Hierarchical Reasoning about Faults in Cyber-Physical Energy Systems using Temporal Causal Diagrams

Ajay D Chhokra[1], Nagabhushan Mahadevan[2], Abhishek Dubey[3], Saqib Hasan[4], Daniel Balasubramanian[5] and Gabor Karsai[6]

[1,2,3,4,5,6] *Institute for Software-Integrated Systems, Vanderbilt University, Nashville, TN 37212, USA*
ajay.d.chhokra@vanderbilt.edu

ABSTRACT

Cyber-physical systems are often equipped with specialized fault management systems that observe the state of the system, decide if there is an anomaly and then take automated actions to isolate faults. For example, in electrical networks relays and breaks isolate faults in order to arrest failure propagation and protect the healthy parts of the system. However, due to the limited situational awareness and hidden failures the protection devices themselves, through their operation (or mis-operation) may cause overloading and the disconnection of parts of an otherwise healthy system. Additionally, often there can be faults in the management system itself leading to situations where it is difficult to isolate failures. Our work presented in this paper is geared towards solution of this problem by describing the formalism of Temporal Causal Diagrams (TCD-s) that augment the failure models for the physical systems with discrete time models of protection elements, accounting for the complex interactions between the protection devices and the physical plants. We use the case study of the standard Western System Coordinating Council (WSCC) 9 bus system to describe four different fault scenarios and illustrate how our approach can help isolate these failures. Though, we use power networks as exemplars in this paper our approach can be applied to other distributed cyber-physical systems, for example water networks.

1. INTRODUCTION

Recent advances in sensor networks, embedded systems, information and communication technology have steered the interest of scientific community towards the development of cyber physical systems (CPSs). A cyber physical system is the integration of physical processes with computation. Tight coupling between physical processes and software is the hallmark of such systems. These ubiquitous engineered systems form the backbone of control infrastructures in modern society. The focus of CPSs is to improve the collaborative link between physical and computational elements for enhancing autonomy and intelligence of the physical systems to be able to plan and modify their actions for evolving environments based on the self-awareness.

According to (Reppa et al., 2015b), the key concerns in designing CPSs are safety, reliability and fault tolerance. In order to address these concerns, the cyber ecosystem of many critical systems such as power systems is empowered with fault management components for arresting failure propagation. These specialized devises have supervision capabilities for diagnosing faults in the physical system and taking appropriate remedial actions for removing faulty components as mentioned in (Blanke et al., 2006; Isermann, 2006). Figure 1 shows a network of interconnected CPSs. The cyber system of each CPS includes a specialized fault management component. A fault management component consists of an anomaly detector and a reconfiguration controller. Anomaly detector detects discrepancies, as a result of a fault in physical plant, from the sensor data and informs the reconfiguration controller about the observed anomaly. The reconfiguration controller instructs the actuator to change its state leading to modification of the operating conditions that can arrest failure effect in the physical system. Various quantitative and qualitative approaches have been developed over the years to diagnose faults in physical plant, sensors and actuators, see (Bouamama et al., 2014) for details. In this paper, we limit the scope of cyber system to fault managers (anomaly detectors and reconfiguration controllers) and communication amongst these computation elements only.

Apart from sensors and actuators, cyber fault management components such as anomaly detectors and reconfiguration controllers can have faults too. Anomalous behavior can cause inadvertent changes in the physical system that can lead to secondary failures. Moreover, in critical systems, the fault management components are based on reflex healing approaches and have to act on local information in a limited amount of time. These actions are devoid of system wide perspective and can cause cascading failures. A similar phenomenon was seen in the blackouts of 2003 in the USA,

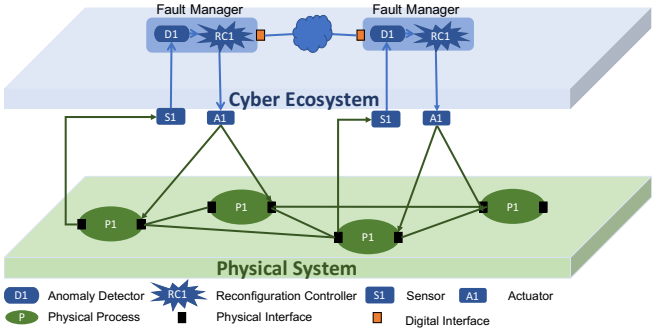

Figure 1. An interconnected system consisting of a network of physical processes, sensors, actuators and fault managers

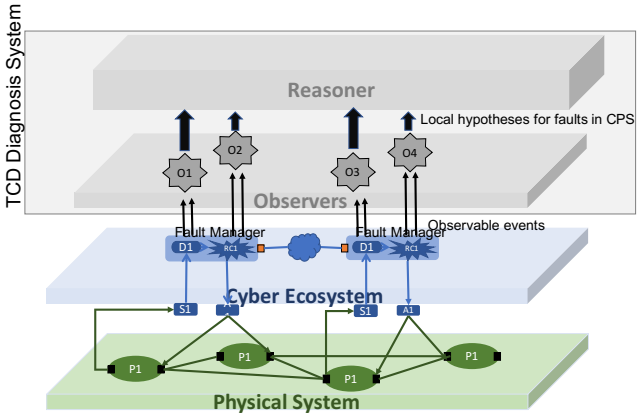

Figure 2. TCD Diagnosis System

where misoperations of protection devices exacerbated the initial disturbances into cascading outages in the other parts of electric grid (North American Electric Reliability Corporation, 2012).

System of Interest: One of the emerging applications of CPS is the modern power system or referred to as **Cyber Physical Energy System** (CPES). *CPES* is the amalgamation of power grid technology with intelligent control, co-ordination and communication between demand and supply side to deliver electricity efficiently. Physical components in power systems such as transmission lines, generators and transformers etc work in dynamic environments resulting from varying load, changing operational requirements and component degradation. To achieve fault tolerance and required level of resiliency, a number of fast acting localized protection mechanisms are used to detect and isolate faults. These protection systems include detection devices such as fast-acting numerical relays that are designed to detect abnormal changes in physical properties (current, voltage, impedance) and actuation devices such as breakers that can be triggered to open the circuit in electrical networks. While these protection devices are effective in detecting and isolating faults in specific regions of a system their decisions are based on local information. This results in a highly conservative reaction from protection devices without considering the consequences of the control actions. Apart from lack of system-wide perspective, these protection devices have faults also. The change in system state due to (mis)operation of the protection devices can eventually increase stress on other parts of the system and thus cause secondary failures. These failures result in the triggering of other protection devices. This domino effect can quickly cascade to the whole system, ultimately leading to complete system shutdown.

Traditional data and model based failure diagnosis approaches, listed in (Sekine et al., 1992), do not fully capture the failure propagation in physical and cyber systems as a result of the interactions between the faults and their effects in the two systems. A new modeling and diagnosis strategy is needed that isolates the faults in physical and cyber components and is robust to the changes in the underlying physical system, cyber fault management system, sensors and actuators.

Our Contributions: In this paper we are presenting a diagnosis approach based on Temporal Causal Diagrams (TCDs) by considering 1) Discrete and continuous dynamics of the underlying components 2) Faults in the physical components 3) Misoperations or malfunctions of the discrete components (sensors, anomaly detectors, actuators, controllers) 4) Propagation of failure effects in both cyber and physical components. A TCD model includes a fault propagation graph as well as the behavioral model of protection devices under nominal and faulty conditions. It is an extension of our prior work on Temporal Failure Propagation Graphs (TFPG) Abdelwahed & Karsai (2007); Dubey et al. (2011).

We present a TCD model based diagnosis scheme which uses local observers and a system level reasoning engine to diagnose faults. The observers are discrete state machines derived from behavior models captured in the TCD model. They use the incoming observable events in real-time to produce *alarms* that are then consumed by the reasoning engine that produces system level *hypotheses* consistent with failure propagation graphs, to identify fault sources and predict impending system level effects, see Figure 1. A key feature of this technology lies in its ability to model and diagnose not only faults in physical system but also protection element or controller failures, where the controllers are tightly coupled with the physical components.

We also describe in this paper, a set of discrete-time behavioral models of widely used power system protection devices. In the end we demonstrate the proposed TCD reasoning technique for single and multi-fault scenarios using a standard Western System Coordinating Council (WSCC) 9 bus system.

Outline: The rest of the paper is organized as follows: Section 2 provides a survey of some of the published works on fault diagnosis for electrical power grids and section 3 highlights the key aspects of our approach in light of lessons learned from past work. Section 4 describes the TCD modeling formalism in detail. Section 5 gives insight into various physical and cyber elements associated with power transmission system and also describes their respective TCD models with the help of simple example. The failure diagnosis approach including observers and reasoning logics are described in section 6. Section 7 presents the case study with diagnosis results for different scenarios. Concluding remarks are provided in section 8.

2. RELATED RESEARCH

Fault diagnosis in cyber physical systems is a challenging tasks due to inherent heterogeneity and large scale of the physical systems. A number of decentralized and distributed schemes for fault detection are proposed in the literature. To enhance fault isolation, hierarchical or multiple levels of diagnosis has also been proposed. The single level of diagnosis is realized by local diagnosers. Specifically, the local diagnosers may exchange estimations (Khalili & Zhang, 2014; Yan & Edwards, 2008; Daigle et al., 2007), or measurements (Ferrari et al., 2012; Boem et al., 2013; Shames et al., 2011) of the interconnected system states or fault signatures (Daigle et al., 2007). Apart from faults in physical systems, a number of approaches have been proposed to diagnose faults in sensors and actuators (Reppa et al., 2015b; Zhang & Zhang, 2013a,b; Reppa et al., 2013, 2015a). In (Zhang & Zhang, 2013a,b), a distributed architecture is designed to isolate single faults while (Reppa et al., 2013, 2015a,b) can detect and isolate multiple sensor faults. However, a little attention has been given on diagnosing the behavior of anomaly detectors and reconfiguration controllers. In order to correctly isolate faults in interconnected systems, a holistic approach is required that covers components in physical and cyber systems.

Apart from the distributed and multilevel diagnosis discussed above, there exist a vast literature on the methodologies fine tuned for power systems. Fault diagnosis in power systems is an active area of research see (Ferreira et al., 2016) for details. Many technical papers have focused on fault segment estimation. The diagnosis approach can be broadly classified into three categories based on their underlying technique: **expert system** (Yongli et al., 1994; Huang, 2002; Cardoso et al., 2008; Jung et al., 2001), **artificial neural network** (Cardoso et al., 2004; Mahanty & Gupta, 2004; Thukaram et al., 2005; Bi et al., 2002) and **analytical model optimization** (Wu et al., 2005; Wen & Chang, 1997; He et al., 2009; Guo et al., 2010). In addition, approaches based on **petri networks** (Sun et al., 2004) and **cause-effect bayesian networks** (Chen et al., 2001, 2011; Guo et al., 2009; Chen, 2012; Yongli et al., 2006) have also been proposed.

Expert Systems are one of the earliest techniques to solve the failure diagnosis problem in Power Systems. The diagnosis process in an expert system can be rule based or model based. A comprehensive survey of such knowledge based approaches is available in (Sekine et al., 1992). The expert systems in general suffer from a number of drawbacks related to the maintenance of the knowledge database and slow response time. These approaches are expected to work well if all the received alarms are correct. Missing and incorrect alarms force the diagnosis technique to produce wrong hypotheses.

Artificial neural networks (ANNs) are adaptive systems inspired by biological systems. ANNs model the complex relationships between inputs and outputs without the explicit description of rules to precisely define the power system protection schemes i.e. based on operational data. Multilayer feedforward perceptron with backward propagation is the most commonly used neural network model (MPNN) for failure diagnosis as described in (Cardoso et al., 2004). However, this learning methodology suffers from slow training and low capability of inference with limited training data. In (Bi et al., 2002; Mahanty & Gupta, 2004), a neural networks with radial basis function (RBF) are presented. Authors in (Thukaram et al., 2005) discuss support vector machine (SVM) in order to avoid the shortcomings of MPNN. The artificial neural networks based approaches in general suffer from convergence problems. Further, the ANNs have to be retrained whenever there is a change in network topology as the weights are dependent upon the structure of the power system.

A number of model based analytical methods have been devised over the years for diagnosing failures in power systems, see (Wu et al., 2005; Wen & Chang, 1997; He et al., 2009) for details. Optimization techniques such as genetic algorithm (Wen & Chang, 1997), particle swarm optimization (He et al., 2009) and evolution algorithm (Wu et al., 2005), have been used to generate optimal failure hypotheses that best explain all the events/ alarms. The analytical model presented in (Guo et al., 2010) not only estimates the faults in the physical component but also hypothesizes the state of protections relays and circuit breakers. But these techniques rely heavily on critical and computationally expensive tasks such as the selection of an objective function, development of exact mathematical models for system actions and protective schemes, which greatly influence the accuracy of the failure diagnosis.

Cause effect networks have also been used to diagnose faults in power systems, as mentioned in (Chen et al., 2001, 2011; Guo et al., 2009; Chen, 2012; Yongli et al., 2006). A cause effect network consists of nodes and edges where nodes represent failures and relaying system actions. Edges imply the causal relationship between faults and relay actions. The accuracy of the diagnosis approach presented in (Chen et al., 2001, 2011) decreases if there is uncertainty in the behavior

of protection relays (PR) and/or circuit breakers (CB). Authors in (Chen, 2012; Yongli et al., 2006) consider the anomalous behavior of PR and CB by extending the cause effect approach with fuzzy digraphs and Bayesian networks. However these techniques do not provide hypotheses related to the state of PRs and CBs. An on-line alarm analysis approach is presented in (Guo et al., 2009) for diagnosing failure modes in the physical plant as well as in a relaying system based on a temporal causal network. But this approach does not take into account the operating modes and conditions of the system that influence the failure propagation.

The approach described in this paper differs from the current methodologies where fault analysis and mitigation relies on a logic-based approach that depends on hard thresholds and local information assisted by manual system level analysis. The causal model presented in this paper is based on the timed failure propagation graph (TFPG) introduced in (Abdelwahed & Karsai, 2006; Padalkar et al., 1991; Abdelwahed & Karsai, 2007), which is conceptually related to the temporal causal network approach presented in (Guo et al., 2009). We have extended this work to take into account local protection action in a subsystem which could arrest the fault or lead to larger cascading faults. This is primarily done by considering the discrete behavior of the protection devices and incorporating their effects in fault propagation. Our approach can improve the effectiveness of isolating failures in large-scale systems such as Smart Electric Grids, by identifying impending failure propagation which increases the system reliability and reduces the losses accrued due to power failures.

3. TIMED FAILURE PROPAGATION GRAPHS AND THEIR LIMITATIONS

In the past, we have used the Timed Failure Propagation Graph (TFPG) based models and reasoning schemes to diagnose faults in physical systems (Abdelwahed & Karsai, 2006) and software systems (Dubey et al., 2011). A temporal failure propagation graph is a labeled directed graph where nodes are either failure modes or discrepancies. Discrepancies are the failure effects, some of which may be observable. Edges in TFPG represent the causality of the failure propagation and edge labels capture operating modes in which the failure effect can propagate over the edge, as well as a time-interval by which the failure effect could be delayed.

Figure 3 shows a simple failure graph with two failure mode nodes *FM1* and *FM2* with 3 observable discrepancies *D1*, *D2*, *D5* and 2 silent discrepancies *D3*, *D4*. Alarms *A1*, *A2* and *A3* signal the detection of monitored discrepancies. The failure effect of *FM1* reaches *D1* then propagates to *D3* and finally reaches *D5* under operating conditions quantified by modes *a* and *d*. The TFPG reasoner accounts for fault propagation constraints imposed by the operational mode and temporal delays to produce multi-fault hypotheses that are able to consistently explain the observed alarms. For instance, the observation of alarm *A1* at time $t = t_1$ triggers the TFPG reasoner to produce a hypotheses, stating the failure mode FM1 was activated during the interval, $[t_1 - 2, t_1 - 1]$, if the current system mode is either *a* or *d*. Also, the reasoning engine is robust to alarm faults (false positives or false negatives) which are taken into account while computing the metrics that are used to rank the hypotheses (Abdelwahed & Karsai, 2007). For example, if the current system mode is either *b* or *d* and alarm *D5* is observed, then TFPG reasoner will produce two hypothesis. One indicating the presence of fault *FM2* along with missing alarm *A2* and a second is related to false alarm *A3*. The TFPG based diagnosis scheme has been successfully applied to physical systems including industrial plants (Padalkar et al., 1991) and aerospace systems (Mahadevan et al., 2011).

However in certain cyber physical systems such as transmission and distribution networks (e.g. power and water) there are protection devices that try to arrest the failure effect if detected. These protection devices alter the system topology by instructing breakers (switches) to change their state. These devices can also have faults that alter their response to the effect of the failure and control commands.

Figure 3 also depicts the abstract models of an anomaly detector, a protection device and two actuators that conjointly try to stop the effects of failures in the physical system discussed in the previous paragraph.

- **Anomaly Detector** consists of two states $\{S1, S2\}$. The detector generates alarms $\{A1, A2, A3\}$ in response to unobservable events $\{E1, E2, E3\}$, where $\{E1, E2, E3\}$ represent the failure effects modeled by discrepancies $\{D1, D2, D5\}$. The anomaly detector may have a failure mode of its own that will cause the detector to miss the failure effect. The activation of this fault is modeled by an unobservable event $F1$ that pushes the automaton from state *S1* to *S2*.
- **Protection Device** also consists of 2 states $\{S1, S2\}$. While in state *S1*, the protection device appropriately responds to the alarms generated by anomaly detector. The protection device emits commanding events $\{C1, C2\}$ for alarms $\{A1, A2\}$. Similar to the anomaly detector, the protection device also has a missed detection failure mode. The activation of failure mode is represented by the event $F2$.
- **Actuator** consist of 3 states $\{S1, S2, S3\}$. In response to the command *C1* by the protection device the actuator changes its state from *S1* to *S2*. This device also has missed detection fault that forces the breaker to ignore the commands sent by the protection devices. As shown in Figure 3, the system consists of two breakers and the states of the breakers are mapped to system modes as $\{$*(S1, S1) = d; (S1, S2) = a; (S2, S1) = b; (S2, S2) =*

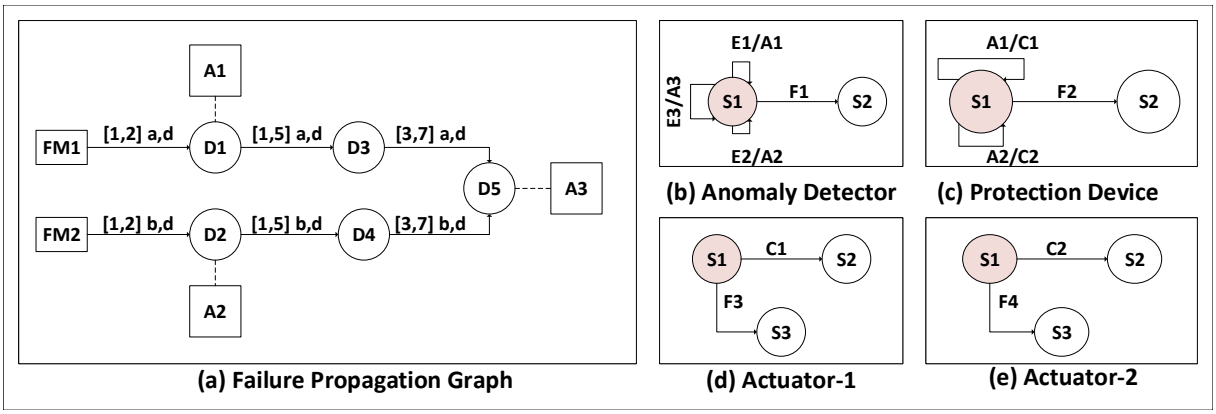

Figure 3. A sample temporal failure graph along with behavior automatons of different cyber components in both faulty and nominal modes

$c\}$, where the first element of the tuple represents state of *Actuator-1* and second implies *Actuator-2*.

One of the valid traces of the system shown in Figure 3 can be explained as follows: Fault *FM1* is injected and after 1.5 secs anomaly detector issued alarm *A1*. The alarm *A1* forces the protection device to emit command *C1* which forces the actuator to change state from *S1* to *S2*. The state change modifies the system mode from *d* to *b*. The mode change takes place within $1.5 + \delta_1 + \delta_2$, where δ_1 and δ_2 are maximum communication delays between the anomaly detector and the protection device, and the protection device and the actuator, respectively.

It can be observed that TFPG based approach could correctly isolate the fault-source *FM1*. However, its difficult to diagnose faults in the cyber infrastructure that includes protection devices along with anomaly detectors and actuators i.e *F1, F2, F3*. This is extremely desirable for cyber-physical systems where realistic assessment of fault propagation is not possible without accounting for the behavior of the deployed sensing and actuation components.

A more comprehensive approach is desired where the behavioral aspects (including faulty behavior) of local protection elements including anomaly detectors and actuator components can be modeled and tracked in conjunction with the fault propagation graph. It is with this objective, that we introduced the Temporal Causal Diagrams (TCDs) based diagnosis scheme in (Mahadevan et al., 2014) which incorporates the TFPG model and takes into account the problems associated with sensing and actuation elements.

Our initial approach using TCD relies on modifying the TFPG model to account for nominal and faulty operation of the cyber components by appending failure graphs with behavior models forming Temporal Causal Graphs. This quickly complicates a simple TFPG model as it introduces all the variants from the behavior model into the failure propagation graph, posing challenges when applying the strategy to large-scale examples of power grids.

Our current approach, as presented in this paper is a refinement of our earlier work using Temporal Causal Diagrams on electrical power grids which is more modular in nature. The refined approach uses a two layer hierarchical reasoning engine, where the lower layer includes observers derived from the behavioral models of the protection equipment. The observers reason about the events observed from their respective components and feed their inference to the higher level TCD reasoner. The TCD reasoner not only handles the fault propagation model (like the TFPG diagnosis engine), but also deals with the derived alarms (or hypotheses) reported by the observer(s). The reasoner uses the fault propagation model to reason about the derived alarms (hypotheses) fed by the observer(s) and computes consistent system level hypotheses. Figure 1 shows the diagnosis system block diagram consisting of multiple observers and the TCD reasoning engine. The hierarchical diagnosis system is supplied with events from the cyber-physical system. A key aspect of this approach is that the reasoner implementation is not affected by any change in the system topology or the behavior of the protection devices. The next section formally explains the structure of a Temporal Causal Diagram.

4. TEMPORAL CAUSAL DIAGRAM

A temporal causal diagram is a behavior-augmented failure propagation graph. It comprises of a directed graph that captures the failure propagation across the whole system in different operating conditions. It is influenced by the behavioral models of various cyber components (i.e. the protection equipment). The following subsections describe the modeling formalism for capturing the failure dynamics and the model of computation used for representing cyber components.

4.1. Temporal Failure Propagation Graphs

A temporal failure propagation graph is a labeled directed graph. In the context of self-correcting cyber physical systems such as power grids, the system mode or operating conditions depends upon the state of sources, sinks and the topology of the system. Identification of all operating conditions, i.e unique system modes is computationally very expensive. In this paper, we use the system topology dictated by the state of the actuators to map an operating condition (i.e. mode) to the failure propagation. However, while such a constraint imposed due to topology of the system is deemed necessary to identify when a fault will not propagate, it is not sufficient to state that the failures will propagate. So we need to extend the TFPG language with an additional map that associates uncertainty to failure edges. Formally, the extended TFPG is represented as a tuple $\{F, D, E, M, ET, EM, ND\}$, where

- F is a nonempty set of failure nodes. A failure node can be in two states either present denoted by ON state or absent represented by OFF state.
- D is a nonempty set of discrepancy nodes.
- $E \subseteq V \times V$ is a set of edges connecting the set of all nodes $V = F \cup D$.
- M is a nonempty set of system modes. At each time instance t the system can be in only one mode.
- $ET : E \rightarrow I$ is a map that associates every edge in E a time interval $[t_{min}, t_{max}] \in I$ that represents the minimum and maximum time for failure propagation over the edge.
- $EM : E \rightarrow \mathcal{P}(M)$ is a map that associates every edge in E with a set of modes in M when the edge is active. For any edge $e \in E$ that is not mode-dependent (i.e. active in all modes), $EM(e) = \varnothing$.
- $ND : E \rightarrow \{True, False\}$ is a map that associates an edge, $e \in E$ to *True* or *False*, where *True* implies the propagation along the edge, e **Will** happen, whereas *False* implies the propagation is uncertain and **Can** happen.

4.2. Discrete Behavior Models

The behavior of discrete devices is modeled using extended time triggered automaton (Krčál et al., 2004).The extension includes sets of failure modes and failure mode guards. Mathematically, an extended time triggered automaton is represented as tuple $(\Sigma, Q, q_0, Q_m, F_{cyber}, D_{cyber}, \mathbb{M}, \alpha(F), \Phi, T)$.

- **Event Set:** Σ is a finite set of events that consists of observable and unobservable events partitioned as $\Sigma = \Sigma_{obs} \cup \Sigma_{unobs}$ such that $\Sigma_{obs} \cap \Sigma_{unobs} = \phi$. Observable events are alarms, commands and messages exchanged between discrete components.Whereas, unobservable events are related to introduction of faults in system components.
- **Locations:** Q is a finite set of locations. $q_0 \in Q$ is the initial location of the automaton and $Q_m \subset Q$ is a finite set of marked locations.
- **Discrepancy Set:** D_{cyber} is a finite set of discrepancies associated with the component behavior, partitioned into observable and unobservable.
- **Failure Mode Set:** F_{cyber} is a finite set of unobservable failure modes associated with the component. Similar to a failure node in TFPG, failure mode also has ON and OFF states. δ_t is a function defined over $F_{cyber} \times \mathbb{R}_+$ that maps a failure mode $f \in F_{cyber}$ at time $t \in \mathbb{R}_+$ to *True* if the state of failure mode is ON and to *False* if the state is OFF.
- **Failure Mode Constraints:** $\alpha(F_{cyber})$ represents the set of all constraints defined over members of set F_{cyber}. An individual failure mode constraint, $\omega_t \in \alpha(F_{cyber})$, is a Boolean expression defined inductively as

$$\omega_t := \delta_t(f) \quad | \quad \neg\delta_t(f) \quad | \quad \omega_t^1 \quad \wedge \quad \omega_t^2 \qquad (1)$$

  where $f \in F_{cyber}$ is a failure mode and ω^1, ω^2 are failure mode constraints. A failure mode constraint is True if the Boolean expression is evaluated to be True and False otherwise.
- **Timing Constraints:** Φ is a set of timing constraints defined as, $\Phi = [n], (n) | n \in \mathbb{N}_+$, where $[n]$ denotes instantaneous constraints and (n) represents periodic constraints. The timing constraints specify a pattern of time points at which the automaton checks for events and failure node constraints. For instance, periodic constraint, (4), on any outgoing transition from the current state forces the automaton to periodically look for events specified by the edge, every 4 units of time whereas in the case of instantaneous constraint, $[4]$, automaton checks only once.
- **Mode Map:** $\mathbb{M} : Q \rightarrow 2^m$ is a function that maps location $q \in Q$ to mode $m \in M$ defined in the failure propagation graph.
- **Edge:** $T \subset Q \times p(\Sigma) \times \Phi \times \alpha(F_{cyber}) \times p(\Sigma) \times Q$ is a finite set of edges. An edge represents a transition between any two locations. The activation conditions of an edge depends upon the timing, failure mode constraints and an input event. For example, an edge $< q_1$, σ_1, [n], $\delta(f_1) \wedge \neg\delta(f_2)$, σ_2, $q_2>$ represents a transition from location q_1 to q_2 with an instantaneous time constraint of n units of time and failure mode constraint $\delta(f_1) \wedge \neg\delta(f_2) \in \alpha(F_{cyber})$ defined over the failure modes $f_1, f_2 \in F_{cyber}$. $\sigma_1 \in \Sigma$, is the required input event for this transition to be valid. $\sigma_2 \in \Sigma$, represents the event generated when the transition is taken. Syntactically, a transition is represented as *Event(timing constraint){failure constraint}/Event*. In the case, no event is mentioned, then the transition is valid only if the failure mode constraint evaluates to true as per the timing

constraints.

5. POWER TRANSMISSION SYSTEM

Figure 4 shows a simple cyber physical energy system where a load *L1* is fed by two generators, *G1* & *G2* via transmission lines *TL1* and *TL2*. Buses *B1*, *B2* and *B3* act as interface points for different system elements. The example system also consists of 4 protection assemblies (*PA1*, *PA2*, *PA3* and *PA4*). Each protection assembly has a relaying system which consists of a transformer (current and voltage), a protection relay and a breaker assembly. This section briefly describes these components along with the TCD model.

5.1. Physical System (Plant)

In the context of power systems, the physical system components can be broadly classified into 3 categories A) power conversion elements B) power delivery elements C) buses. The following subsections present a brief overview of these categories. For more details, please refer to (Dugan, 2016).

Power Conversion Elements convert energy from other forms into electrical energy like generators and loads. Most of the elements in this category have one multi phase terminal. For the scope of this paper the power conversion elements are considered as black boxes where the implementation can be of variable fidelity.

Power delivery elements consists of two or more multi phase terminals. Their basic function is to transfer energy from one place to another. The most common power delivery elements are transmission lines and transformers.

Buses are the interface points for power conversion and delivery elements. Buses can be considered as N- node containers to which other components are connected.

5.2. Cyber System (Protection System)

Cyber systems include components responsible for supervisory control and protection of components in the physical system. In power systems, the cyber components include the protection relays (distance, over-current, differential relays, etc.) and circuit breakers.

Distance relays serve to protect the power grid from faults in transmission lines. A relay can act as a primary protection element for a transmission line and a backup or secondary protection for lines in the neighborhood. Distance relays work on the principle of apparent impedance ratio. The reach of a distance relay is marked in terms of zones that are functions of the impedance ratios and the direction in which the relay is configured to operate. Usually the distance relay is configured with zones 1, 2, 3 defined respectively as 80%, 125%, and 200% of the forward impedance of the transmission line to which the relay is attached.

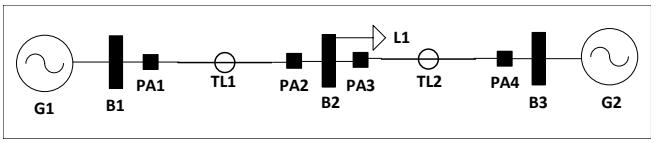

Figure 4. A simple two transmission line system

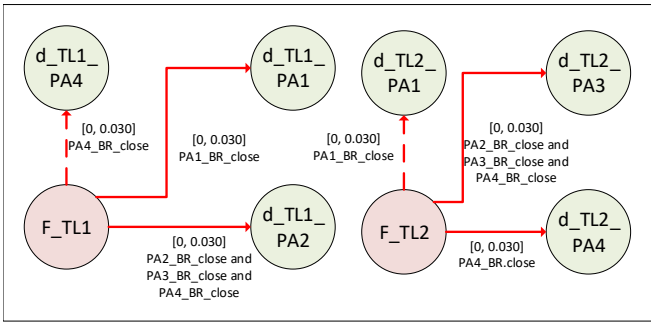

Figure 5. Failure graph for faults in two different transmission lines

When a fault occurs in a configured zone it eventually reaches the relay at which point the relay sends a trip signal to the breakers to arrest the failure effects. For faults in zone1, the distance relay serves as the primary protection element and acts without any delay. For faults in other zones, it serves as a backup and is configured to wait for a certain time (after fault detection) to allow a primary relay to respond to the fault. Typically this value is in the range [0.08, 0.167] sec and [0.250, 1] sec for zone 2 and 3 respectively as mentioned in (E. O. Schweitzer et al., 2014; Kundur et al., 1994). For the system shown in figure 4, distance relays included in PA1, PA2 act as primary protection elements for faults in line TL1 while PA4 serves as back-up or secondary protection device.

Circuit Breakers can be opened or closed to disconnect or restore power flow in the appropriate segment of the power transmission system. This can be used to stop the flow of failure effect by opening and closing the circuit upon receiving the appropriate command from the protection relays.

5.3. TCD Model

This subsection describes the TCD model of an example power system - the two transmission line system in Figure 4.

5.3.1. Failure Graphs

In power systems, protection elements are deployed redundantly to detect and isolate faulty components. The TCD failure graph for power systems is constructed in terms of the faults in the physical system and the effects observed by the protection devices.

The failure graph involving physical faults in a two transmis-

sion line system is shown in Figure 5. The nodes labeled as F_TL1 and F_TL2 represent failures in transmission lines $TL1$ and $TL2$. The effect of these failures is signaled by the alarms raised by distance relays in protection assemblies, *PA1*, *PA2*, *PA3*, *PA4*. The failure propagation is captured by an edge between the failure node, $F\_TL\mathbf{n}$ and discrepancy, $d\_TL\mathbf{n}\_PA\mathbf{k}$, where $F\_TL\mathbf{n}$ represents a fault in line $TL\mathbf{n}$ and $d\_TL\mathbf{n}\_PA\mathbf{k}$ represents an anomaly detected by protection assembly $PA\mathbf{k}$ due to a fault in line $TL\mathbf{n}$. The physical effect corresponding to this anomaly is a reduction in impedance that is observed from relay data in the form of zone 1, 2, 3 alarms (described in next section).

Failure propagation delay depends upon the time taken by the failure effect to reach the bus where the distance relay is installed along with the time taken to detect the fault conditions. Typically, this is close to 30 milliseconds as mentioned in (E. O. Schweitzer et al., 2014). Failure propagation edge activation conditions are expressed in terms of the states of the breakers in the path between the protection assembly and the generator (source). As shown in Figure 5, in order for *PA4* to detect a fault in line TL1 the breakers in assemblies *PA4*, *PA3*, *PA2* should be closed. Thus, the operating condition for the effect of a failure to travel from node F_TL1 to d_TL1_PA4 is captured by the boolean expression, *PA4_BR_close and PA3_BR_close and PA2_BR_close*.

The ability of a protection element to detect a fault depends upon number of factors, mainly, the location of the fault with respect to the protection assembly, nature of the power flow (forward or backward), physical state of the breakers, and the loading conditions. The protection elements located at the remote end are known to over- or under-reach. Hence, the failure propagation links between failure nodes and discrepancies related to remote or back up protection elements are marked uncertain, *ND(e) = False*, and are represented by dotted lines. As shown in Figure 5, *PA4* acts as a back-up protection device for faults in line *TL1*. Thus the link between F_TL1 and d_TL1_PA4 is marked uncertain.

We further classify discrepancies associated with faults in each transmission line as primary and secondary discrepancies, where primary discrepancies are associated with primary protection devices for the faults associated with the transmission line and secondary discrepancies are related to back up protection devices (described in the next section).

5.3.2. Discrete Behavioral Model: Distance Relay

Figure 6 shows the discrete model of a typical relaying system containing a distance relay (protection relay) and a breaker assembly. The distance relay model consists of three zones of protection. Table 1 summarizes the failure modes and events (observable and unobservable) considered in the distance relay model.

Table 1. TCD language elements (Failure Modes and Events) associated with distance relay behavior and observer model

Language Element	Type	Description
F_de1	Failure Mode	This fault prevents the distance relay from detecting faults in the transmission line.
F_de2_z1, F_de2_z2, F_de3_z3	Failure Mode	These faults correspond to incorrectly detecting a physical fault in zone 1, 2 and 3 reach respectively.
E1, E2, E3	Event	These unobservable events represent presence of zone 1, zone 2 and zone 3 fault conditions.
Z1, Z2, Z3	Event	These events are triggered after detecting zone 1, zone 2 and zone 3 fault conditions.
cmd_open, cmd_close	Event	These events are related to the command sent by distance relay to breaker to open, close the line.
c_reset	Event	This event forces the distance relay to be reset-ed to idle state from tripped state.
TripSen	Event	This event ensures the distance relay has sent the permissive trip signal to the relay at the other end of line after detecting a zone 1 fault condition either due to E1 or F_de2_z1.
TripRec	Event	This event is associated to arrival of trip permission from the relay at the other end.
h_Z1, h_Z2, h_Z3	Event	These alarms are issued by the observer state machine to signal presence of zone fault conditions
h_Z1′, h_Z2′, h_Z3′	Event	These alarms are issued by the observer state machine to signal disappearance of zone fault conditions

Table 2. Different states of distance relay model

State Label	Description
idle	In this state, automaton continuously checks for anomalies in plant layer.
chkZ2 (chkZ3)	This state implies the automaton has detected a zone 2(3) fault and waiting for zone 2(3) wait time to expire.
waiting1 (waiting2)	This state implies zone 2(3) wait time has expired.
tripped	The state represents that distance relay has issued an command to open a line.
detErr1	This state implies a missed detected fault in relay.
detErr2 (detErr3)	This state shows the presence of zone 2(3) spurious detection fault.

The automaton consists of 9 states, which are described, in Table 2. Initially the automaton is in the idle state and looks for fault-condition i.e. events - *E1*, *E2*, *E3*, and checks the status of failure modes every *R* seconds. If the distance relay detects zone 1 fault conditions (modeled by the presence of the event *E1*) , then the distance relay moves to the *tripped* state and issues a *Z1* alarm and commands the breaker to open (cmd_open). For zone 2 and zone 3 faults conditions (E2, E3), the protection relay does not issue an open command after moving to the *chkZ2* or *chkZ3* states. The state machine waits for predefined time, $zn2wt, zn3wt \in \mathbb{R}_+$ and checks again for the presence of the fault conditions. If the fault is still present, the relay commands the breaker to open. Additionally, distance relays may be configured with overreach trip transfer protocols. In this case, the primary relays associated with a transmission line send permissive trip signals to each other, in order to avoid zone 2 wait time.

In the presence of internal faults, the distance relay may not

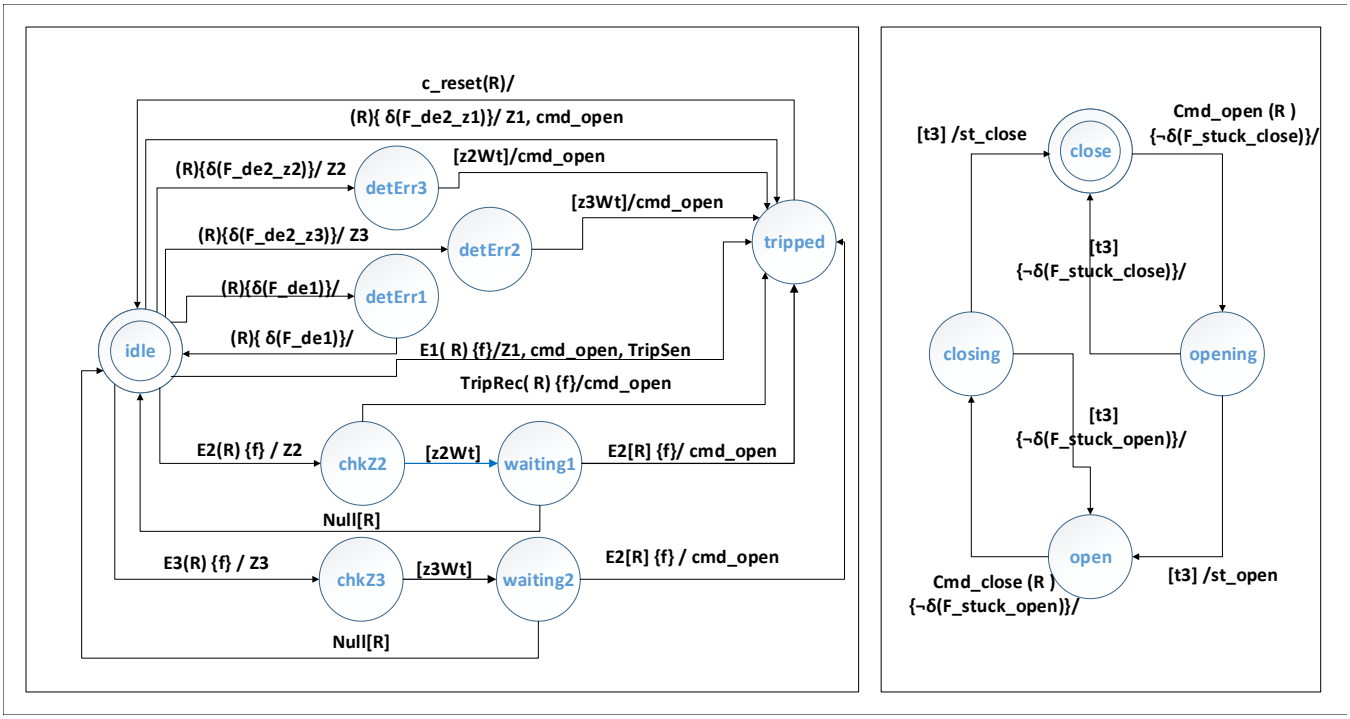

Figure 6. Protection System Behavior Components (Left: Distance Relay; Right: Breaker), where f is a failure mode constraint defined as, f:$\neg\delta$(F_de1) $\wedge\neg\delta$(F_de2_z1) $\wedge\neg\delta$(F_de2_z2) $\wedge\neg\delta$(F_de2_z3)

detect physical faults. This is modeled by the presence of a missed detection fault, *F_de1*, where the relay jumps to *detErr1* state and does not detect any physical faults. In certain situations the distance relay could have internal faults related to spurious detection (*F_de2_z1*, *F_de2_z2*, *F_de2_z3*). In such cases, as modeled in the automaton, it incorrectly reports zone 1, zone 2 or zone 3 faults by moving to *detErr2*, and *detErr3* and instructs the breaker to open. In this model, the faults (*F_de1*, *F_de2_z1*, *F_de2_z2*, *F_de2_z3*) are assumed to be mutually exclusive.

5.3.3. Discrete Behavioral Model: Circuit Breaker

Figure 6 also shows an abstract model of a single phase breaker. The different failure modes, and events associated with the breaker behavioral model are summarized in Table 3. The state machine consists of 4 states:

- **open**: This state implies that the physical state of the breaker is open.
- **close**: This state represents that the physical state of the breaker is close.
- **opening**: Due to the mechanical nature of the breaker assembly and zero crossing detection, the transition from *open* state to *close* is not instantaneous. The *opening* state represents the intermediate state where the breaker has received the command to open but the physical state is not open.
- **closing**: Similar to *opening* state, *closing* is an intermediate state that implies that breaker assembly has received a closing command but the status is not yet closed.

State *close* is the initial state of the automaton and after every R seconds, the automaton checks for *cmd_open* event and the presence of F_stuck_close failure mode. If the failure mode is not present, the breaker state machine moves to *opening* state. In *opening* state, the state machine waits for *t3* units of time before transitioning to *open* state. *t3* is a parameter of the behavior model that captures the lag due to the mechanical nature of the breaker and is of the range $[0, 50]$ milliseconds as mentioned in (E. O. Schweitzer et al., 2014). Similarly, in the *open* state, the automaton looks for *cmd_close* event and the status of F_stuck_open failure mode. The automaton moves to *closing* state and after *t3* seconds moves to *close* state.

The TFPG model shown in Figure 5, and multiple copies of the behavioral models shown in Figure 6 constitute the system TCD model for the two transmission line system. A valid sample trace of such a system will be as follows: 3 phase to ground fault introduced in the middle of the line at t=0.5 secs. This causes zone 1 fault conditions for primary relays in assemblies *PA1* and *PA2* and zone 3 for the backup *PA4*. All the relays detect the fault at t = 0.501 secs and instructs the breaker to open. The breaker changes the mode and isolates the fault at t = 1.502 secs.

Table 3. Language Elements (Failure Modes and Events) for breaker behavior and observer model

Language Element	Type	Description
F_stuck_open (F_stuck_close)	Failure Mode	These faults force the breaker to remain in open (close) state irrespective of commands received from distance relay.
cmd_open (cmd_close)	Event	These events are related to the command received by breaker to open (close) the line.
st_open (st_close)	Event	These events are related to change in the state of breaker from close to open (open to close) .
h_stuck_open, h_stuck_close	Event	These events signify the presence of stuck open and closed faults respectively.
h_stuck_open′, h_stuck_close′	Event	These events signify the disappearance of stuck open and closed faults respectively.
h_open, h_close	Event	These are output events that confirms the state of breaker has changed from close to open and vice-a-versa.

6. DIAGNOSIS SYSTEM

The TCD based diagnosis system employs a hierarchical framework as shown in Figure 1. The lower layer includes observers that track the operation of cyber components (distance relays and circuit breakers) to detect and locally diagnose faults in physical and protection systems. The observers feed their results to the reasoning engine. The TCD reasoning engine produces a set of hypotheses that explain the current system states as per the output of various observers by traversing the failure propagation graph. The traversal is constrained by the state of the protection system as predicted by observers tracking it. The following subsections provides a detailed description of the model and operation of the observers and the TCD reasoner.

6.1. Observers

Observers are responsible for detecting and diagnosing faults in the cyber components (protection equipment in electric grids) by tracking their behavior. The observers monitor the observable events generated by the cyber components.The timed events produced by the various observers fall into two categories; an estimation of a state change in discrete components, and a discrepancy detection. The detected anomalies and the local estimate of the state of different components in the plant and protection layer are passed by the observer to the next layer for system level diagnosis. The observer models related to the distance relay and the circuit breaker are described as follows:

6.1.1. Observer: Distance Relay

The TTA model of a distance relay observer can be seen in Figure 7. The state machine has 8 states with *idle* being the initial state. The events attributed to the distance relay observer machine are summarized in Table 1 (last two rows). The observer remains in the *idle* position until zone fault conditions are reported by the corresponding distance relay. Once the distance relay fires a *Z1* event, the observer machine jumps to the *chkZ1* state. The observer machine waits for *t2* seconds for open command (cmd_open event). If received, the observer moves to the tripped state, otherwise transitions back to idle state. *t2* is a parameter of the distance relay observer machine that models propagation delay and relay frequency. Please note that the transition from *chkZ1* state to the *idle* state implies a communication channel fault, but in this paper we are not considering such faults.

Similarly, the observer machine moves to the *chkZ2* state when the distance relay reports a Z2 event after detecting zone 2 fault conditions. Upon the confirmation of zone 2 fault, the observer waits *t3* seconds for the arrival of the *cmd_open* command. *t3* is a parameter which is equal to the sum of zone 2 wait time and t2. If the *cmd_open* event is not observed within *t3* seconds the automaton moves back to the *idle* state and concludes that the zone 2 fault condition has disappeared. The observer machine moves from *chkZ2* state to *chkZ2_Z1* state if the event *TripRec* occurs and waits for the *cmd_open* event. In a similar fashion, the distance relay observer diagnoses zone 3 faults. The observer layer generates *h_Z1*, *h_Z2*, *h_Z3* time stamped events to signal the TCD reasoner regarding the local diagnosis of physical faults (zone 1, zone 2, zone 3) and emits *h_Z1′*, *h_Z2′*, *h_Z3′* to signal the disappearance of zone 1, 2, and 3 fault conditions. From the tripped state the observer moves to idle state when a reset signal is observed and updates the physical component to be fault free by issuing *h_Z1′*, *h_Z2′*, *h_Z3′* events.

6.1.2. Observer: Circuit Breaker

The breaker observer model is shown in the right side of Figure 7. It consists of 4 states labeled as *open*, *close*, *opening* and *closing* and correlate directly to the 4 states of the breaker automaton. Table 3 lists all the events associated with the breaker observer model. Initially the state machine is in the *close* state and jumps to the *opening* state after observing *cmd_open* event. The breaker observer transitions to the open state if it receives an *st_open* event from the breaker assembly within *t4* seconds. *t4* is a model parameter that is equal to the sum of propagation time and the maximum time required to open the breaker. If the event is observed in the time limit, the observer concludes the physical state of breaker is open. Otherwise it hypothesizes that the breaker has a stuck fault. The fault is signaled by generating an *h_stuck_close* event. Similarly, when the breaker is in the *open* state it has the same timed behavior and an *h_stuck_open* event is generated if an *st_close* event is not observed within *t4* seconds of receiving the *cmd_close* event. The above mentioned observers (local diagnosers) are created manually by merging edges and states that do not contain observable events associated with them. There exists a number of approaches for generating discrete diagnosers for dynamic systems based on (Sampath et al., 1995; Tripakis, 2002).

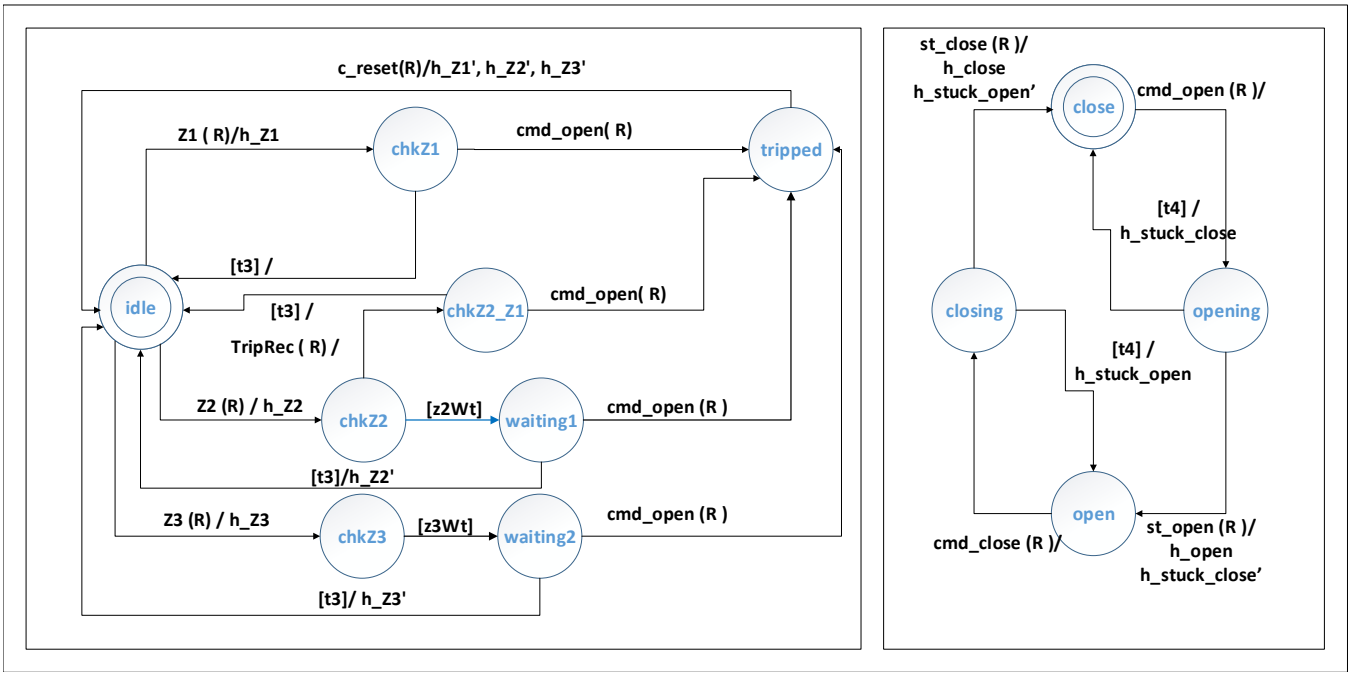

Figure 7. Protection System Observer Models, Distance Observer Model (Left); Breaker Observer Model (Right)

Table 4. Failure propagation graph for 2 transmission line system (Physical Failures)

Failure Mode	Discrepancies	ET (secs)	ND	Derived Alarms
F_TL1	d_TL1_PA1	[0, 0.030]	True	PA1_DR_OBS_h_Z1, PA1_DR_OBS_h_Z2
F_TL1	d_TL1_PA2	[0, 0.030]	True	PA2_DR_OBS_h_Z1, PA2_DR_OBS_h_Z2
F_TL1	d_TL2_PA4	[0, 0.030]	False	PA4_DR_OBS_h_Z2, PA4_DR_OBS_h_Z3
F_TL2	d_TL1_PA4	[0, 0.030]	True	PA4_DR_OBS_h_Z1, PA4_DR_OBS_h_Z2
F_TL2	d_TL1_PA3	[0, 0.030]	True	PA3_DR_OBS_h_Z1, PA3_DR_OBS_h_Z2
F_TL2	d_TL2_PA1	[0, 0.030]	False	PA1_DR_OBS_h_Z2, PA1_DR_OBS_h_Z3

Various observers in the TCD diagnosis system consume the input events from both discrete components and generate alarms for the higher level TCD reasoner. The TFPG includes such mappings between observable discrepancies related to faults in the physical plant to observer alarms. These mappings keep the reasoning engine independent from the changes in the behavioral models, while allowing for the events to be consumed by both the observer and the reasoning engine. The resultant TFPG for physical faults in the two transmission line system is listed in Table 4.

One more failure graph is created for linking cyber faults with derived alarms produced by the various observers. These cyber faults are summarized in Table 5. The failure mode, *F_PAn_DR_de1* embodies a missed detection fault in the protection relay *PAn_DR*. The associated discrepancy is *d_PAn_DR_de1*. *F_PAn_DR_de2_zk* implies a zone k spurious detection fault in the PAn_DR protection relay. These two families of cyber faults are not related to any alarms as they are inferred by the TCD reasoner by looking at the system failure propagation graph. The faults *F_PAn_BR_SC*, *F_PAn_BR_SO* imply stuck close and stuck open faults in the breaker *PAn_BR*. These are linked to discrepancies *d_PAn_BR_SC* and *d_PAn-_BR_SO* which are signaled by alarms *h_stuck_close* and *h_st-uck_open* by their respective observers.

6.2. TCD Reasoner

This section discusses the model based reasoning engine focusing on a graph-based diagnosis approach, diagnosis inputs, hypothesis structure and ranking metrics. Based on the TCD model of the system, the diagnosis engine tries to explain the observed events from the protection system (relay and breaker observers) in terms of the faults associated with the physical and/ or cyber components of the protection systems, taking into account the operating mode of the system.

6.2.1. System States and Maps

The diagnosis engine hypothesizes on the state of the nodes in the failure graph based on the outputs of the observer models. The states of a node in a failure propagation graph can be categorized as *Physical* (Actual), *Observed* and *Hypothetical State* (Abdelwahed & Karsai, 2006).

- *Physical state* corresponds to the actual state of the nodes and edges. At any time t, the physical state of any node is given by the map, $PNode_t : V \rightarrow \{ON, OFF\} \times \mathbb{R}$,

Table 5. Failure propagation graph for 2 transmission line system (Cyber Failures)

Failure Mode	Discrepancies	Alarms
F_PA1_DR_de1	d_PA1_DR_de1	–
F_PA2_DR_de1	d_PA2_DR_de1	–
F_PA3_DR_de1	d_PA3_DR_de1	–
F_PA4_DR_de1	d_PA4_DR_de1	–
F_PA1_DR_de2 _z1	d_PA1_DR_de2_z1	–
F_PA1_DR_de2 _z2	d_PA1_DR_de2_z2	–
F_PA1_DR_de2 _z3	d_PA1_DR_de2_z3	–
F_PA2_DR_de2 _z1	d_PA2_DR_de2_z1	–
F_PA2_DR_de2 _z2	d_PA2_DR_de2_z2	–
F_PA2_DR_de2 _z3	d_PA2_DR_de2_z3	–
F_PA3_DR_de2 _z1	d_PA3_DR_de2_z1	–
F_PA3_DR_de2 _z2	d_PA3_DR_de2_z2	–
F_PA3_DR_de2 _z3	d_PA3_DR_de2_z3	–
F_PA4_DR_de2 _z1	d_PA4_DR_de2_z1	–
F_PA4_DR_de2 _z2	d_PA4_DR_de2_z2	–
F_PA4_DR_de2 _z3	d_PA4_DR_de2_z3	–
F_PA1_BR_SC	d_PA1_BR_SC	PA1_BR_h_stuck_close
F_PA1_BR_SO	d_PA1_BR_SO	PA1_BR_h_stuck_open
F_PA2_BR_SC	d_PA2_BR_SC	PA2_BR_h_stuck_close
F_PA2_BR_SO	d_PA2_BR_SO	PA2_BR_h_stuck_open
F_PA3_BR_SC	d_PA3_BR_SC	PA3_BR_h_stuck_close
F_PA3_BR_SO	d_PA3_BR_SO	PA3_BR_h_stuck_open
F_PA4_BR_SC	d_PA4_BR_SC	PA4_BR_h_stuck_close
F_PA4_BR_SO	d_PA4_BR_SO	PA4_BR_h_stuck_open

where $V = D \cup F$ is the set of failure and discrepancy nodes. An *ON* state for a failure node implies the presence of the fault otherwise it is in an *OFF* state. For discrepancy nodes an ON state implies that the failure effect has reached that node. Similarly, for edges the function $PEdge_t : E \rightarrow \{ON, OFF\} \times \mathbb{R}$ gives the physical state of an edge at time t. The *ON (OFF)* state implies the edge is active (inactive). The $PNode_t$*(v).state,* $PEdge_t$*(e)*.state represents the state of a node *n* and edge *e* at time *t*. $PNode_t$*(v).time*, $PEdge_t$*(e).time* denotes the last time the state of nodes and edges were updated.

- An *observed state* is the same as the *physical state* except when there are sensor/alarm failures. The *observed state* at time t is also represented as a function defined over observable discrepancies as $ONode_t : D_{obs} \rightarrow \{ON, OFF\} \times \mathbb{R}$ where $D_{obs} \subset D$, are observable discrepancies.
- A *hypothetical state* is an estimate of the node's physical state and the time since the last state change happened. Formally a hypothetical state at time t is defined as a map $HNode_t$*:* $V \rightarrow \{ON, OFF\} \times \mathbb{R} \times \mathbb{R}$. The hypothetical state is defined for both discrepancies and failure modes. $HNode_t$*(v).terl* and $HNode_t$*(v).tlat* denotes the earliest and latest time estimates for the state changes of node v i.e. from ON to OFF or vice-a versa.

$HSet_t$ is a set that contains all hypotheses generated by the TCD reasoner. Every hypothesis, h_f in $HSet_t$ has its own $HNode_t$ map. The structure of hypothesis is defined in the following subsection.

6.2.2. Reasoner Hypothesis

Hypothesis is a tuple, where elements are related based on temporal consistency. Formally, hypothesis h_f=*{f, terl, tlat, S, C, I, M, E, U}* where:

- $\boldsymbol{f} \in \boldsymbol{F}$ is a physical failure mode projected by the hypothesis, h_f and F is the set of physical failure modes defined in section 4.1. We are using single physical fault hypothesis which lists only one fault per element of the physical system along with multiple faults in protection system.
- $\boldsymbol{S} \subseteq \boldsymbol{F_{cyber}}$ is a set of faults active in the system. These faults are related to components in the protection system layer as defined in 4.2.
- The interval $[terl, tlat]$ is the estimated earliest and the latest time during which the failure mode f could have been activated. The time estimate for protection layer faults is not supported in the current implementation.
- $\boldsymbol{C} \subseteq \boldsymbol{D}$ is the set of discrepancies that are consistent with the hypothesis h_f, where *D* is the set of physical discrepancies described in section 4.1. These discrepancies are referred to as consistent discrepancies. We partition the set *C* into two disjoint subsets, *C1*, *C2* where, *C1* consists of primary discrepancies and *C2* contains secondary discrepancies. A discrepancy, d w.r.t hypotheses h_f is called primary if the failure propagation linking the discrepancy, *d*, is certain otherwise its termed as secondary as defined in section 5.3.1.
- $\boldsymbol{E} \subseteq \boldsymbol{D}$ is the set of discrepancies which are expected to be activated in the future according to h_f. This set is also partitioned into *E1* and *E2* that contain primary and secondary discrepancies respectively.
- $\boldsymbol{M} \subseteq \boldsymbol{D}$ is the set of discrepancies that are missing according to the hypothesis h_f i.e. alarms related to these discrepancies should have been signaled. This set is also composed of two disjoint sets *M1* and *M2* based on primary and secondary discrepancies.
- $\boldsymbol{I} \subseteq \boldsymbol{D}$ is the set of discrepancies that are inconsistent with the hypothesis h_f. These are the discrepancies that are in the domain of f but cannot be explained in the current mode.
- $\boldsymbol{U} \subseteq \boldsymbol{D}$ is the set of discrepancies which are not explained by this hypothesis h_f as there is no failure propagation link between $d \in U$ and $s \in f \cup S \cup C$ i.e. the discrepancy is not in the domain of *f*.

For every scenario, the reasoner creates one special hypothesis (conservative), **H0** that associates a spurious detection fault with each of the triggered alarms.

Temporal Consistency:
The estimated states in a hypotheses need to be temporally consistent with respect to the estimated state of other nodes. Temporal consistency is a node-pair relationship that can be

applied to any arbitrary child-parent pair in the failure propagation graph (Abdelwahed & Karsai, 2006). Formally, a discrepancy *d*, is temporarily consistent with respect to a hypothesis h_f if :

- $\text{HNode}_t^{h_f}$(d) = OFF and for all (v, d) ∈ E:
  - $\text{HNode}_t^{h_f}$(v) = OFF, or
  - $\text{HNode}_t^{h_f}$(v) = ON ∧ PEdge_t(v,d).state = ON ∧ t < max($\text{HNode}_t^{h_f}$(v).tlat, PEdge_t(v,d).time) + ET(v,d).tmax
- $\text{HNode}_t^{h_f}$(d) = ON and all the following hold:
  - $\text{HNode}_t^{h_f}$(d).terl ≥ $\min_{v \in U_d}\{\text{HNode}_t^{h_f}(v).\text{terl} + \text{ET(v,d).tmin}\}$
  - $\text{HNode}_t^{h_f}$(d).tlat ≤ $\min_{v \in U_d}\{\text{HNode}_t^{h_f}(v).\text{tlat} + \text{ET(v,d).tmax}\}$, where U_d = {v ∈ V | (v,d) ∈ E and $\text{HNode}_t^{h_f}$(v) = ON}

Hypothesis Ranking:
The quality of the generated hypotheses are measured based on three metrics, Plausibility, Robustness and Failure Mode Count as explained in (Mahadevan et al., 2014). We are extending this list by adding a new criterion, called Rank. The complete metric list is defined as follows:

- ***Plausibility***: It is a measure of the degree to which a given hypothesis explains the current fault and its failure signature. Mathematically, it's is defined as

$$Plausibility = \frac{|C1|+|C2|}{|C1|+|C2|+|M1|+|I|}$$

- ***Robustness***: It is a measure of the degree to which a given hypothesis will remain constant. Mathematically, it's is defined as

$$Robustness = \frac{|C1|+|C2|}{|C1|+|C2|+|M1|+|E1|+|E2|+|I|}$$

- ***Rank***: It is a measure that a given hypothesis (a single physical fault along with multiple cyber faults) completely explains the system events observed. Mathematically, it is defined as, $Rank = |C1|+|C2|-|M1|-|U|$
- ***Failure Mode Count***: is a measure of how many failure modes are listed by the hypothesis. The reasoner gives preference to hypotheses that explain the alarm events with a limited number of failure modes (parsimony principle). This metric plays an important role while pruning out **H0** from the final hypothesis report.

6.2.3. Reasoner Input Events

There are three types of events that invoke the reasoner to update the hypotheses. The first two are external physical events related to a change in the physical state of a monitored discrepancy and system mode. The third event is an internal timeout event that corresponds to the expectation of an alarm. A physical event is formally defined as a tuple *e* = (x,t), where x ∈ $D_0 \cup M$ is either an observable discrepancy or a system mode. The timeout event is described as a tuple *e* = < h_f, d, t> which implies as per hypothesis h_f, any alarm related to discrepancy d should have been signaled by time t.

6.2.4. Reasoner Response

This section describes in details the behavior of the TCD reasoner by explaining the underlying algorithms that handle both internal and external events. The algorithm, *HandleDisccrepancyStateChangeEvent* is invoked to update appropriate hypothesis in $HSet_t$. If none of the hypotheses are able to explain this event a new hypothesis is created as described by the algorithm, *CreateNewHypothesis*. The mode change and time out events are handled by *HandleModeChangeEvent* and *HandleTimeOutEvent* respectively. The following subsections discuss these algorithms in more detail.

CreateNewHypothesis(d,t,m): Algorithm 1 deals with creation of new hypotheses to explain the change in state of a discrepancy, *d*. This procedure is triggered by the reasoner when the new state of the discrepancy *d* is not consistent with any of the existing hypotheses in $HSet_t$. A new hypothesis is created (line 2-3) for each failure mode with which the discrepancy d is temporally consistent. Further, for each hypothesis the set of consistent (line 4-5), expected (line 6-7), missing (line 8), inconsistent (line 9) and unrelated (line 10) discrepancies are identified. Appropriate timeout events are added to the global event queue for every discrepancy in the expected set (line 15-18).

HandleDiscrepancyStateChangeEvent(e,m): Algorithm 2 deals with updating every hypothesis in the set $HSet_t$ when a change is observed in the state discrepancy *d*. The change in discrepancy state is signaled by the event (d, t). For every hypothesis in $HSet_t$, the temporal consistency of discrepancy *d* is checked by routine $TConsis_t()$ (line 9), based on the constraints described in section 6.2.2.

If the new state of *d* is ON and is temporally consistent with the hypothesis, then the discrepancy is moved from the expected sets (*E1* or *E2*) to the consistent sets (*C1* or *C2*) (line 9-20). Further, new discrepancies are added to the expected sets (*E1*,*E2*) based on the failure propagation from discrepancy *d* (line 21-31). Also, timeout events are created for each new discrepancy that is added to the expected set, based on the maximum propagation time listed in *ET* map (line 23-29). If the state of *d* is OFF and it is temporally consistent, then the discrepancy is removed from the consistent sets (*C1*, *C2*) and corresponding child discrepancies are deleted from the expected sets (*E1*,*E2*) (line 32-49).

If the discrepancy *d* is not temporally consistent in the current system mode, then it is moved to the inconsistent set (line 50-

Algorithm 1 *CreateNewHypothesis(d, t, m)*: Algorithm for creating a new hypothesis

1: Input: d, where d$\in$ D , t$\in \mathbb{R}_+$, m (current system mode)
2: **for all** f $\in$ Parent(d) **and** F **do**
3: **if** PEdge(f,d).state = ON **and** ET(f,d).tmin $\leq$ (t - PEdge(f,d).time) $\leq$ ET(f,d).tmax **and** EM(f,d) $\vdash$ m **then**
4: C1 = {d if ND(f,d) == TRUE}
5: C2 = {d if ND(f,d) == FALSE}
6: E1 = {$\forall$ d_1 in Child(d) s.t. ND(d,d_1) == TRUE}
7: E2 = {$\forall$ d_1 in Child(d) s.t. ND(d,d_1) == FALSE}
8: M1 = ϕ; M2 = ϕ
9: I = {$\forall$ d_1 in Reach(f) - {d} s.t. ONode$_t$(d_1).state == ON}
10: U = {$\forall$ d_1 in D s.t. ONode$_t$(d_1).state == ON }-I-{d}
11: h_f = {f, ϕ, [t-ET(f,d).tmin , t - ET(f,d).tmax], C1, C2, E1, E2, M1, M2, I, U}
12: HSet$_t$.add(h_f)
13: HNode$_t^{h_f}$(f) = {ON, [t - ET(f,d).tmin , t - ET(f,d).tmax]}
14: HNode$_t^{h_f}$((d) = {ON, [t, t]}
15: **for all** d_1 $\in$ E1 $\cup$ E2 **do**
16: t_1 = ET(d,d_1).t_{max}
17: EventQueue.add(h_f, d_1, t_1) ▷*Timeout event*
18: **end for**
19: **end if**
20: **end for**

51) based on whether the observed state of the discrepancy is ON or OFF. The discrepancy d is added to the unrelated set, when *d* is ON, but not in the domain of f (line 52-53). The above steps are bypassed if the discrepancy is associated to cyber faults. In that case, parent failure mode is added to secondary failure mode set of every hypothesis in $HSet_t$.

HandleModeChangeEvent(e,m): Algorithm 3 updates the hypotheses in $HSet_t$ after every mode change. A mode change is reported to the reasoner when any of the underlying observers detect a change in the system mode. The expected set for each hypothesis is updated using the routine MConsis$_t$() to include only those discrepancies that are reachable from the nodes in $f \cup C$ in the current system mode (line 3-16). The timeout events are suitably updated based on the changes to the expected set (line 17-33).

HandleTimeOutEvent(e): Algorithm 4 updates the hypothesis h_f for a timeout event (h_f, d_a, t) that is triggered when the observed state of the discrepancy does not change to ON by time *t*. The discrepancy, d_a, listed in the expected set *E1* (*E2*) is moved to the missing set *M1* (*M2*). Also, a protection relay missed detection failure mode i.e *F_PAn_DR_de1*, is added to the set $h_f.S$ if d_a is a primary discrepancy associated to protection device *PAn_DR*.

Algorithm 2 *HandleDsicrepancyStateChnageEvent(e,m)*: Algorithm for handling discrepancy state change event

1: Input: e = (d, t), where d$\in$ D , t$\in \mathbb{R}_+$; m (current mode)
2: isExplained = FALSE
3: **for all** h $\in$ $HSet_t$ **do**
4: **if** d $\in$ D_{cyber} **then**
5: h.S.add(Parent(d))
6: isExplained = TRUE
7: continue
8: **end if**
9: **if** TConsis$_t$(h, d) **then**
10: isExplained = TRUE
11: HNode$_t^h$(d).terl = t; HNode$_t^h$(d).tlat = t
12: HNode$_t^h$(d).state = ONode(d).state
13: **if** ONode(d).state == ON **then**
14: **if** d $\in$ h.E_1 **then**
15: h.C_1.add(d)
16: h.E_1.remove(d)
17: **else**
18: h.C_2.add(d)
19: h.E_2.remove(d)
20: **end if**
21: **for all** d_1 $\in$ Child(d) **do**
22: **if** d_1 $\notin$ h.C1 $\cup$ h.C2 $\cup$ h.E1 $\cup$ h.E2 $\cup$ h.M1 $\cup$ h.M2 **and** EM(d.d_1) $\vdash$ m **then**
23: t_1 = ET(d,d_1).t_{max}
24: **if** ND(d,d_1) **then**
25: h.E1.add(d_1)
26: **else**
27: h.E2.add(d_1)
28: **end if**
29: EventQueue.add(h, d_1, t_1) ▷*Timeout Event*
30: **end if**
31: **end for**
32: **else**
33: **if** d $\in$ h.C1 $\cup$ h.C2 **then**
34: **if** d $\in$ h.C1 **then**
35: h.C1.remove(d)
36: h.M1.add(d)
37: **else if** d $\in$ h.C2 **then**
38: h.C2.remove(d)
39: h.M2.add(d)
40: **end if**
41: **for all** d_1 $\in$ Child(d) **do**
42: **if** d_1 $\in$ h.E1 **then**
43: h.E1.remove(d_1)
44: **else if** d_1 $\in$ h.E2 **then**
45: h.E2.remove(d_1)
46: **end if**
47: **end for**
48: **end if**
49: **end if**
50: **else if** d $\in$ Domain(h) **and** ONode(d).state == ON **then**
51: h.I.add(d)
52: **else if** d **not** $\in$ Domain(h) **and** ONode(d).state == ON **then**
53: h.U.add(d)
54: **end if**
55: **end for**
56: **if** isExplained == FALSE **and** ONode(d).state == ON **then**
57: CreateNewHypothesis(d, t, m)
58: **end if**

Algorithm 3 *HandleModeChangeEvent(e)*: Algorithm for handling mode change event

```
Input: e = (m, t), where m is the updated system mode , t∈ ℝ+
for all h ∈ HSet_t do
    for all d ∈ h.E1 do
        if MConsis_t(h,d,m) then
            continue
        else
            h.E1.remove(d)
        end if
    end for
    for all d ∈ h.E2 do
        if MConsis_t(h,d,m) then
            continue
        else
            h.E2.remove(d)
        end if
    end for
    for all d ∈ h.C1 ∪ h.C2 do
        for all d_1 ∈ Child(d) do
            if d_1 ∈ h.C1 ∪ h.C2 ∪ h.E1 ∪ h.E2 ∪ h.M1 ∪ h.M2 and (d.d_1).EM ⊢ m then
                continue
            else
                t_1 = ET(d,d_1).t_max
                if (d,d_1).ND then
                    h.E1.add(d_1)
                else
                    h.E2.add(d_1)
                end if
                EventQueue.add(h, d_1, t_1)        ▷Timeout event
            end if
        end for
    end for
end for
```

Algorithm 4 *HandleTimeOutEvent(e)*: The update hypothesis algorithm for timeout event

```
Input: e = (h_f, d_a, t) where h_f ∈ HSet_t, d_a ∈ D, t∈ ℝ+
if d_a ∈ h_f.E_1 then
    h_f.E_1.remove(d_a)
    h_f.M_1.add(d_a)
    h_f.S_f.add(F_PAn_DR_de1)
else if d_a ∈ h_f.E_2 then
    h_f.M_2.add(d_a)
    h_f.E_2.remove(d_a)
end if
```

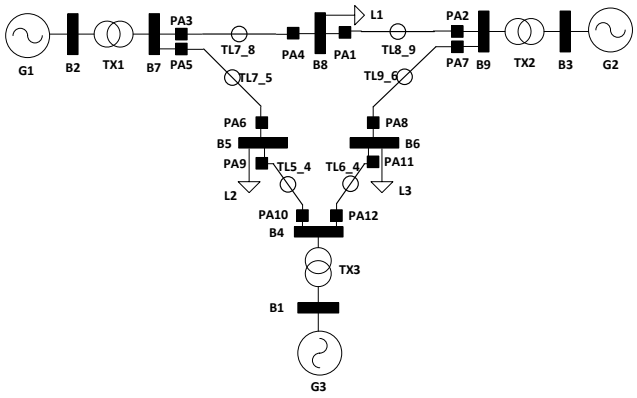

Figure 8. WSCC 9 Bus System One Line Diagram

7. CASE STUDY

The effectiveness of the reasoning approach is tested on a standard 9 Bus system (Kundur et al., 1994). This system is an approximation of the Western System Coordinating Council to an equivalent system containing 9 buses and 3 generators. Figure 8 shows the one line diagram of the 9-bus system. Table 6 lists the failure signatures for the transmission line faults. The failure graph related to cyber faults is similar to Table 5 and is not shown due to lack of space. The four scenarios considered in this paper include

- **Scenario 1**: A 3 phase to ground fault is introduced in the transmission line labeled TL_7_8, located between buses 7 and 8.
- **Scenario 2**: A Zone 3 spurious detection fault is introduced in the relay PA4_DR that forces the breaker PA4_BR to open.
- **Scenario 3**: A 3 phase to ground fault is introduced in the line TL_7_8, located between buses 7 and 8 and a stuck closed fault is injected in the breaker assembly PA4_BR.
- **Scenario 4**: A 3 phase to ground fault is introduced in the line TL_7_8, located between buses 7 and 8. A missing detection fault in relay PA4_DR and stuck closed fault in breaker PA2_BR are introduced in the protection assemblies.

The following subsections present the simulation and diagnosis results.

7.1. Event Generation

Simulink's Simscape and Stateflow toolboxes (*Simscape Power Systems: For Use with MATLAB;[user's Guide]*, 2017) are used to model and simulate the cyber physical system under study to produce the appropriate events that are fed to the diagnosis system. The simulation is carried using a fixed step discrete solver with a step size of 1 ms in phasor

Table 6. Temporal Failure Propagation Graph for WSCC 9 Bus System

Failure Mode	Discrepancies	ET (secs)	ND	Alarms
F_TL6_4	d_TL6_4_PA11	[0, 0.030]	True	PA11_DR_OBS_h_Z1, PA11_DR_OBS_h_Z2
F_TL6_4	d_TL6_4_PA12	[0, 0.030]	True	PA12_DR_OBS_h_Z1, PA12_DR_OBS_h_Z2
F_TL6_4	d_TL6_4_PA7	[0, 0.030]	False	PA7_DR_OBS_h_Z2, PA7_DR_OBS_h_Z3
F_TL5_4	d_TL5_4_PA9	[0, 0.030]	True	PA9_DR_OBS_h_Z1, PA9_DR_OBS_h_Z2
F_TL5_4	d_TL5_4_PA10	[0, 0.030]	True	PA10_DR_OBS_h_Z1, PA10_DR_OBS_h_Z2
F_TL5_4	d_TL5_4_PA5	[0, 0.030]	False	PA5_DR_OBS_h_Z2, PA5_DR_OBS_h_Z3
F_TL8_9	d_TL5_4_PA1	[0, 0.030]	True	PA1_DR_OBS_h_Z1, PA1_DR_OBS_h_Z2
F_TL8_9	d_TL5_4_PA2	[0, 0.030]	True	PA2_DR_OBS_h_Z1, PA2_DR_OBS_h_Z2
F_TL8_9	d_TL5_4_PA3	[0, 0.030]	False	PA3_DR_OBS_h_Z2, PA3_DR_OBS_h_Z3
F_TL9_6	d_TL9_6_PA7	[0, 0.030]	True	PA7_DR_OBS_h_Z1, PA7_DR_OBS_h_Z2
F_TL9_6	d_TL9_6_PA8	[0, 0.030]	True	PA8_DR_OBS_h_Z1, PA8_DR_OBS_h_Z2
F_TL9_6	d_TL9_6_PA12	[0, 0.030]	False	PA12_DR_OBS_h_Z2, PA12_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA5	[0, 0.030]	True	PA5_DR_OBS_h_Z1, PA5_DR_OBS_h_Z2
F_TL7_5	d_TL7_5_PA6	[0, 0.030]	True	PA6_DR_OBS_h_Z1, PA6_DR_OBS_h_Z2
F_TL7_5	d_TL7_5_PA7	[0, 0.030]	False	PA7_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA8	[0, 0.030]	False	PA8_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA10	[0, 0.030]	False	PA10_DR_OBS_h_Z2, PA10_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA11	[0, 0.030]	False	PA11_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA4	[0, 0.030]	False	PA4_DR_OBS_h_Z2, PA4_DR_OBS_h_Z3
F_TL7_5	d_TL7_5_PA2	[0, 0.030]	False	PA2_DR_OBS_h_Z3
F_TL7_8	d_TL7_8_PA3	[0, 0.030]	True	PA3_DR_OBS_h_Z1, PA3_DR_OBS_h_Z2
F_TL7_8	d_TL7_8_PA4	[0, 0.030]	True	PA4_DR_OBS_h_Z1, PA4_DR_OBS_h_Z2
F_TL7_8	d_TL7_8_PA2	[0, 0.030]	False	PA2_DR_OBS_h_Z2, PA2_DR_OBS_h_Z3

simulation mode. The figures 9, 10, 11 and 12 show the results of the four scenarios. Figures in the first column show the zone alarms triggered by the distance relays. Figures in the second column highlight the commands sent by the distance relays to their respective breakers whereas the third column shows the physical state of breakers (value 0 implies the state of the breaker is open)

In scenario 1, a three phase to ground fault is injected in the line at t = 0.5 secs and both the primary protection elements (PA3_DR, PA4_DR) along with secondary backup (PA2_DR) detect the fault by issuing Z1, Z2, Z3 events at t = 0.501 secs. The PA3_DR sends trip signals to relay PA4_DR and breaker PA3_BR at time t = 0.501 secs. The trip signal is received by relay PA4_DR which reduces the zone wait time and forces the relay to issue a trip signal to PA4_BR at t = 0.502 secs. The breaker assemblies PA3_BR, PA4_BR changes their state from close to open at t = 0.532 and t = 0.533 secs respectively, to isolate the fault.

In scenario 2, a spurious detection fault *F_de2_z3* is injected in the relay, PA2_DR at t = 0.3 secs. This failure mode forces the relay to issue a Z3 event even in the absence of any transmission line fault. After waiting for zone 3 wait time (1 sec), the relay issues a trip signal to breaker PA2_BR. The state of the breaker is changed at t = 1.331 secs.

In scenario 3, a three phase to ground fault is injected in the line at t = 0.5 secs and a stuck close fault is activated in breaker PA4_BR. Similar to scenario 1, PA3_DR, PA4_DR and PA2_DR all detect the fault conditions and issue Z1, Z2, Z3 events followed by trip signals from PA3_DR to PA_DR and PA3_BR. The breaker assemblies PA3_BR and PA4_BR receive trip commands at t = 0.501 and t = 0.502. PA3_BR changes its state to Open at t = 0.5332 secs. However, due to the stuck close fault in PA_BR, the trip request is ignored and PA4_BR remains in closed position. At t = 1.502, the zone 3 wait time expires and PA2_DR checks for the fault condition again. Since the fault is not cleared from B8 side, PA2_DR detects the fault and send a trip signal to breaker PA2_BR. The breaker clears the fault by taking out the line TL8_9 at t = 1.533 secs.

In scenario 4, along with a three phase transmission line fault, a missed detection fault in PA4_DR and breaker stuck close fault in PA2_BR are injected at t = 0.5 secs. PA3_DR and PA2_DR detect the fault conditions and issue Z1 and Z3 events at t = 0.501 secs. And due to the missed detection fault, PA4_DR skips the detection. PA3_DR and PA2_DR issue trip signals to their respective breakers at t = 0.501 and 1.502 secs. The state of the breaker PA3_BR changes at t= 0.532 but PA2_BR remains in the closed state due to the stuck close fault.

7.2. Diagnosis Results

Figures 13, 14, 15 and 16 show the output of various observers and the TCD reasoning engine for the fault scenarios discussed in the previous section.

In scenario 1, a persistent transmission fault is introduced at t = 0.5 sec. The distance relays PA3_DR, PA4_DR and PA2_DR detect the fault and report Z1, Z2 and Z3 events. The corresponding observers acknowledge these events and generate h_Z1, h_Z2 and h_Z3 alarms which are fed to the TCD reasoner. These alarms activate d_TL7_8_PA3, d_TL7_8_PA4, d_TL7_8_PA2, d_TL8_9_PA4 discrepancies and invoke the discrepancy state change event. These discrepancies produce three hypotheses labeled as H0, H1, H2. H0 is a special hypothesis that blames a spurious detection fault in all the relays. H1 points towards 3 phase to ground fault in TL7_8 with three consistent discrepancies whereas H2 lists a fault in TL8_9 with one consistent discrepancy. At t = 0.531 sec, a timeout event occurs which removes the discrepancies from

[0]Figures in the first column show the zone alarms triggered by the distance relays. Figures in the second column shows the commands sent by the distance relays to their respective breakers. Third column shows the physical state of breakers

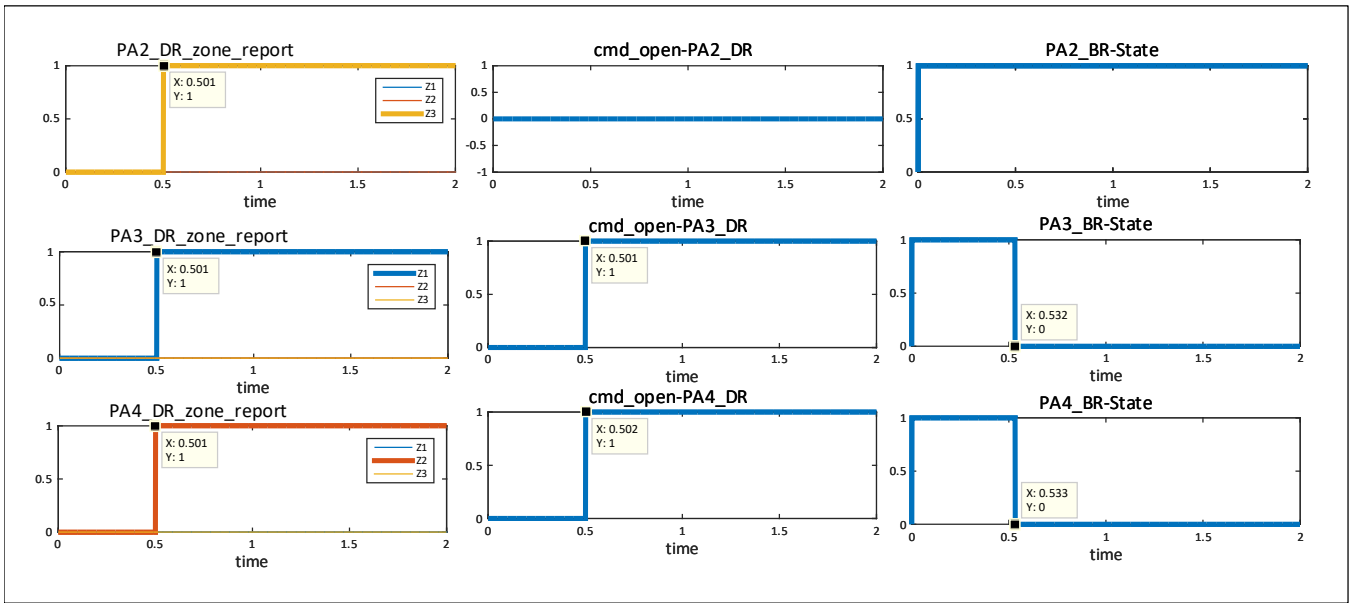

Figure 9. Simulation results for scenario 1

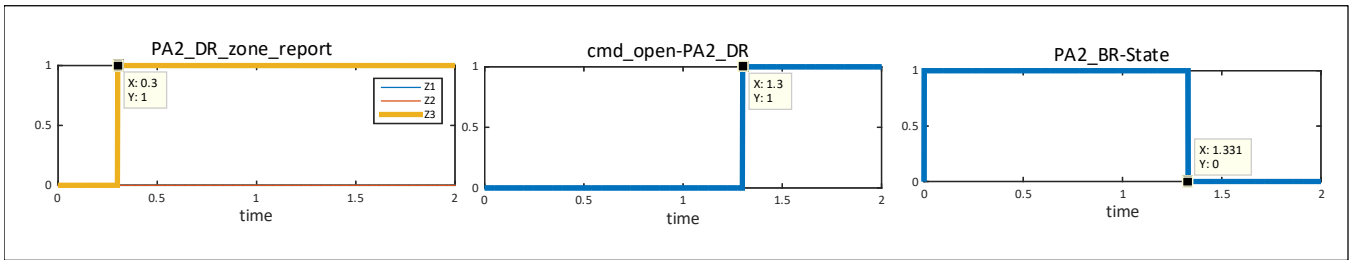

Figure 10. Simulation results for scenario 2

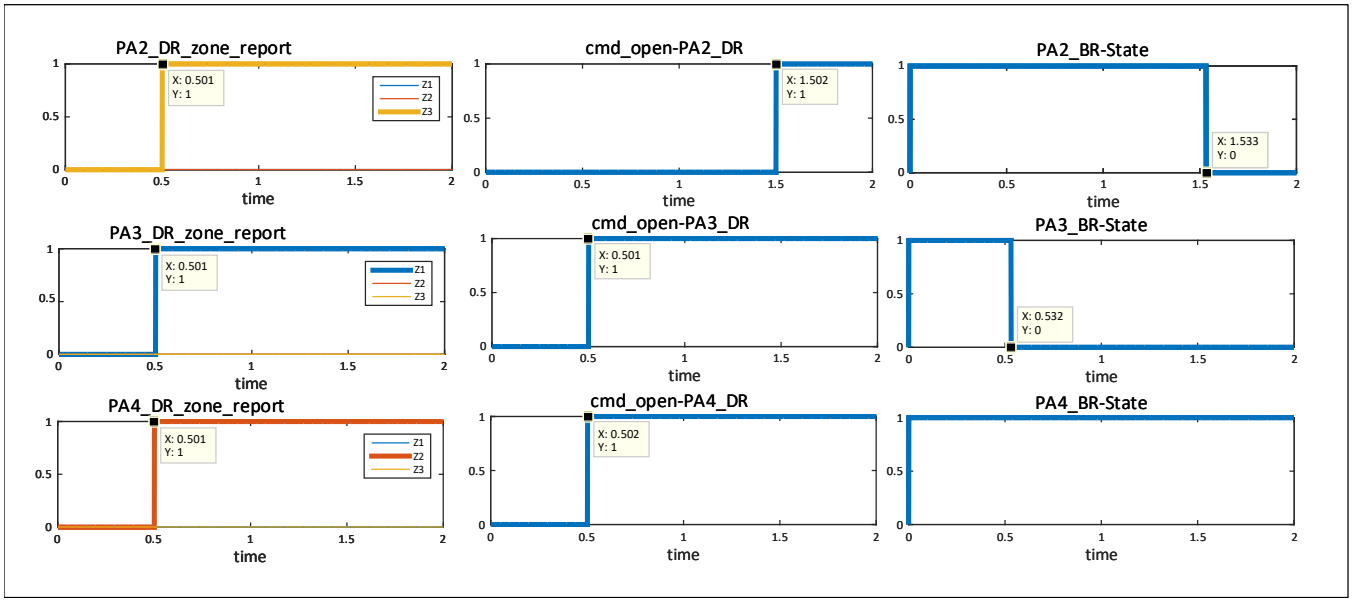

Figure 11. Simulation results for scenario 3

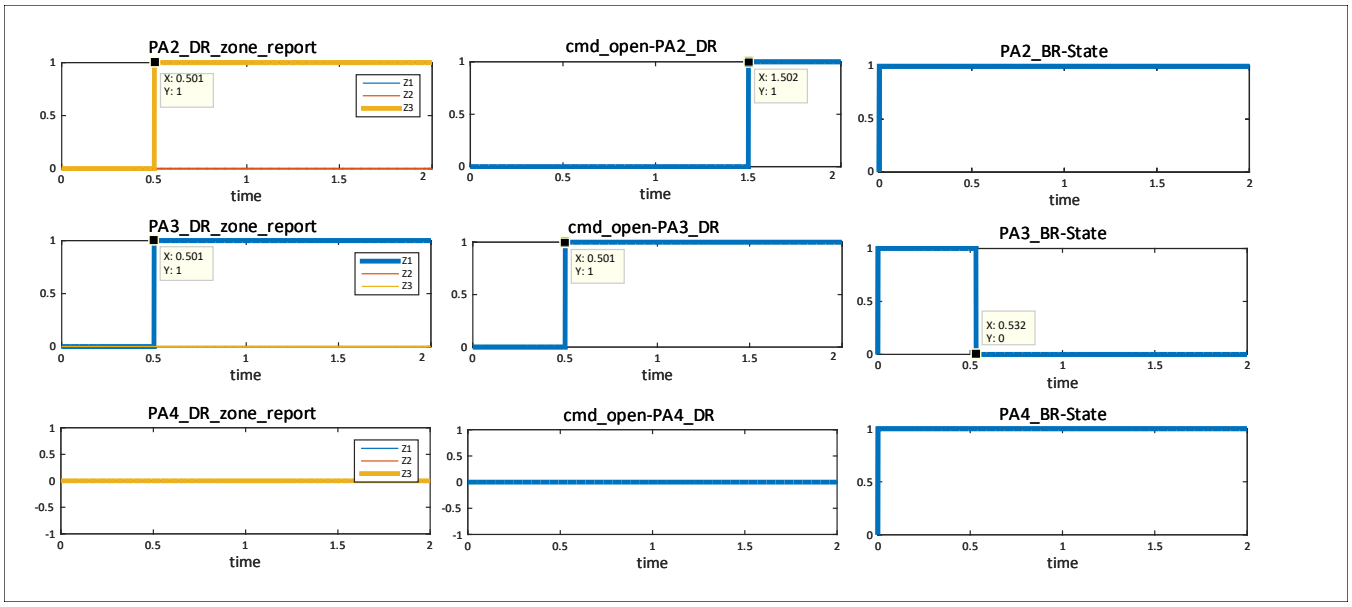

Figure 12. Simulation results for scenario 4

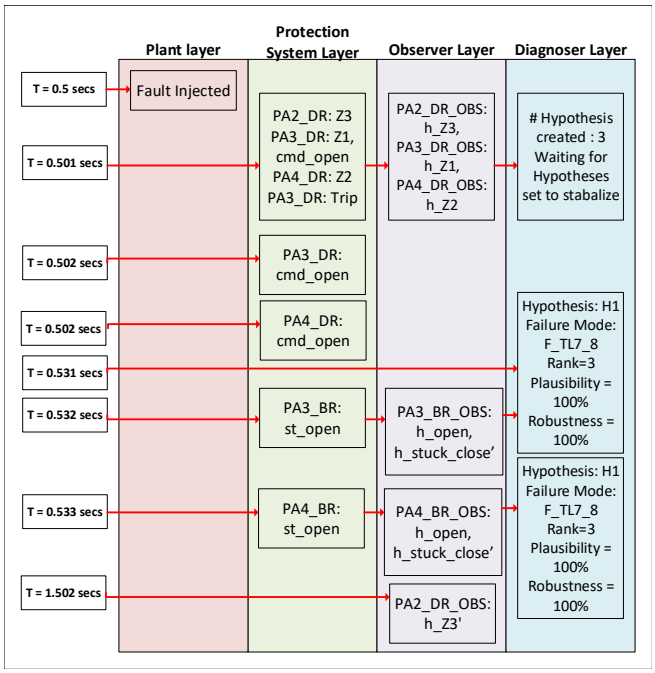

Figure 13. Diagnosis results for scenario 1

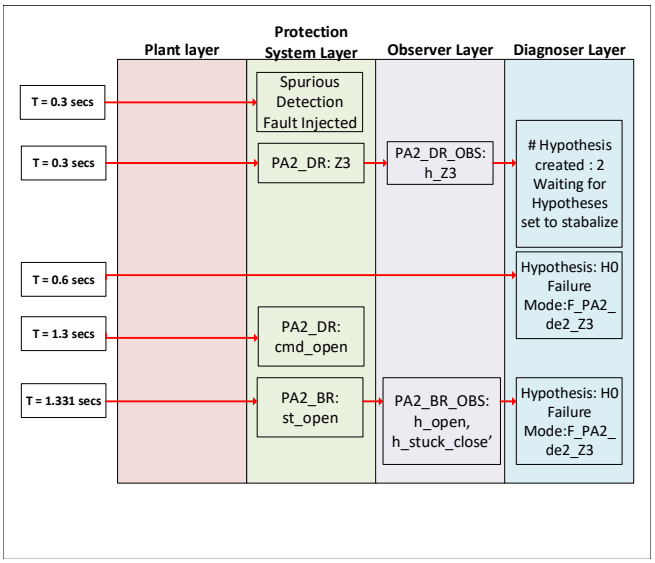

Figure 14. Diagnosis results for scenario 2

the expected set and adds them to the missing set. H1 is selected as the best hypothesis, which correctly identifies the fault with 100% plausibility. At t = 0.532 and 0.533 sec, mode change events are triggered by the observers, PA3_BR_OBS, PA4_BR_OBS due to the state change signaled by breakers PA3_BR and PA4_BR.

In scenario 2, a spurious zone 3 detection fault is introduced at t = 0.3 secs in relay PA4_DR. The observer reports the alarm to the TCD reasoner which leads to generation of two hypotheses H0, H1. H0 highlights a spurious detection fault while H1 shows a fault in line TL7_8 with one consistent discrepancy. At t = 0.6 secs, the hypothesis set stabilizes and H0 emerges as a best hypothesis (law of parsimony) as H1 lists three failure modes, (transmission line plus the missed detection faults in PA3_DR and PA4_DR).

In scenario 3, a transmission fault in TL7_8 and a stuck_close fault in the breaker assembly is injected at t = 0.5 sec. The hypothesis set evolves in similar fashion as described in scenario 1 until t = 0.532 secs. However, the observer PA4_BR_OBS

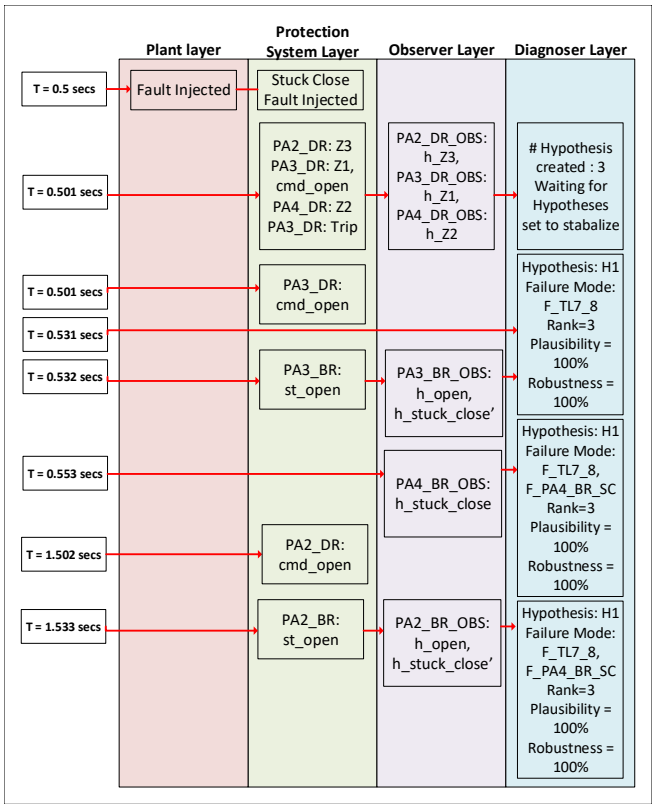

Figure 15. Diagnosis results for scenario 3

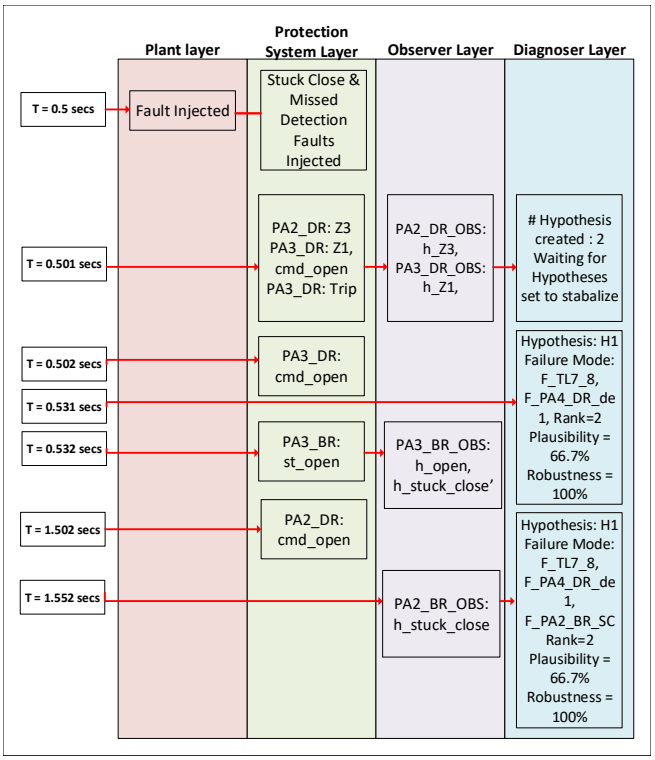

Figure 16. Diagnosis results for scenario 4

does not report a mode change and waits until t = 0.552 secs. At t = 0.553, the observer concludes stuck_close fault in the breaker and issues an alarm h_stuck_close which is transformed into a cyber fault and added to every hypothesis in the hypothesis set.

Scenario 4 involves three faults, a transmission line fault in TL7_8 along with stuck fault in PA2_BR and a missed detection fault in PA4_DR. At t = 0.501, PA3_DR_OBS and PA2_DR_OBS report h_Z1 and h_Z3 alarms. These alarms produces two hypotheses H0, H1. H1 lists faults in line TL7_8 with two consistent discrepancies and expects a zone alarm from PA4_DR_OBS. At t = 0.531, timeout forces the expected discrepancy to shift to the missing set. H1 and H0 both point towards two failure modes. H1 lists physical faults associated with line TL7_8 along with a missed detection fault in PA4_DR whereas H0 blames both the distance relays for having spurious detection faults. At t = 1.552, PA2_BR_OBS concludes a stuck fault in breaker PA2_BR after failing to receive a state change event. Both the hypotheses are updated to reflect the breaker fault. The hypothesis H1 is given preference over H0 as the probability of two cyber faults is less than a physical and a cyber fault (E. Schweitzer et al., 1997).

8. CONCLUSION

In this paper we showed a new approach to diagnosing fault in cyber-physical systems while considering the possible faults in controllers that can change the mode of behavior of the system. This approach called Temporal Causal Diagrams extends our prior work on Temporal failure propagation graphs by capturing the interaction between failure propagation graphs and discrete time behavior models, that capture the controller semantics.

The TFPG definition is extended to include uncertain edges. However, the uncertainty leads to an inherent limitation of not being able to diagnose missed detection faults in secondary protection devices.

We finally, demonstrated the extended diagnostic procedure on an WSCC-9 bus power transmission system. We are currently working on extending the diagnostic technique to provide a holistic solution that predicts imminent failure modes and presents fault mitigation strategies. We are also interested in automatic way of synthesizing TCD models from system topology. However, writing such transformations are domain dependent and require a good understanding of the underlying domain.

ACKNOWLEDGMENT

This work is funded in part by the National Science Foundation under the award number CNS-1329803. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF. The authors will like to thank Rishabh Jain, Srdjn Lukic, Scott Eisele and Amogh Kulkarni for their help and discussions related to the work presented here.

NOMENCLATURE

$\boldsymbol{D}_{cyber}$	Finite set of discrepancies associated with cyber failure modes.
d_PAn_BR_SC	Discrepancy associated with stuck closed fault in breaker, PAn_BR.
d_PAn_BR_SO	Discrepancy associated with stuck open fault in breaker, PAn_BR.
d_PAn_DR_de1	Discrepancy associated with missed detection fault in distance relay, PAn_DR.
d_PAn_DR_de2_zk	Discrepancy associated with zone k spurious detection fault in distance relay, PAn_DR.
d_TLn_PAk	Discrepancy related to fault in component, *TLn*, signaled by distance relay in protection assembly *PAk*
$\boldsymbol{D}$	Nonempty set of discrepancy nodes related faults in physical components.
$\boldsymbol{F}_{cyber}$	Finite set of failure modes associated with cyber components.
F_PAn_BR_SC	Stuck closed fault in breaker, PAn_BR.
F_PAn_BR_SO	Stuck open fault in breaker, PAn_BR.
F_PAn_DR_de1_z1	Missed detection fault associated with distance relay, *PAn_DR*
F_PAn_DR_de2_zk	Zone *k* spurious detection fault associated with distance relay, *PAn_DR*
F_TLn	Failure in transmission line, *TLn*
$\boldsymbol{F}$	Nonempty set of failure nodes in physical components.
$\boldsymbol{h}_f$	Hypothesis related to physical fault *f*.
HNode$_t$(n)	Map that defines hypothetical state of a node *n* in failure graph at time *t*.
$\boldsymbol{HSet}_t$	Set of all hypotheses at time *t*.
ONode$_t$(n)	Map that defines observed state of a node *n* in failure graph at time *t*.
PAn_BR	Circuit breaker in protection assembly, *PAn*
PAn_DR	Distance relay in protection assembly, *PAn*
PAn	Protection assembly labeled as *PAn*
PEdge$_t$(e)	Map that defines physical state of an edge *e* in failure graph at time *t*.
PNode$_t$(n)	Map that defines physical state of a node *n* in failure graph at time *t*.
TLn	Transmission line labeled as *TLn*

REFERENCES

Abdelwahed, S., & Karsai, G. (2006, Sept). Notions of diagnosability for timed failure propagation graphs. In *Autotestcon, 2006 ieee* (p. 643-648). doi: 10.1109/AUTEST.2006.283740

Abdelwahed, S., & Karsai, G. (2007). Failure prognosis using timed failure propagation graphs. *Electrical Engineering*.

Bi, T., Yan, Z., Wen, F., Ni, Y., Shen, C., Wu, F. F., & Yang, Q. (2002). On-line fault section estimation in power systems with radial basis function neural network. *International journal of electrical power & energy systems*, *24*(4), 321–328.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., & Schröder, J. (2006). *Diagnosis and fault-tolerant control* (Vol. 691). Springer.

Boem, F., Ferrari, R. M., Parisini, T., & Polycarpou, M. M. (2013). Distributed fault diagnosis for continuous-time nonlinear systems: The input–output case. *Annual Reviews in Control*, *37*(1), 163–169.

Bouamama, B. O., Biswas, G., Loureiro, R., & Merzouki, R. (2014). Graphical methods for diagnosis of dynamic systems: Review. *Annual Reviews in Control*, *38*(2), 199 - 219. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1367578814000388` doi: https://doi.org/10.1016/j.arcontrol.2014.09.004

Cardoso, G., Rolim, J. G., & Zurn, H. H. (2004, July). Application of neural-network modules to electric power system fault section estimation. *IEEE Transactions on Power Delivery*, *19*(3), 1034-1041. doi: 10.1109/TPWRD.2004.829911

Cardoso, G., Rolim, J. G., & Zurn, H. H. (2008, July). Identifying the primary fault section after contingencies in bulk power systems. *IEEE Transactions on Power Delivery*, *23*(3), 1335-1342. doi: 10.1109/TPWRD.2008.916743

Chen, W. H. (2012, April). Online fault diagnosis for power transmission networks using fuzzy digraph models. *IEEE Transactions on Power Delivery*, *27*(2), 688-698. doi: 10.1109/TPWRD.2011.2178079

Chen, W.-H., Liu, C.-W., & Tsai, M.-S. (2001, Oct). Fast fault section estimation in distribution substations using matrix-based cause-effect networks. *IEEE Transactions on Power Delivery*, *16*(4), 522-527. doi: 10.1109/61.956731

Chen, W. H., Tsai, S. H., & Lin, H. I. (2011, April). Fault section estimation for power networks using logic cause-effect

models. *IEEE Transactions on Power Delivery*, *26*(2), 963-971. doi: 10.1109/TPWRD.2010.2093585

Daigle, M. J., Koutsoukos, X. D., & Biswas, G. (2007). Distributed diagnosis in formations of mobile robots. *IEEE Transactions on Robotics*, *23*(2), 353–369.

Dubey, A., Karsai, G., & Mahadevan, N. (2011). Model-based software health management for real-time systems. In *Aerospace conference, 2011 ieee* (pp. 1–18).

Dugan, R. (2016). Opendss manual. *Electrical Power Research Institute*. Retrieved from http://sourceforge.net/apps/mediawiki/electricdss/index.php

Ferrari, R. M., Parisini, T., & Polycarpou, M. M. (2012). Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach. *IEEE Transactions on Automatic Control*, *57*(2), 275–290.

Ferreira, V., Zanghi, R., Fortes, M., Sotelo, G., Silva, R., Souza, J., ... Gomes Jr, S. (2016). A survey on intelligent system application to fault diagnosis in electric power system transmission lines. *Electric Power Systems Research*, *136*, 135–153.

Guo, W., Wei, L., Wen, F., Liao, Z., Liang, J., & Tseng, C. L. (2009, April). An on-line intelligent alarm analyzer for power systems based on temporal constraint network. In *Sustainable power generation and supply, 2009. supergen '09. international conference on* (p. 1-7). doi: 10.1109/SUPERGEN.2009.5347900

Guo, W., Wen, F., Ledwich, G., Liao, Z., He, X., & Liang, J. (2010, July). An analytic model for fault diagnosis in power systems considering malfunctions of protective relays and circuit breakers. *IEEE Transactions on Power Delivery*, *25*(3), 1393-1401. doi: 10.1109/TPWRD.2010.2048344

He, Z., Chiang, H.-D., Li, C., & Zeng, Q. (2009). Fault-section estimation in power systems based on improved optimization model and binary particle swarm optimization. In *Power & energy society general meeting, 2009. pes'09. ieee* (pp. 1–8).

Huang, Y.-C. (2002, May). Fault section estimation in power systems using a novel decision support system. *IEEE Transactions on Power Systems*, *17*(2), 439-444. doi: 10.1109/TPWRS.2002.1007915

Isermann, R. (2006). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media.

Jung, J., Liu, C.-C., Hong, M., Gallanti, M., & Tornielli, G. (2001, Apr). Multiple hypotheses and their credibility in on-line fault diagnosis. *IEEE Transactions on Power Delivery*, *16*(2), 225-230. doi: 10.1109/61.915487

Khalili, M., & Zhang, X. (2014, Dec). Distributed fault detection in interconnected nonlinear uncertain systems. In *53rd ieee conference on decision and control* (p. 6548-6553). doi: 10.1109/CDC.2014.7040416

Krčál, P., Mokrushin, L., Thiagarajan, P., & Yi, W. (2004). Timed vs. time-triggered automata. In *Concur 2004-concurrency theory* (pp. 340–354). Springer.

Kundur, P., Balu, N., & Lauby, M. (1994). *Power system stability and control*. McGraw-Hill. Retrieved from https://books.google.com/books?id=2cbvyf8Ly4AC

Mahadevan, N., Dubey, A., & Karsai, G. (2011). Application of software health management techniques. In *Proceedings of the 6th international symposium on software engineering for adaptive and self-managing systems* (pp. 1–10). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1988008.1988010 doi: 10.1145/1988008.1988010

Mahadevan, N., Dubey, A., Karsai, G., Srivastava, A., & Liu, C.-C. (2014). Temporal causal diagrams for diagnosing failures in cyber-physical systems. *Annual Conference of the Prognostics and Health Management Society*. Retrieved from http://www.phmsociety.org/node/1439

Mahanty, R. N., & Gupta, P. B. D. (2004, March). Application of rbf neural network to fault classification and location in transmission lines. *IEE Proceedings - Generation, Transmission and Distribution*, *151*(2), 201-212. doi: 10.1049/ip-gtd:20040098

North American Electric Reliability Corporation. (2012). *2012 state of reliability* (Tech. Rep.). Retrieved from http://www.nerc.com/files/2012_sor.pdf

Padalkar, S., Karsai, G., Biegl, C., Sztipanovits, J., Okuda, K., & Miyasaka, N. (1991, June). Real-time fault diagnostics. *IEEE Expert*, *6*(3), 75-85. doi: 10.1109/64.87689

Reppa, V., Polycarpou, M. M., & Panayiotou, C. G. (2013). Multiple sensor fault detection and isolation for large-scale interconnected nonlinear systems. In *Control conference (ecc), 2013 european* (pp. 1952–1957).

Reppa, V., Polycarpou, M. M., & Panayiotou, C. G. (2015a). Decentralized isolation of multiple sensor faults in large-scale interconnected nonlinear systems. *IEEE Transactions on Automatic Control*, *60*(6), 1582–1596.

Reppa, V., Polycarpou, M. M., & Panayiotou, C. G. (2015b, March). Distributed sensor fault diagnosis for a network of interconnected cyberphysical systems. *IEEE Transactions on Control of Network Systems*, *2*(1), 11-23. doi: 10.1109/TCNS.2014.2367362

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995, Sep). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555-1575. doi: 10.1109/9.412626

Schweitzer, E., Fleming, B., Lee, T. J., Anderson, P. M., et al. (1997). Reliability analysis of transmission protection using fault tree methods. In *Proceedings of the 24th annual western protective relay conference* (pp. 1–17).

Schweitzer, E. O., Kasztenny, B., Guzmán, A., Skendzic, V., & Mynam, M. V. (2014). Speed of line protection–can we break free of phasor limitations? In *41st annual western protective relay conference, spokane, washington usa.*

Sekine, Y., Akimoto, Y., Kunugi, M., Fukui, C., & Fukui, S. (1992). Fault diagnosis of power systems. *Proceedings of the IEEE*, *80*(5), 673–683.

Shames, I., Teixeira, A. M., Sandberg, H., & Johansson, K. H. (2011). Distributed fault detection for interconnected second-order systems. *Automatica*, *47*(12), 2757–2764.

Simscape power systems: For use with matlab;[user's guide]. (2017). MathWorks.

Sun, J., Qin, S.-Y., & Song, Y.-H. (2004, Nov). Fault diagnosis of electric power systems based on fuzzy petri nets. *IEEE Transactions on Power Systems*, *19*(4), 2053-2059. doi: 10.1109/TPWRS.2004.836256

Thukaram, D., Khincha, H. P., & Vijaynarasimha, H. P. (2005, April). Artificial neural network and support vector machine approach for locating faults in radial distribution systems. *IEEE Transactions on Power Delivery*, *20*(2), 710-721. doi: 10.1109/TPWRD.2005.844307

Tripakis, S. (2002). Fault diagnosis for timed automata. In *International symposium on formal techniques in real-time and fault-tolerant systems* (pp. 205–221).

Wen, F., & Chang, C. (1997). Probabilistic approach for fault-section estimation in power systems based on a refined genetic algorithm. In *Generation, transmission and distribution, iee proceedings-* (Vol. 144, pp. 160–168).

Wu, Y.-X., ning Lin, X., hong Miao, S., Liu, P., qing Wang, D., & bin Chen, D. (2005). Application of family eugenics based evolution algorithms to electric power system fault section estimation. In *Transmission and distribution conference and exhibition: Asia and pacific, 2005 ieee/pes* (p. 1-5). doi: 10.1109/TDC.2005.1546813

Yan, X.-G., & Edwards, C. (2008). Robust decentralized actuator fault detection and estimation for large-scale systems using a sliding mode observer. *International Journal of control*, *81*(4), 591–606.

Yongli, Z., Limin, H., & Jinling, L. (2006, April). Bayesian networks-based approach for power systems fault diagnosis. *IEEE Transactions on Power Delivery*, *21*(2), 634-639. doi: 10.1109/TPWRD.2005.858774

Yongli, Z., Yang, Y. H., Hogg, B. W., Zhang, W. Q., & Gao, S. (1994, Feb). An expert system for power systems fault analysis. *IEEE Transactions on Power Systems*, *9*(1), 503-509. doi: 10.1109/59.317573

Zhang, Q., & Zhang, X. (2013a). Distributed sensor fault diagnosis in a class of interconnected nonlinear uncertain systems. *Annual Reviews in Control*, *37*(1), 170–179.

Zhang, Q., & Zhang, X. (2013b). Distributed sensor fault diagnosis in a class of interconnected nonlinear uncertain systems. *Annual Reviews in Control*, *37*(1), 170–179.