System Diagnosis using Hybrid Failure Propagation Graphs[1]

Sherif Abdelwahed and **Gabor Karsai** and **Gautam Biswas** [2]

Abstract. This paper presents an approach for robust diagnosis of a general class of dynamic systems based on a temporal failure propagation model. The proposed approach can be applied to a general class of systems with both time and event driven dynamics such as hybrid and discrete event systems. The paper presents the syntax and semantics of the proposed model and introduces the diagnosis approach.

1 Introduction

Diagnostic algorithms detect, isolate and estimate system failures using observed signals and measurements from the system sensors and actuators, and comparing them against a model that captures nominal and/or faulty behavior. The observed behavior is explained by a set of hypotheses that capture parameterized changes in the system components.

Two kinds of modeling paradigms have been commonly used to describe the behavior of engineering systems: *analytical models* and *associative models*. Analytical models such as differential equations, state machines, and hybrid automata are used to describe the nominal (correct) system behavior. The analytical approach, depends on the availability of an accurate mathematical model, which may be complex and difficult to obtain for practical real-life systems.

Associative models such as fault trees, cause-consequence diagrams, diagnosis dictionaries, and expert systems describe system behavior when faults are present [2, 7, 8, 1, 9]. The underlying fault models usually describe qualitatively the causal relationship between observed signals and failure sources. Associative modeling and diagnosis techniques are more common in practice due its simplicity and computational efficiency.

In this paper, we present a qualitative approach to failure diagnosis based on a temporal fault model referred to as hybrid failure propagation graph (HFPG). HFPG is an extension of the timed failure propagation graphs (TFPG) [4, 5]. The TFPG model is closely related to the fault model presented in [6, 3] and used for an integrated fault diagnoses and process control system. The HFPG model adds mode dependency constraints on the propagation links which can then be used to handle failure scenarios in hybrid and switching systems. This paper presents the formal description of the diagnosis problem and the main elements of the diagnostic system based on hybrid failure propagation graph models.

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

[2] Emails: {sherif,gabor,biswas}@isis.vanderbilt.edu. Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, 37203

2 Hybrid Failure Propagation Graphs

The hybrid failure propagation graph (HFPG) model 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. A discrepancy 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 capture propagation of failure effects over time in the dynamics system. To accommodate mode-switching in systems, edges in the HFPG model can be activated/deactivated based on the current operating mode of the system.

2.1 Syntax

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, where $V = F \cup D$
- M is a nonempty set of system modes. We assume that at each time instance t the system can be in only one mode
- $\mathsf{ET} : E \to Int$ where Int denotes finite time intervals
- $\mathsf{EM} : E \to \mathcal{P}(M)$, where $\mathsf{EM}(e) \neq \varnothing$ for any edge $e \in E$
- $\mathsf{DC} : D \to \{\texttt{AND}, \texttt{OR}\}$, defines the type of each discrepancy
- $\mathsf{DS} : D \to \{\texttt{True}, \texttt{False}\}$ is a map defining the monitoring status of the discrepancy as either `ON` for discrepancies attached to monitored alarms or `OFF` otherwise

Some of the discrepancies are monitored as defined by the map DS. The set of monitored discrepancies will be denoted D_a. An edge $e = (v, v') \in E$ if and only if a state change of v can propagate and participate in changing the state of v'. The map ET associates each edge $e \in E$ with the minimum and maximum time for the failure to propagate 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 (at 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 that HFPG models do not contain self loops and that failure modes cannot be a destination of any edge. Also every discrepancy must be caused by another discrepancy or a failure mode.

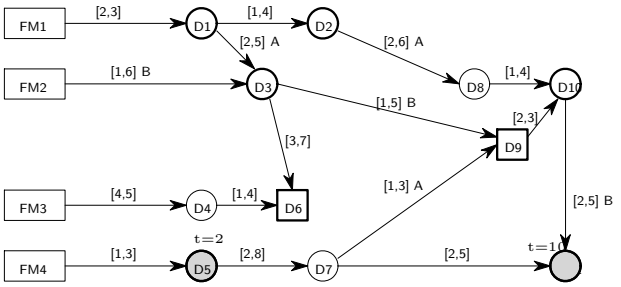

Figure 1. A hybrid failure propagation graph

Figure 1 shows an example of an HFPG model. In this figure, rectangles represent the failure modes while circles and squares represent OR and AND type discrepancies, respectively. Monitored discrepancies appears as bold circles or rectangles. Edges between nodes captures failure propagation in the system. Propagation through edges are parameterized by an interval, $[t_{min}, t_{max}]$, and the set of modes in which the edge is active. Sequence of alarms are identified by shaded discrepancies. The time at which the alarm is observed is shown above the corresponding discrepancy.

2.2 Semantics

Failure propagation between nodes depends primarily on: (i) the current mode of the system and the states of the observed discrepancies, and (ii) the past history of the system. Since we are dealing with hybrid systems, the past history includes both the monitored alarm status and the sequence of mode changes that occur in the system. Mode changes can activate and deactivate edges, therefore, to maintain the consistency in the reasoning process, one has to explicitly link the time of occurrence and the status of discrepancies with the time of mode changes. In this section, the semantics of failure propagation in HFPG models is defined based on the concept of system events and states.

System events

There are two types of system events: a state change event and a mode change event. System events are represented as a tuple $v = (x, t)$, where $x \in D \cup F \cup M$ and t is the time at which the signal is observed. We will write $\mathsf{Signal}(v)$ to identify the source of the event, and $\mathsf{T}(v)$ to identify the time at which the event occurs. Therefore, $v = (\mathsf{Signal}(v), \mathsf{T}(v))$. An event v is triggered whenever a node state is changed or the system switches to a new mode. We write v_k to identify the kth event. An event v is an external event if $\mathsf{Signal}(v) \in F \cup M$ otherwise it is an internal event. Clearly, external events are generated by the system environment and are unpredictable while internal events are consequences of external events. We will write I to denote the index set for the system events.

System states

A system state corresponds to the current state of all nodes and edges in the HFPG model. Formally, a state is a map $\mathsf{State} : I \rightarrow \{\texttt{ON}, \texttt{OFF}\}^{N+M}$, where N is the number of nodes and M is the number of edges in the HFPG model. Therefore, $\mathsf{State}(k)$ or simply State_k denotes the state of the system immediately after the occurrence of the kth event v_k. We will also write State_t where $t \in \mathbb{R}$ to denote the state at time t. From the definition, $\mathsf{State}(k)$ is a binary vector that defines the state of to each node and edge in the HFPG model at time k. For a node, a state ON indicates that the failure effect reached this node, otherwise it is set to OFF. For an edge, a state of ON indicates that the edge is currently active, otherwise it is set to OFF. For $z \in F \cup D \cup E$ we will write $\mathsf{State}_k(z)$ to indicate the state of z immediately after event v_k.

In the HFPG model, changes in the current node states depends on the last activation time of propagation edges. The map $\mathsf{ETime} : E \rightarrow \mathbb{R}$ where $\mathsf{ETime}(e)$ defines the time of the last state change of e. The map ETime is time dependent and therefore we may write ETime_k to denote the map immediately after the event v_k.

System dynamics

The system starts with an initial state, State_o in which all system nodes are initially OFF, and the initial mode defines the set of active edges. External events drive state changes for failure modes and propagation links. Such changes can happen at any time instant. The state change is defined by the incoming event, namely for all $i \in I$

- $\mathsf{Signal}(v_i) \in F \iff \mathsf{State}_i(\mathsf{Signal}(v_i)) = \texttt{ON} \wedge \mathsf{State}_{i-1}(\mathsf{Signal}(v_i)) = \texttt{OFF}$
- $\mathsf{Signal}(v_i) \in M \implies (\forall e \in E)\ \mathsf{State}_i(e) = \texttt{ON} \Leftrightarrow \mathsf{Signal}(v_i) \in \mathsf{EM}(e)$

That is, a failure mode event will change the state of a failure mode node to ON. Also, under the assumption of persistent failure, such signal can only be generated if the previous state of the failure mode is OFF.

Internal events are direct consequences of external events, particulary failure modes. The state changing event $v_i = (d, t)$ where $i \in I$, $d \in D$ and $\mathsf{DC}(d) = \texttt{OR}$ can occur only if all of the following holds,

- $\mathsf{State}_{i-1}(d) = \texttt{OFF} \wedge \mathsf{State}_i(r, d) = \texttt{ON}$,
- $(\exists (r, d) \in E)(\exists j < i)$ such that $\mathsf{State}_j(r) = \texttt{ON}$ and, $t_{min}(r, d) \leq \mathsf{T}(i) - \max(\mathsf{ETime}(r, d)), \mathsf{T}(j)) \leq t_{max}(r, d)$

That is a change of state of a discrepancy node d of type OR can only occur if a parent node r is currently ON and the edge (r, d) has been active long enough to allow the propagation of failure from r to d. Such change of state should occur definitely at the maximum time defined by the propagation link. Formally, $v_i = (d, t)$ where $i \in I$, $d \in D$ and $\mathsf{DC}(d) = \texttt{OR}$ should occur if there exists a parent of d, say r, and a time index $j \leq i$ such that

$$\mathsf{State}_j(r) = \texttt{ON} \wedge \mathsf{State}_t(r, d) = \texttt{ON} \wedge t - \max(\mathsf{ETime}(r, d)), \mathsf{T}(j)) = t_{max}(r, d)$$

In case of AND type discrepancy, d, all the parents of d need to be active before the activation of d becomes active. However, not all the failure effect need to reach d at the same time. We assume that once a failure effect from a parent reach the AND node it will remain in effect even if the connection between the two node is deactivated latter. Condition for the activation of AND type node can be formalized similar to the above conditions for OR type nodes.

During operation, the system is triggered by a sequence of events and state changes. Such sequence forms what is called an execution trajectory. The set of all possible execution trajectories defines the HFPG behavioral semantics.

3 The Diagnosis Problem

The HFPG diagnoser is a reactive module that is triggered by signals from the system sensors. The diagnoser generates a set of logically consistent hypotheses for the current state of the system, based on the observed sequence of alarms. The hypotheses are then ranked based on the number of supporting alarms versus the number of inconsistent ones. The set of hypotheses with the highest rank is then selected as the most plausible estimations of the current state of the system.

3.1 Failure Mode Hypotheses

A failure mode 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 $h_f = (f, t_e, t_l, r, SP, SS, IN, MI, PN)$, 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. The elements SP, SS, CS, IN, MI, and PN are sets of discrepancies defined as follows:

- $SP \subseteq D_a$ is the set of primary signalling discrepancies that support the hypothesis h_f. Alarms in SP are either triggered as an immediate consequence of f, or can only be explained based on the occurrence of f alone.
- $SS \subseteq D_a$ is the set of secondary signalling discrepancies that support the hypothesis h_f. Alarms in SS are triggered as a consequence of alarms already explained as a consequence of f and are supporting the hypothesis h_f.
- $CS \subseteq D_a$ is the set of secondary signalling discrepancies that support the hypothesis h given that other hypothesis are valid. Alarms in CS are either a consequence of a set of failure modes $F' \subseteq F$ where $f \in F'$ or a consequence of alarms already explained as a consequence of F'.
- $IN \subseteq D_a$ is the set of signalling monitored discrepancies that are inconsistent with the hypothesis h_f. These alarms are connected to f but cannot be explained based on h_f.
- $MI \subseteq D_a$ is the set of silent monitored discrepancies that are inconsistent with the hypothesis h_f. These inactive alarms are connected to the f but should be singling according to the hypothesis h_f.
- $PN \subseteq D_a$ 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.

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 logically valid hypothesis. 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.

3.2 Observable events

Observable events trigger the HFPG diagnoser to update the current set of hypotheses. There are two types of observable events; physical and hypothetical events. Physical events correspond to observed signals from the system sensors, while hypothetical events correspond to confirmed state inconsistencies according to a given hypothesis. Physical events can be either a signalling alarm or a mode switching event. We assume here that mode change observations are accurate. On the other hand, a sensor may fail and therefore the underlying event may not correspond to actual system event.

A physical event is represented by the tuple $v = (x, t)$, 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. The event can be written also as $v = (\mathsf{Signal}(v), \mathsf{T}(v))$. The diagnostic system keeps a record of the sequence of all timed events from the system initial start to the current time. We will also write v_k to identify the kth physical event.

Hypothetical events are referred to as time-out events. 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.

3.3 Physical and Hypothetical States

Based on the way the state of a given node is evaluated we distinguish here between two types of state estimations 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 these two types of states defines the overall consistency relationship between alarms and hypothesis.

Physical states are only be defined for monitored discrepancies. The physical state of a monitored discrepancy can be either active or inactive. The map $\mathsf{PState} : D_a \rightarrow \{\mathtt{ON}, \mathtt{OFF}\}$, assigns to each monitored alarm $d \in D_a$ its current measured states which can be ON if the alarm is active, otherwise it is OFF. We define another map $\mathsf{PTime} : D_a \rightarrow \mathbb{R}$ where $\mathsf{PTime}(d)$ is the time of the last change in the physical state of $d \in D_a$. These two maps are time dependent and, therefore, we may write PState_k and PTime_k to denote the maps after the kth event, v_k. Initially, the physical states of all alarms are set to OFF and the corresponding times are set to zero.

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

$\mathsf{HState}^{h_f} : V \rightarrow \{\texttt{ON}, \texttt{OFF}, \texttt{UDF}\}$, where UDF denotes undefined values. The map HState^{h_f} assigns to each node $v \in V$ its state according to h_f. The map HState^{h_f} is only defined for the set of nodes that can be reached from the failure mode f at any system mode including the node f. The hypothetical state of a node reachable from f can be ON if the node should be active according to the hypothesis h_f, otherwise it is OFF.

The map $\mathsf{HTime}^{h_f} : V \rightarrow Int \cup \{\texttt{UDF}\}$ assigns to each node in V the time interval in which the node must have been activated within according to the hypothesis h_f. Similar to HState^{h_f}, the map HTime^{h_f} is only defined for nodes that are reachable from f. For silent discrepancies the map $\mathsf{HTime}^{h_f}(d)$ shows the time interval at which the discrepancy d should be active according to h_f.

In general, a hypothetical state may be dependant on more than one hypothesis. Such situation is attributed to the fact that the state of an AND-type discrepancy depends on the combined states of all its parent nodes. Such dependency is modeled using the map $\mathsf{DSet}^{h_f} : V \rightarrow \mathcal{P}(H)$ 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 $\mathsf{DSet}^{h_f}(v)$.

4 The Diagnostic Reasoner

The diagnoser updates the set of most plausible valid hypotheses based on the causal and timing consistency between the discrepancies. Consistency between discrepancy nodes depends on their types and the current mode of the system.

4.1 Causality Relationship

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 and OR. Causality relationship is 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 an OR discrepancy and the hypothetical state of one of its parent nodes, while AND-causality is a relationship between the state of an AND node and the states of all its parents.

Causality is independent on the state type of the node, therefore, we it can be defined for a general state. 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 $\mathsf{Tmin}(v) = \mathsf{Tmax}(v) = \mathsf{PTime}(v)$. Let $v', v \in V$ be two nodes in the HFPG model such that, $DC(v) = \texttt{OR}$ and $(v', v) \in E$. 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.

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

The above simply 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 underlying propagation link as well as the time at which this link is enabled. OR-causality is illustrated in Figure 2.

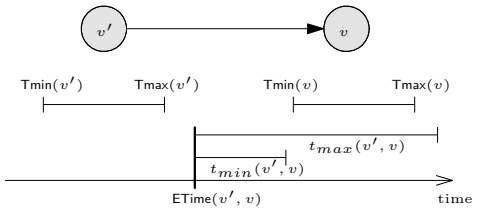

Figure 2. OR-Causality relationship

AND-causality can be defined similarly. Let $v \in V$ be a node in the HFPG model such that $\mathsf{DC}(v) = \texttt{AND}$. 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

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

That is, the $\mathsf{State}_k(V')$ (possibly) caused the current change of the $\mathsf{State}_k(v)$ to ON if state of every $v' \in V'$ is also 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 3.

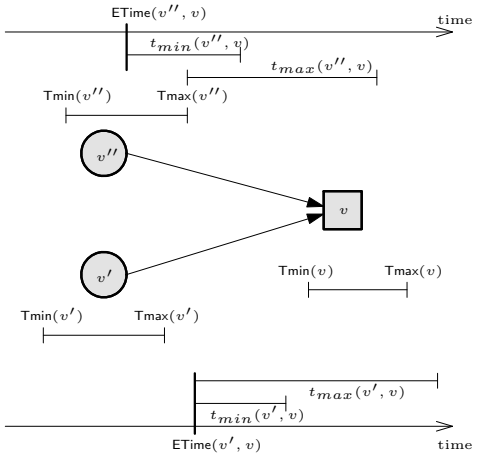

Figure 3. AND-Causality relationship

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. Intuitively, this condition ensures that the node v' will not be activated until it receives all the failure effects from its parents.

4.2 Temporal Consistency

Based on the observed alarms, the diagnose tries to estimate the new system state by updating the set of hypothesis. This update includes creating new hypothesis and/or reevaluating the current set of hypotheses. Based on the principle of parsimony, the diagnoser will create a new hypothesis only if current alarm cannot be explained using current hypotheses.

Consider the event $v_k = (d, t)$ indicating that a monitored discrepancy d has changed its state from `OFF` to `ON` at time $t = \mathsf{Time}(v_k)$. 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 h if $(\exists v' \in \mathsf{Parents}(d))$ such that

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

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 for `AND`-type nodes 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 parents of d have 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 of the parents of d must have a hypothetical state with respect to the same hypothesis h. In the 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 $\mathsf{HState}_k^h(d) = \mathsf{PState}_k(d) =$ `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 $\mathsf{Consis}_k \subseteq H_k \times V$. Therefore, $\mathsf{Consis}_k(h, v)$ is true when the node v is consistent with the hypothesis $hinH_k$.

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

- $(\forall d_j \in \mathsf{Parents}(d))(\exists h_j \in H') \quad (\{\mathsf{HState}_k^{h_j}(d_j) | j \in J\}, \mathsf{PState}_k(d)) \in \mathsf{AC}_k$
- $(\exists j \in J)\ \ h_j = h$
- $H' = \left(\bigcup_{j \in J} \{h_j\} \cup \mathsf{DSet}^{h_j}(d_j)\right) - \{h\}$

In the above conditions, J 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 is defined as follows. 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 $H' \subset H_k$ if $\exists v' \in \mathsf{Parents}(d)$ such that

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

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 $\mathsf{HState}_k^h(d) =$ `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'.

Based on the parsimony principle, 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 that requires the occurrence of several failure modes none of which can be used as a single explanation of the alarm will be considered.

For every event $v_k = (d, t)$, $d \in D_a$, the reasoner will try to explain v_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. A new hypothesis for f is created if all the following holds.

- $\mathsf{EState}_k((f, d)) =$ `ON`,
- $t_{min}((f, d)) \leq \left(\mathsf{T}(k) - \mathsf{ETime}((f, d))\right) \leq t_{max}((f, d))$

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

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

Note that condition for generating h_f ensures that $t_1 \leq t_2$. Also, in the new hypothesis h_f, the corresponding state and time maps is then updated by setting $\mathsf{HState}_k^{h_f}(d) = \texttt{ON}$, 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 alarm cannot be explained using the new hypotheses the corresponding sensor will be declared faulty.

Algorithm 1 The Update Hypothesis algorithm

```
Input: v_k = (d, t)
Explained := false
for all h in H_k do
  if Consis_k(h, d) then
    Explained := true
    h.SS.add(d); Rank(h) := Rank(h) + 1
    HState_k^h(d) := ON; HTime_k^h(d) := [t, t]
  else if DConsis_k(h, d, H') then
    Explained := true; h.CS.add(d)
    HState_k^h(d) := ON; HTime_k^h(d) := [t, t]; DSet_k^h(d) := H'
  end if
end for
if not Explained then
  for all f in Parents(d) ∩ F do
    if EState_k(f, d) = ON and t_min((f, d)) ≤ (t − ETime((f, d))) ≤ t_max((f, d)) then
      if DC(d) := OR then
        h_f = H_k.AddNewHypo(f); Explained := true
      else
        h_f = H_k.AddNewTempHypo(f)
      end if
      HState_k^{h_f}(f) := ON; Rank(h_f) := 1
      HTime_k^{h_f}(f) := [(Time(k) − t_min(f, d)), min(ETime((f, d)), Time(k) − t_max(f, d))]
      HState_k^{h_f}(d) := ON; HTime_k^{h_f}(d) := [t, t]
    end if
  end for
  if DC(d) = AND then
    for all h in H_k do
      if DConsis_k(h, d, H') then
        Explained := true; h.CS.add(d); DSet_k^h(d) := H'
        HState_k^h(d) := ON; HTime_k^h(d) := [t, t]
      end if
    end for
  end if
end if
if not Explained then
  H_k.RemoveTempHypotheses
  FalseAlarms.add(d); DS(d) = False
end if
```

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 a non-monitored discrepancy, that is, $\mathsf{DS}(d)$ will be set to False and any physical event correspond to the discrepancy d will be ignored. Algorithm 1 shows the main part of the updating hypotheses procedure.

4.3 Generating Failure Report

In this stage, current hypotheses are examined for possible conflicts due to common paths. 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.

4.4 Complexity Analysis

The HFPG model is implemented as an adjacency matrix, and therefore both BFS and DFS searching algorithms are $O(n+m)^2$ where n is the number of failure modes and n is the number of discrepancies. 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 done by searching 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 hypotheses.

REFERENCES

[1] R. Hessian, B. Salter, and E. Goodwin, 'Fault-tree analysis for system design, development, modification, and verification', *IEEE Transactions on Reliability*, **39**(1), 87–91, (1990).

[2] 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).

[3] 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).

[4] A. Misra, *Sensor-Based Diagnosis of Dynamical Systems*, Ph.D. dissertation, Vanderbilt University, 1994.

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

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

[7] 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).

[8] 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).

[9] J. Richman and K. R. Bowden, 'The modern fault dictionary', in *International Test Conference*, pp. 696–702, (1985).