [A] Add pending alarm:(DB1) for hypthesis:(fm=FM3, terl=0, tlat=3)
05:30:02 PM: [A] Add pending alarm:(DB1) for hypthesis:(fm=FM1, terl=0, tlat=4)
Fault Report:
Fault: FM2
   rank:4, terl:0, tlat:2, S:['D3', 'DA2', 'DA3', 'DB1'], I:[], M:['D2'], D:[]
Pending Alarms: []
Missing Alarms: ['D2']
False Alarms: []
```

Figure 9: The HFPG diagnostic tool

References

[1] Y. Ishida, N. Adachi, and H. Tokumaru. Topological approach to failure diagnosis of large-scale systems. *IEEE Trans. Syst., Man and Cybernetics*, 15(3):327–333, 1985.

[2] G. Karsai, J. Sztipanovits, S. Padalkar, and C. Biegl. Model based intelligent process control for cogenerator plants. *Journal of Parallel and Distributed Systems*, 15:90–103, 1992.

[3] A. Misra. *Sensor-Based Diagnosis of Dynamical Systems*. PhD thesis, Vanderbilt University, 1994.

[4] A. Misra, J. Sztipanovits, and J. Carnes. Robust diagnostics: Structural redundancy approach. In *SPIE's Symposium on Intelligent Systems*, 1994.

[5] S. Padalkar, J. Sztipanovits, G. Karsai, N. Miyasaka, and K. C. Okuda. Real-time fault diagnostics. *IEEE Expert*, 6(3):75–85, 1991.

[6] S. V. Nageswara Rao and N. Viswanadham. Fault diagnosis in dynamical systems: A graph theoretic approach. *Int. J. Systems Sci.*, 18(4):687–695, 1987.

[7] S. V. Nageswara Rao and N. Viswanadham. A methodology for knowledge acquisition and reasoning in failure analysis of systems. *IEEE Trans. Syst., Man and Cybernetics*, 17(2):274–288, 1987.