Towards Diagnosing Cascading Outages in Cyber Physical Energy Systems using Temporal Causal Models

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

[1,2,3,4,5] *Institute of Software Integrated Systems, Vanderbilt University, Nashville, TN, 37212, U.S.A*
ajay.d.chhokra@@Vanderbilt.edu
nag.mahadevan@Vanderbilt.Edu
abhishek.dubey@Vanderbilt.Edu
daniel.a.balasubramanian@Vanderbilt.Edu
gabor.karsai@Vanderbilt.Edu

ABSTRACT

Cascading failures in critical cyber physical systems such as power systems are rare but lead to huge social and economic implications. Timely diagnosis of faults in these systems is a challenging task due to inherent heterogeneity and scale of the system. In the past, we have successfully demonstrated a robust technique for diagnosing independent component faults using Temporal Causal Diagrams (TCD) at sub-system level. In this paper, we present a systematic approach of using the sub-system level fault models to auto-generate a system-level fault model that helps in diagnosing cascading failures. We show the time complexity of our model generation algorithm using industry standard Power Transmission networks. Further, we describe the updates to the existing TCD reasoner algorithms and report the TCD diagnosis results for simulated multi fault scenario on a standard power system.

1. INTRODUCTION

Cascading failures in networked systems are defined as a set of one or more independent events that triggers a sequence of dependent events. The cascading chain of failures successively weakens the system resulting in total system collapse. According to North American Reliability Corporation, (NERC, 2005), the uncontrolled loss of any system facilities or load, whether because of thermal overload, voltage collapse, or loss of synchronism, except those occurring as a result of fault isolation. Utilities are required by regulators (NERC for the US) to ensure that the system does not operate at any time with a possibility of a critical outage (NERC, 2013). This makes the timely diagnosis of faults in power systems operations and planning an important task for ensuring the smooth running of the system.

Power systems are large complex cyber-physical systems that contain tightly coupled components of both continuous and discrete nature. Physical components (continuous) are transmission lines, loads, generators etc which are controlled and protected by embedded implementations of control algorithms such as Automatic Generation Control (AGC) and microprocessor based relays. The cyber infrastructure includes protection devices along with Energy Management System (EMS) and Supervisory Control And Data Acquisition devices (SCADA).

The protection system helps in preventing failure propagation by isolating faulty components. However, these devices rely on hard thresholds and local information, often ignoring system-level effects. This has lead to scenarios wherein a local mitigation in a subsystem could trigger a failure cascade, possibly resulting in a blackout (North American Electric Reliability Corporation, 2012). Moreover, power systems are going through transformational changes to account for distributed and decentralized generation (Jones, 2014), which has increased the stress on the aging legacy devices, thereby increasing the chances of failures (Di Fazio et al., 2013). The large size, inherent complexity, dynamic environments and software faults have deemed the manual diagnosis of faults infeasible and at the same time has increased the need for a robust and fast on-line management system that aids operators in failure diagnosis and prognosis.

There are number of challenges in creating an on-line management system. The foremost challenge is to create a fault model to analyze the progression of different faults in physical system while accounting for faulty and nominal behavior of components of cyber system. Creating a monolithic fault model for power systems will be difficult to manage and its more desirable to use a component based approach where a system fault model is composed by connecting smaller fault models. The other key challenge is imposed due to the geographical size of the power system that causes timing delays in failure progression between sub systems. The diagnosis approach should be able to account for these delays.

A number of model based and data driven approaches exists in the scientific literature (Ferreira et al., 2016). These approaches diagnose faults in components of both physical and cyber systems. However, none of the approaches model the causal relationship between the faults in one sub system to another (Hare, Shi, Gupta, & Bazzi, 2016). This requirement is essential for analyzing cascading scenarios in any domain. Our approach uses Temporal Causal Diagrams (TCD-s) to effectively model the dynamics of failures in both physical and cyber sub systems. We use a component based approach, where TCD models of different segments of power transmission network are connected together to represent a system level model that appropriately models the cascading failure progressions through out the system. TCD based diagnosis system is hierarchical in nature where a set of local level discrete diagnosers track the protection devices and feed their hypothesis to a system level reasoner which produces system level hypotheses.

The main contributions of this paper are as follows :-

1. Describing component TCD fault models.
2. Showcasing a systematic approach of generating component TCD models from system topology.
3. Synthesizing system TCD model by connecting individual fault models.
4. Discussing the timing complexity of fault model generation algorithm by using standard IEEE test systems.
5. Modifying the TCD reasoner hypothesis structure and reasoning algorithm to account for secondary faults.
6. Showing the efficacy of the diagnosis framework with the help of simulated case study involving a multi fault scenario.

2. RELATED RESEARCH

Fault diagnosis in power systems is an active area of research. 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, Yang, Hogg, Zhang, & Gao, 1994; Huang, 2002; Cardoso, Rolim, & Zurn, 2008; Jung, Liu, Hong, Gallanti, & Tornielli, 2001), **artificial neural network** (Cardoso, Rolim, & Zurn, 2004; Mahanty & Gupta, 2004; Thukaram, Khincha, & Vijaynarasimha, 2005; Bi et al., 2002) and **analytical model optimization** (Wu et al., 2005; Wen & Chang, 1997; He, Chiang, Li, & Zeng, 2009; Guo et al., 2010). In addition, approaches based on **petri networks** (Sun, Qin, & Song, 2004) and **cause-effect bayesian networks** (Chen, Liu, & Tsai, 2001; Chen, Tsai, & Lin, 2011; Guo et al., 2009; Chen, 2012; Yongli, Limin, & Jinling, 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, Akimoto, Kunugi, Fukui, & Fukui, 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 feed-forward perceptron with backward propagation is the most commonly used neural network model (MPNN) for failure diagnosis (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) neural networks with radial basis function (RBF) are presented. (Thukaram et al., 2005) discusses 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 (Wu et al., 2005; Wen & Chang, 1997; He et al., 2009). 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 (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). (Chen, 2012; Yongli et al., 2006) considered 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. (Guo et al., 2009) presents on-line alarm analyzer for diagnosing failure modes in the physical plant as well as in a relaying system based on a temporal causal network. But (Guo et al., 2009) does not take into account the operating modes and conditions of the system that influence the failure propagation.

TCD based diagnosis system is different from current methodologies where fault mitigation depends upon logic-based approach bound by hard thresholds and manual system level analysis. Moreover, these approaches are able to diagnose faults in physical and cyber sub-systems but cannot reason about the secondary physical faults induced in the system as a consequence of protection system (mis)operation. This is an important requirement for diagnosing cascading outages and predicting secondary and tertiary failure effects. Our approach can improve the situational awareness of system operators and help in preventing failures in large-scale systems such as Smart Electric Grids, by identifying impending secondary failures, thereby, increasing the system reliability and reducing the losses accrued due to power failures.

Rest of the paper is organized as follows, section 3 provides an overview of the relevant physical and cyber components in power systems. In the same section, we describe the component fault model of a section of transmission network, followed by discussion on the systematic approach of generating component and system level TCD models. Section 5 highlights hierarchical diagnosis framework by describing the behavior of low level diagnosers and TCD reasoner. It also lists an updated reasoning algorithm followed by a case study involving cascading failures in section 6 and concluding remarks in section 7.

3. TCD MODEL

A TCD model (Mahadevan, Dubey, Karsai, Srivastava, & Liu, 2014) is a behavior augmented TFPG model where faulty and non faulty behaviors of sensing, actuating and protection devices are explicitly modeled. Thus, a TCD model captures:

- Failure modes, discrepancies and failure propagation across the physical system including sensors, actuators and protection devices.
- The nominal operation of the protection system in terms of the observed effects, the control action and its application on the modes that control the state of the actuators.
- The failure modes associated with protection system and their effect on the operating modes of the system and thereby altering the failure propagation in physical plat. These failures include: 1) Missed detection faults: faults in protection system when it does not act and 2) Spurious detection faults: faults in protection system where it acts unnecessarily.

A system-level TCD model is hierarchical and composed of component fault models. A component model includes Timed Failure Propagation Graphs (TFPG) and/ or Timed Triggered Automata (TTA) models. The TCD model captures the interactions between the TFPG and TTA models both within the component, as well as across component boundaries. The interactions between the TFPG and TTA models are captured implicitly through the state changes in TTA models as a response to changes in observed and hypothetical states of discrepancy and failure mode nodes in TFPG model. The state transitions associated with TTA models leads to system mode change that enable or disable failure propagation edges in TFPG model. These interactions extends through the boundaries of a component i.e activation of a failure mode node in a TFPG model of one component can influence state transition in TTA models of other components and vice-a-versa. Similarly, TTA model of one component can influence change in states of TTA models of the same component as well as the others. Figure 1 shows an abstract system TCD model composed of two sub-system level models that are composed of two component fault models. The TFPG models of different components can be explicitly connected to model physical failure propagation amongst TCD components models.

Appendix provides a brief overview of the TCD modeling formalism, and for detailed description please refer to (Chhokra, Dubey, Mahadevan, & Karsai, 2017). The following sections give an overview of power transmission system, describe component TCD model of a part of the system and discuss the fault model generation algorithm in detail.

4. POWER TRANSMISSION NETWORK

Figure 2 shows a segment of power transmission network with two transmission lines (*TL1*, *TL2*) and three buses (*B1*, *B2*, *B3*). Every transmission line is protected by a pair of protection assemblies attached to it's ends. A protection assembly is a collection of current and potential transformers (sensors), protection relays (controllers) and breakers (actuators) that help in arresting failures by isolating the faulty component from the system. These protection assemblies are installed at the sub stations labeled as *SS1*, *SS2* and *SS3*. Additionally, protection assemblies of the nearby transmission lines act as backup or secondary protection devices. For instance, the relays contained inside the protection assemblies, *PA1* and *PA2*, act as primary source of protection against phase to phase, phase to ground faults in *TL1* while protection assembly *PA4* acts as backup.

Distance relays (E. O. Schweitzer, Kasztenny, Guzmán, Skendzic, & Mynam, 2014) detect fault conditions by inspect-

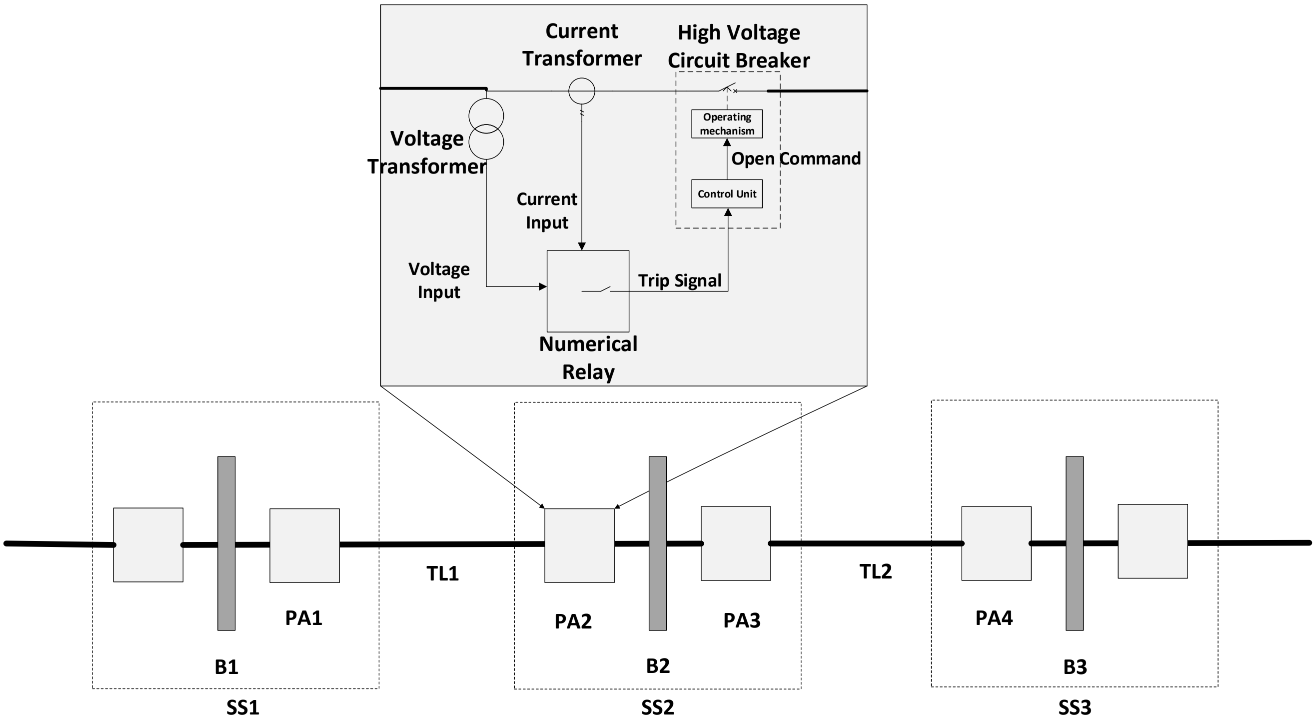

Figure 2. A segment of power transmission network

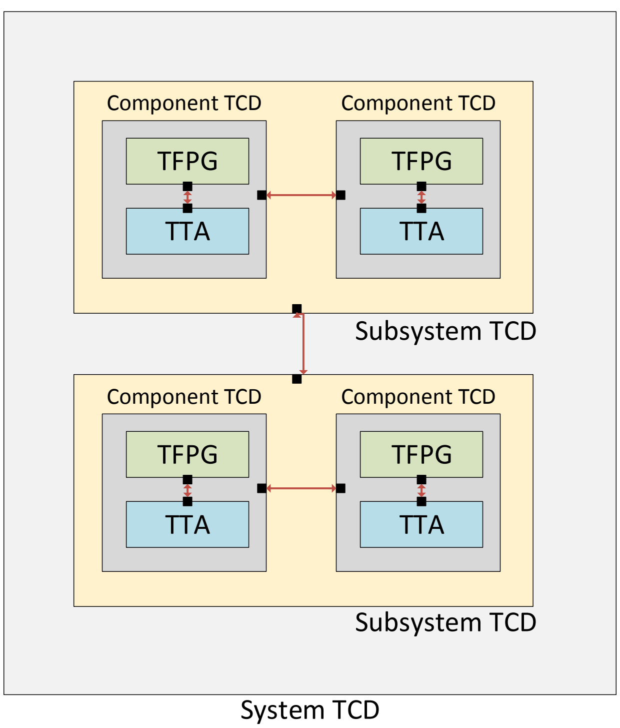

Figure 1. A TCD model of a system consists of interacting subsystems containing components, where each component consists of an interacting TFPG and TTA models.

ing the apparent impedance (V/I). When a phase to phase or phase to ground fault is introduced in a transmission line, the current flowing through the conductor increases and voltage at the bus terminals drops resulting in decrease in impedance seen by the distance relay. Distance relays depending upon the value of the impedance detected conclude the location of the fault. Typically, distance relays are configured to operate in three zones. A distance relay infers a zone 1 fault when the measured impedance is less than 0.8 times the impedance of the transmission line. In zone 1, distance relay acts as a primary protection element and instantly commands a breaker to trip. A zone 2 fault is detected when the measured impedance is greater than 0.8 but less than 1.25 times of the transmission line. In this zone, distance relay waits for 0.05 - 0.1 secs before sending the trip signal. A zone 3 fault forces the apparent impedance seen by the relay to be 1.25 - 2 times the impedance of the transmission line and the relay waits for 1-1.5 secs before sending a trip signal to the breaker. Under zone 2 and 3 fault conditions, a distance relay acts as a backup protection element. (Chhokra et al., 2017) presents TCD fault model involving faults in transmission lines by utilizing the alarms signaled by the distance relays and the estimated state of the breakers.

However, this fault model is incomplete as it does not show how failure propagates from one TCD model of transmission line to another. Typically, in power systems, the secondary effects of isolating faults in physical components are bus voltage collapse, branch overloads and loss of synchronism. These secondary effects, if not dealt with, can cause serious damage, thereby injecting secondary physical faults. Moreover, control actions taken by line operators to remove these secondary effects have worsen the situation in the past (North American Electric Reliability Corporation, 2012) as these actions are solely based on local information, and have caused same secondary effects in other parts of the system, causing a domino effect.

For instance, increased power flowing through the conductor can damage the insulation. To avoid any permanent damage to the conductor, fuses or over-current relays are used which opens the breaker to stop the flow of the power through the transmission line. The opening of breaker causes change in the flow of power and may over load some other part of the system and causes overload protection to engage again. Thus alarms that signal anomalies related to secondary effects i.e overloads form a causal link between failure propagation among different TCD models[1].

4.1. Component TCD Model

Component TCD model for power transmission networks include TFPG model of a transmission line and TTA models of its respective protection devices (controllers and sensors) and breakers (actuators). Figure 3 shows an fault model of transmission line, that contains an embedded TFPG model and behavioral models of distance relays, over-current relays and breakers that serve primary and secondary protection elements. The TFPG model shows the failure signature of physical faults associated with transmission line and also captures the effect of isolating faults. The behavior models display the working of relays and breakers in faulty and non faulty conditions. The failure propagation depends upon the operating modes which is a function of state of the breakers. The behavioral models are hand crafted by leveraging information from the user manuals of the discrete devices while TFPG model is automatically synthesized based upon the location of the physical element and its respective protection devices. The following sub sections describe the different parts of the TCD model:

4.1.1. Distance Relay Behavioral Model

Figure 3 shows an abstract time triggered automaton of a distance relay configured to operate in 3 zones of protection. We have considered 4 detection faults, *F_de1*, *F_de2_z1*, *F_de2_z2* and *F_de2_z3* in this paper. Fault, *F_de1*, is a missed detection fault that forces the relay to skip the detection of fault and *F_de2_z1*, *F_de2_z2*, *F_de2_z3* are spurious detection faults associated with zone 1, 2, 3 fault conditions respectively. Relay produces *Z1*, *Z2*, *Z3* alarms to signal the presence of zone 1, 2, 3 fault conditions respectively. For more description of the distance relay behavior, please refer to (Chhokra et al., 2017).

4.1.2. Breaker Behavioral Model

It is important to consider faults in the breaker behavior as it's faulty operation has contributed towards blackouts in the past (North American Electric Reliability Corporation, 2012). Figure 3 shows a simplified time triggered automaton of a single phase circuit breaker. The automaton describes the operation of breaker in nominal mode and in the presence of stuck open and stuck close faults. The stuck open fault forces the breaker to remain in `open` state while the stuck close fault makes sure the breaker never transitions from `close` to `open` state. The breaker responds to commands received by relays, *cmd_open*, *cmd_close* and produces events *st_open* and *st_close* to signify successful state transition from `open` to `close` and vice-versa. For more information about the breaker behavior, please refer to (Chhokra et al., 2017).

4.1.3. Over Current Relay Behavioral Model

The objective of the overload protection is to prevent damage to a physical component in an electric circuit when the component is subjected to a prolonged overload conditions. Overload protection can be achieved using a variety of means: fuses, low-voltage (LV) circuit breakers like miniature circuit breakers (MCBs) and molded-case circuit breakers (MCCBs), over-current relays used in conjunction with high-voltage (HV) circuit breakers, etc.

Figure 3 shows time triggered automaton of a single step time definite over-current relay. The automaton consists of two failure modes *F_de1* and *F_de2*. *F_de1* models missed detection fault and *F_de2* represent spurious detection fault. Figure also lists 3 different failure mode constraints, $\delta(F\_de1)$, $\delta(F\_de2)$ and $\neg\ \delta(F\_de1) \wedge \neg\ \delta(F\_de2)$. The presence of failure modes *F_de1* and *F_de2* enables the first two failure mode constraints while the third constraint evaluates to true only if none of the detection faults are present. Four different events are used to model the over-current relay behavior. The event labeled as *E1* is an un-observable that represents increase in current beyond permissible threshold (150% - 300% of the maximum load current). The observable event *OR* is an alarm produced by the relay to signal overload, and *cmd_open* marks the event when the relay sends a trip signal to the breaker.

The state machine consists of 6 locations with `idle` being the initial location. Every R seconds (1 milliseconds), the relay looks for event *E1* and evaluate failure mode constraints. If *E1* is present and both the failure modes are absent then it transitions to `waiting` location. In `waiting` state, it waits for a pre-defined amount of time (200 secs) ensured by the instantaneous timing constraint, [*WT*], and transitions to `chk`

[1]This paper only cover overloads as secondary effects but can be easily extended to include others

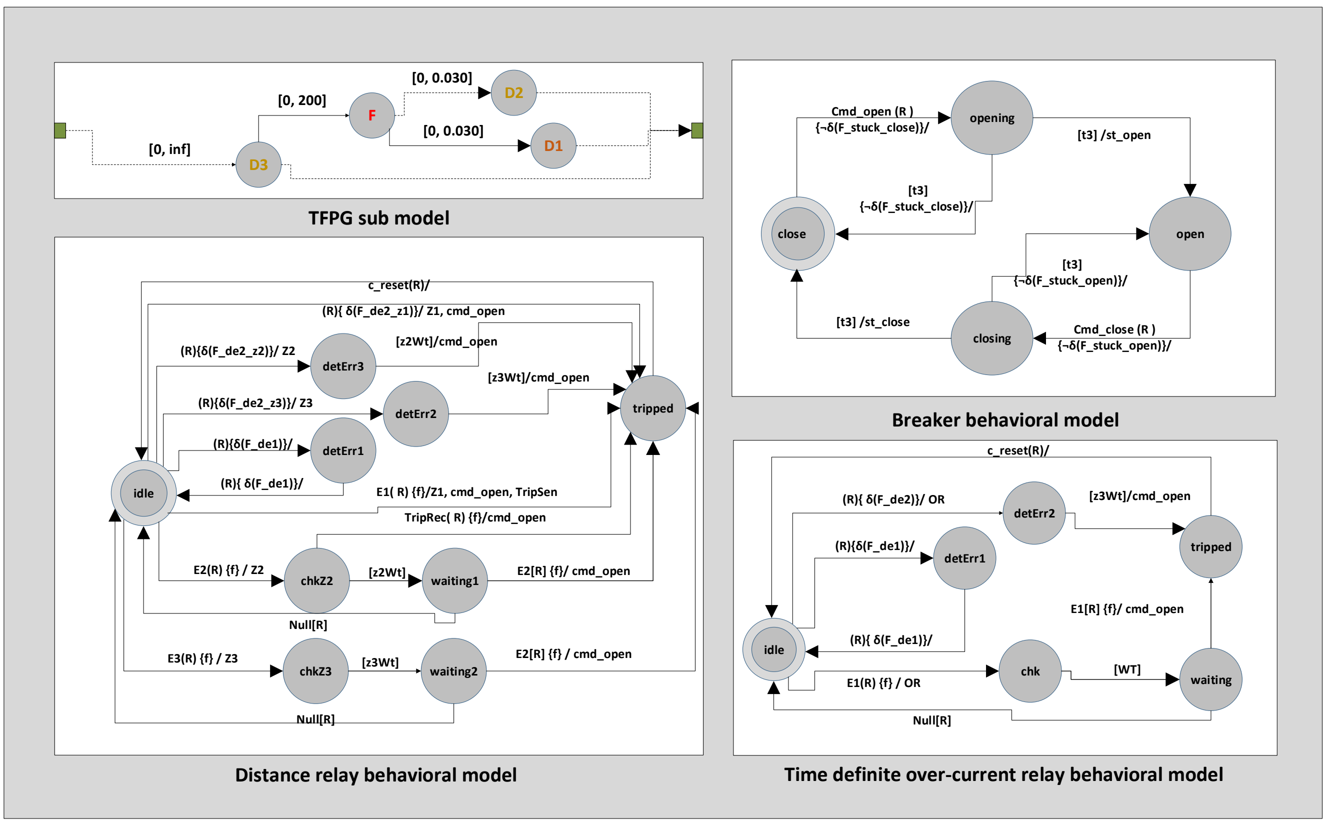

Figure 3. **Top-Left Figure**: TFPG model to represent failure effects and their propagation as a result of physical fault and corrective actions of isolating physical faults. **Top-Right Figure**: Behavior model of a breaker while taking into account stuck open and close faults. *R*, is the sampling time and *t3*, models time to change breaker states. **Bottom-Left Figure**: Behavioral model of a distance relay with 3 zones of protection operating at a sampling rate, *R* with zone 2 and 3 wait times represented by parameters, *z2wt* and *z3wt*, respectively. **Bottom-Right Figure**: Behavioral model of an over-current relay with a single step.

location. In `chk` location, it again checks for the overload condition and presence of failure modes. If the overload condition still exist it transitions to `tripped` location. The deviation from the nominal operation is caused if either $\delta(F\_de1)$ or $\delta(F\_de2)$ evaluates to true at any time. For instance, if in `idle` state and constraint $\delta(F\_de1)$ evaluates to true (implying *F_de1* is present) then automaton moves to `detError1` location and stays there until *F_de1* disappears. Similarly, if *F_de2* is present then automaton transitions to `detError2` location and jumps to `tripped` location without checking *E1*.

4.1.4. TFPG Model

Figure 3 shows a generic TFPG model for a transmission line. It consists of 4 different sets of nodes described as follows :

- **F**: It is a set of failure mode nodes that represent physical faults such as phase to phase and phase to ground.
- **D1**: It is a set of observable discrepancy nodes. These discrepancy nodes represent the reduction in impedance due to fault $f \in \mathbf{F}$. These discrepancies are signaled by zone 1 or 2 alarms (*Z1, Z2*), triggered by primary protection devices. Since there are two primary protection device per transmission line, the size of this set is 2. For instance, the set *D1* in a TFPG model for line TL1 contains two discrepancies, (*d_TL1_PA1_DR*, *d_TL1_PA2_DR*) where *d*, *TLn*, *PAk* and *DR* denote the type of TFPG node (discrepancy), transmission line label, protection assembly label and component type in the protection assembly (Distance Relay).
- **D2**: It is a set of observable discrepancy nodes similar to D1 but are signaled by zone 2 or zone 3 alarms produced by backup protection elements. The size of this set depends upon the number of backup protection devices. For example, *D2* in TFPG model of TL1 contains one discrepancy, *d_TL1_PA4_DR*.
- **D3**: It is a set of observable discrepancy nodes that imply the increase in current flowing through the transmission line. The discrepancy is signaled by alarms generated by the over-current relays. The size of the set is 1.

The state of the breakers in primary protection assemblies constraints the failure propagation from nodes in *F* to discrepancy nodes in *D1*. For instance, in TFPG model of TL1, the failure effect can only reach discrepancy *d_TL1_PA1_DR*, if the breaker, *PA1_BR* (in protection assembly *PA1*) is in `close` state. The time taken by the failure effect to propagate from nodes in *F* to *D1* is equal to the time taken by the respective relays to detect the fault. The fault detection depends upon the sampling time of the relay and the frequency of the microprocessor (E. Schweitzer, Fleming, Lee, Anderson, et al., 1997). Fast numerical relays with high sampling rates (64 samples per cycle) can accomplish sub cycle[2] fault detection while relays with low sampling rate (2 samples per cycle) can take upto 2 cycles for detecting fault conditions (Venkatesh & Swarup, 2012). We consider 30 milliseconds to be upper threshold on the failure propagation time interval, as shown in figure. The edges between F and D2 have same operating and timing constraints with an exception of being uncertain, represented by dotted line in Figure 3. The uncertainty arises due to the fact single failure node is representing fault through out the length of the transmission line.

The failure edges between nodes in *D3* and *F* model the sagging or loss of insulation around the conductor that injects secondary failure in the transmission line. The time duration for the effect to propagate depends upon the thermal characteristics of the transmission line and is of the order of minutes (200 secs in our implementation). The operating conditions for this effect to reach nodes in D3 also depends upon the state of breakers along the path. The outgoing edges from nodes, (*D1*, *D2*, *D3*) in TCD model of a line to *D3* in other TCD models capture the effect of corrective actions, thereby, accounting for cascades.

4.2. Generation of TCD Model

Due to the large size of power systems, its advantageous to automate the process of generating component TCD models and synthesize a system TCD model by appropriately connecting them. Each component fault model contains replicas of user created behavioral models and a TFPG model. There are two keys requirements for generating system wide TCD model :-

1. To identify primary and secondary distance relays that are responsible for detecting physical fault in all lines.
2. To identify probable set of transmission lines that can be overloaded by the control actions of the protection devices associated with a given line.

A transmission line network can be considered as a connected graph with nodes of types, {*Generator*, *Line*, *Transformer*, *Load*, *Bus*, *Protection Assembly*} and edges between them implies power flow. The graph is stored as map, $adjacencyList$, where keys are node labels and value is an adjacency list. The underlying algorithm of finding the primary and secondary protection relays is based on recursive graph traversal as listed in Algorithm 1. The algorithm update two maps *PPE* and *SPE*, which store primary and secondary protection elements associated with a branch (line or transformer) respectively. The keys are branch labels and value associated with a key is set of protection element labels. The input parameters[3] of the algorithm include

1. *node* : Starting node object.
2. *visited* : Set of all visited nodes at a given iteration.
3. *PA_label* : The label of starting node.

[2]One cycle equals 16.67 milliseconds

[3]The parameters, *PA_label*, *Bus_label* and *max_imp* do not change.

4. *Bus_label* :The label of the bus to which *node* is attached. This parameter is required in order to avoid traversing the graph in the reverse direction.
5. *imp* : Cumulative impedance at each iteration, initially the value is 0.
6. *max_imp* : Impedance reach of the highest configured zone, i.e. zone 3.
7. *flag* : Parameter to reflect that a branch has been identified for which *PA_label* acts as primary protection. Initial value is False.

The routine *iterateGraph* is invoked for every protection assembly and recursively traverses the graph until *imp* reaches the threshold of *max_imp* (line 5). Depending upon the type of the node, *imp* is updated (line 3-4). The relay, *PA_label* is considered as primary relay of the current node if it matches the following three conditions

- The type of node is either *Line* or *Transformer*. (line 3)
- *imp* is less than max_imp. (line 4)
- Boolean variable, *Flag*, is False. (line 6)

If only the first condition evaluates to true then SPE is updated (line 9, 15). The routine calls itself for every child node (line 11-12, 19-21, 24-26) except

1. If the current node type is either *Line* or *Transformer* and the condition $imp < max\_imp$ evaluates to false i.e max zone reach has reached. (line 4)
2. If the current node type is *Bus* and node label is *Bus_label*. This condition restricts traversal in the reverse direction. (line 18)

There is one more map, CB where key value pair relates to a branch outage and a set of probable branch outages that can happen in future. Ideally, this map requires very large number of simulations to capture every cascading scenario in all possible topology configurations (exponential in the size of number of branches). We use a hybrid, off-line and on-line approach to find all probable overloads of a given branch outage. In off-line mode, using graph theoretic appraoch, we identify a set of branches that can never be overloaded as a result of the given branch outage and at run time (on-line mode), this set is further reduced by performing on demand load flow calculation using steady state power flow solver, OpenDSS (Dugan, 2016). The underlying graph theoretic algorithm that updates the cascade map is shown in 2. The algorithm is invoked for every branch and removes the branches that cannot be overloaded. It recursively traverses each node outwards from the given branch until its visits a node with a degree more than two (line 4).

The generation algorithms are based on exhaustive search that has exponential timing complexity. However, the graph traversal is restricted by zone reach and degree of the components. These constraints make the algorithms to have polynomial time complexity. We performed the timing analysis

Algorithm 1 Algorithm for updating PPE and SPE system maps: **iterateGraph**(node, visited, PA_label, Bus_label, imp, max_imp, Flag)

```
 1: if node ∉ visited then
 2:     visited ← visited ∪ node
 3:     if node.type ∈ {'Line', 'Transformer'} then
 4:         imp ← imp + node.Impedance
 5:         if imp < max_imp then
 6:             if ¬ Flag then
 7:                 PPE[node.label] ← PPE[node.label] ∪ {PA_label}
 8:             else
 9:                 SPE[node.label] ← SPE[node.label] ∪ {PA_label}
10:             end if
11:             for all n ∈ adjacencyList[node] do
12:                 iterateGraph(node, visited, PA_label, Bus_label, imp,
                    max_imp)
13:             end for
14:         else
15:             SPE[node.label] ← SPE[node.label] ∪ {PA_label}
16:         end if
17:     else if node.type = 'Bus' then
18:         if node.label ≠ Bus_label then
19:             for all n ∈ adjacencyList[node] do
20:                 iterateGraph(node, visited, PA_label, Bus_label, imp,
                    max_imp)
21:             end for
22:         end if
23:     else
24:         for all n ∈ adjacencyList[node] do
25:             iterateGraph(node, visited, PA_label, Bus_label, imp,
                max_imp)
26:         end for
27:     end if
28: end if
```

Algorithm 2 Algorithm for updating CB system map: **iterateGraph**(node, visited, Branch_label)

```
 1: if node ∉ visited then
 2:     visited ← visited ∪ node
 3:     neighbors ← adjacencyList[node]
 4:     if neighbors.size ≤ 2 then
 5:         for all n ∈ neighbors do
 6:             if n.type ∈ {'Line', 'Transformer'} then
 7:                 CB[Branch_label] ← CB[Branch_label] \ n.label
 8:             end if
 9:             iterateGraph(n, visited, Branch_label)
10:         end for
11:     end if
12: end if
```

by generating fault models for standard IEEE test systems[4] of small, medium and larges sizes. Table 1 shows the parameters of the test topology and the generated fault model. The last column shows the time taken for model generation which includes, parsing of IEEE common data format (Group, 1973), creating a graph in memory, generating fault model and serializing the fault model into a xml file.

5. TCD DIAGNOSIS FRAMEWORK

TCD diagnosis framework employs a hierarchical, discrete event based reasoning methodology. Events related to zone detection alarms, breaker commands, breaker state change messages are consumed by lower level diagnosers, called *Observers*. The output of these Observers are passed to graph based TCD reasoner which produces hypotheses consistent

[4]https://www2.ee.washington.edu/research/pstca/

Table 1. TCD Models for IEEE Test Systems

Topology Name	Topology Parameters		TCD Model Parameters					
	Nodes	Branches	Failure Modes	Discrepancies	Alarms	Modes (2^n)	Edges	Generation Time (sec)
WSCC 9 Bus System	24	9	144	98	189	18	476	0.48
IEEE 14 Bus System	50	20	320	339	420	40	2059	1.65
IEEE 30 Bus System	98	41	656	883	861	82	7853	9.36
IEEE 118 Bus System	449	186	2976	5892	3906	372	145368	93.46
IEEE 300 Bus System	978	411	6576	14038	8631	822	691996	1968.08

with TCD model of the system. The following sub sections give brief overview of Observers and TCD reasoner.

5.1. Observers

Observers are discrete, finite state machines that consume events produced by their respective tracked devices. There exists a number of approaches for generating discrete diagnosers for dynamic systems based on (Tripakis, 2002) and (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). Figure 4 shows the observer models for the protection relays and breakers. These state machines accepts observable events, such as fault detection alarms and trip commands, to estimate the presence of faults in both physical and cyber components. These observers produce their hypotheses in the form of observable events that are passed to TCD reasoner. Following subsections give more detail about their operation.

5.1.1. Observer: Distance Relay

The time triggered automaton model of a distance relay observer can be seen in Figure 4. The state machine has 8 locations with `idle` being the initial state. The observer machine consumes the observable zone alarms (*Z1*, *Z2*, *Z3*), commands sent to breaker (*cmd_open*) and reset events. It produces *h_Z1*, *h_Z2*, *h_Z3* to indicate or confirm the presence of zone 1, 2, 3 faults. The observer also produces *h_Z1*$'$, *h_Z2*$'$and *h_Z3*$'$to indicate absence of zone 1, 2, 3 fault conditions. $, t3, z2wt, z3wt \in \mathbb{R}_+$, are the parameters of relay observer that model propagation delay, zone 2 and 3 wait times respectively. For detailed information of observer behavior, please refer to (Chhokra et al., 2017).

5.1.2. Observer: Breaker

The breaker observer model is also shown in Figure 4. It consists of 4 states labeled as `open`, `close`, `opening` and `closing` and correlate directly to the 4 states of the breaker automaton. The observer consumes *cmd_open*, *cmd_close* *st_open* and *st_close*and produces *h_open*, *h_close* to signal state change from close to open and vice-versa respectively. The observer also emits (*h_stuck_open*; *h_stuck_open*$'$) and (*h_stuck_close*; *h_stuck_close*$'$) to indicate the presence and absence of stuck open and close faults. $t4 \in \mathbb{R}_+$ is a parameter of the breaker observer that models the delay associated with state transition due to it's mechanical nature. The detailed working of breaker observer model is presented in (Chhokra et al., 2017).

5.1.3. Observer: Over-current Relay

The observer model tracking the behavior of over current relay is shown in Figure 4. The automaton consists of 4 states, `idle`, `chk`, `waiting` and `tripped` with `idle` being the initial location. The observer consumes *OR*, *cmd_open* and *c_reset* events from the relay and generates *h_OR*, *h_OR*$'$to signal the presence and absence of overload conditions. While in the `idle` state, the automaton periodically checks for the *OR* event. After detecting the overload conditions, the observer generates *h_OR* and jumps to *chk* location. After waiting for $WT \in \mathbb{R}_+$, the state machine transitions to `waiting` state. While in `waiting` state, observer checks for the *cmd_open* even. If *cmd_open* event is received with in *t3* seconds, then state machine moves to the `tripped` state otherwise transitions to `idle` state while emitting *h_OR*$'$to signal the overload has disappeared. *t3* is a parameter of the observer machine that models propagation delay.

5.2. TCD Reasoner

The TCD reasoner relies on the fault propagation graph and the output of various observers to hypothesize about the anomalies observed in the system. In order to relate to the alarms generated by observers with the failure graph few modifications are performed. The alarms signaled by relays are replaced by their corresponding observers i.e. *Zn* is replaced by *h_Zn*. The reasoner attempts to explain the observations in terms of consistency relationship between the states of the nodes and edges in the fault propagation graph. The states of a node in a fault 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.
- An *Observed state* is the same as the *Physical state*, but only defined only for observable nodes.
- A *Hypothetical state* is an estimate of the node's physical state and the time since the last state change happened by the TCD reasoner.

Every reasoner hypothesis, $h \in HSet_t$ consists of a map, $HNode_t$ that associates to every node in the failure graph an evaluation, (*ON*, *OFF*) and time estimate (t_1, t_2). The time

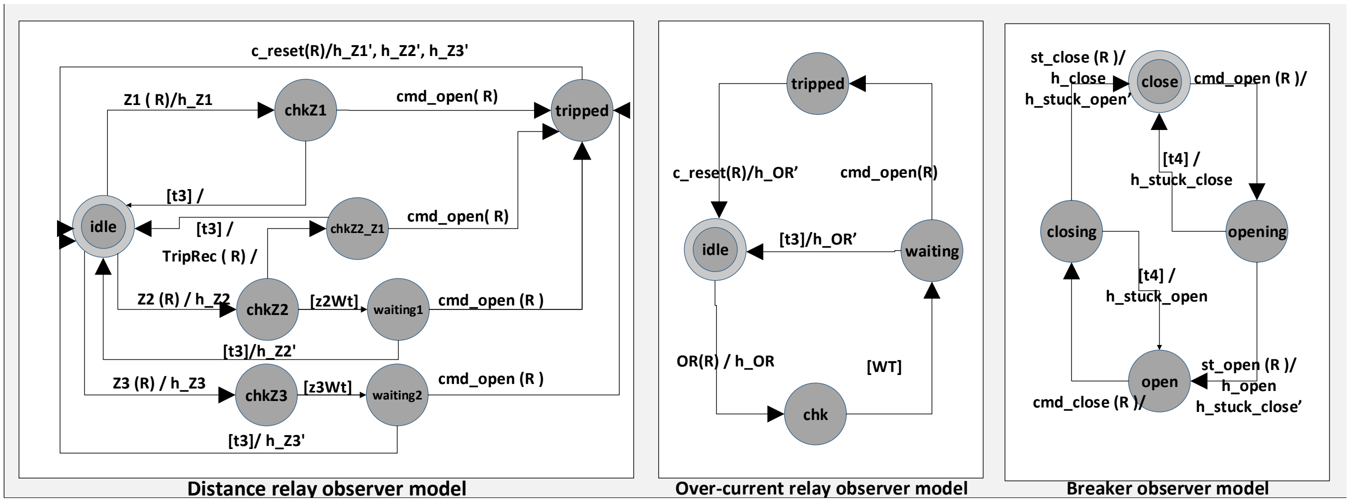

Figure 4. Protection System Observer Models

estimate, (t_1, t_2) denotes the earliest and latest time estimates for the state changes of node v i.e. from ON to OFF or vice-a versa. The structure of a hypothesis is described as follows:

Hypothesis is a tuple, where elements are related based on temporal consistency. Formally, hypothesis h=\{*F*, S_{cyber}, $F_{physical}$, *C, I, M, E, ES* \} where:

- ***F*** $\subseteq F_{physical}$ is a subset of physical failure modes projected by the hypothesis.
- $S_{cyber} \subseteq$ ***F***$\boldsymbol{F_{cyber}}$ is a set of faults active in the system. These faults are related to detection faults and stuck faults in relays and breakers.
- $S_{physical} \subseteq \boldsymbol{F_{cyber}}$ is a set of secondary physical faults caused due faults in F.
- ***C*** $\subseteq D_{physical}$ is the set of discrepancies that are consistent with the hypothesis h, where $D_{physical}$ is the set of physical discrepancies related with faults in $F \cup S_{physical} \subseteq F_{physical}$. 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* is called primary if the fault propagation linking the discrepancy, *d*, is certain otherwise its termed as secondary.
- ***E*** $\subseteq D_{physical}$ is the set of discrepancies which are expected to be activated in the future according to h. This set is also partitioned into *E1* and *E2* that contain primary and secondary discrepancies, respectively.
- ***ES*** $\subseteq F_{physical}$ is the set of expected secondary failure modes to be activated in the future as per h.
- ***M*** $\subseteq D_{physical}$ is the set of discrepancies that are missing according to the hypothesis h 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.
- ***I*** $\subseteq D_{physical}$ 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.

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

The quality of the generated hypotheses are measured based on four metrics 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

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

- ***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 (i.e., it follows the parsimony principle).

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. (Chhokra et al., 2017) describes the underlying algorithms to handle events but the monitored discrepancy state change algorithm has to be extended to include the effect of secondary physical failure modes which is shown in 3.

Algorithm 3 *HandleDsicrepancyStateChnageEvent(e,m)*: Algorithm for handling discrepancy state change event

```
Input: (d, t), m
isExplained ← FALSE
for all h ∈ HSet_t do
    if d ∈ D_cyber then
        UpdateS_cyberSet(h,d)
        isExplained ← TRUE
        continue        ▷increment h to next hypothesis in HSet
    end if
    if TConsis_t(h, d) then
        isExplained ← TRUE
        UpdateHNodeMap(h,d)
        UpdateConsistentSet(h,d)
        UpdateExpectedSet(h,d)
        AddTimeOutEvents(h,d,t′)
    else
        UpdateInconsistentSet(h,d)
    end if
end for
if ¬isExplained then
    H_new ← CreateNewHypothesis(d, t, m)
    for all h′∈ H_new do
        for all h ∈ HSet do
            h″← h        ▷Temporary placeholder
            if h″.ES ∩ h′.F ≠ ∅ then
                h″.S_physical ← h″_physical ∪ h′.F
            else
                h″.F ← h″.F ∪ h.′.F
            end if
            MergeConsistentSet(h″, h′)
            MergeExpectedSet(h″, h′)
            MergeS_cyberSet(h″, h′)
            UpdateInconsistentSet(h″, d)
            Addhypothesis(HSet, h″)
        end for
    end for
end if
```

6. CASE STUDY

We validated the TCD fault model and diagnosis framework with the help of a standard WSCC 9 Bus system[5]. WSCC 9 Bus system is a simple approximation of the Western System Coordinating Council electrical network. It consists of 3 generators, 9 Buses, 6 transmission lines and 3 loads as shown in Figure 5.

The test system is modeled in Simulink[6], where Simscape Power Systems[7] toolbox provides models of physical components and Stateflow[8] charts are used to create time triggered automatons of protection system. Different scenarios are simulated in Simulink and their outputs are serialized into XML files. These XML files are parsed by python based TCD diagnosis prototype.

Tables 2,3 list the timed events produced by the protection system along with output of various observers and TCD reasoner for a blackout causing multi fault scenario. The cascading sequence initiates with a 3 phase to ground fault in line, **TL_B7_B8** followed by a secondary fault in the adjacent line, **TL_B8_B9**. The secondary fault is caused due line sagging and coming in contact with nearby vegetation. In the end, overload protection relays[9] isolate lines **TL_B5_B7** and **TL_B4_B5** causing more than two-third of the total load to be de-energized (Blackout). TCD diagnosis system correctly diagnose the cascading outages and lists a total of 4 hypothesis. Hypothesis H3, perfectly explains the system events with 100% plausibility and least number of estimated component failures.

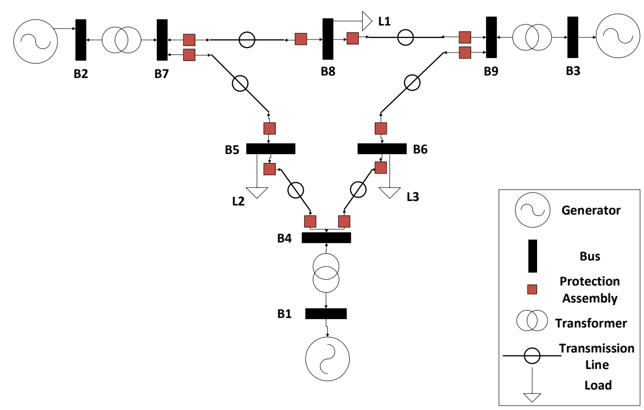

Figure 5. Test System: WSCC 9 Bus System

7. CONCLUSION

In this paper we presented a component based approach to model cascading outages using TCD formalism. We showcased the results of the generation algorithm. We also described the TCD diagnosis framework by discussing in detail the timed discrete models of protection devices and showed the efficacy of the TCD reasoning scheme by accurately diagnosing primary and secondary failures in a multi fault scenario in WSCC 9 Bus system. As a part of our future work , we would like to extend TCD diagnosis framework by adding prognostics and cascade mitigation capabilities.

ACKNOWLEDGMENTS

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, and Saqib Hasan for their help and discussions related to the work presented here.

[5] http://icseg.iti.illinois.edu/wscc-9-bus-system/
[6] https://www.mathworks.com/products/simulink.html
[7] https://www.mathworks.com/products/simpower.html
[8] https://www.mathworks.com/products/stateflow.html

[9] To reduce the simulation time, the wait time is reduced from 200 secs to 10 secs and TCD model is updated accordingly

Table 2. System Events

Time Stamps (secs)	Cyber-Physical System Events	Observer Events	Reasoner Hypotheses
1	A 3 phase to ground fault is injected in transmission line, **TL_B7_B8**		
1.001	**PA_B7_TL_B7_B8_DR**, **PA_B8_TL_B7_B8_DR** detect zone 1 fault conditions, respective state machines transition to `tripped` location after producing *Z1* and *cmd_open* events. **PA_B9_TL_B8_B9_DR** detects zone 3 fault conditions, and transitions to `chkZ3` state after emitting *Z3* event. Breakers, **PA_B8_TL_B7_B8_BR** and **PA_B7_TL_B7_B8_BR** acknowledge the relay command and transition to `opening` state.	The observers associated with **PA_B7_TL_B7_B8_DR** and **PA_B8_TL_B7_B8_DR**, first transition to `chkZ1` state and then jump to `tripped` state. They produce *h_Z1* event. The observer tracking the behavior of relay, **PA_B9_TL_B8_B9_DR**, transitions to `chkZ2` location and emits *h_Z3* event. The breaker observers move to `opening` state after detecting *cmd_open* event.	The TCD reasoner generates two hypotheses, H0 and H1, where H1 hypothesizes fault in line, **TL_B7_B8** with 75% robustness, 100% plausibility and failure count of 1. The second hypothesis, H0, blames distance relays for incorrectly detecting faults (spurious detection fault). The failure count for H0 is 3. According to law of parsimony H1 is more probable than H0.
1.051	Breakers, **PA_B7_TL_B7_B8_BR** and **PA_B8_TL_B7_B8_BR** change their state to `open` and produce *st_open* events.	On detecting state change events, the corresponding observers also transition to `open` state and produce mode change (*h_open*) and alarm state change (*h_stuck_close'*) events.	The mode change event causes the robustness of H1 to decrease from 75% to 50% as H1 expects over-current relay alarms from protection assemblies of nearby transmission lines.
1.052	The over-current relays associated with lines, **TL_B5_B7**, **TL_B8_B9** and **TL_B4_B6**, produce *OR* alarms to signal overload conditions and update their state to `chk`.	The observers tracking the behavior of these over-current relays transition to `chk` state after detecting *OR* event and produce *h_OR* to conclude overloading conditions.	Increase in robustness of H1 hypothesis, from 50% to 100% and increase in the number of spurious detection faults estimated by H0, from 3 to 9.
2.001	The zone 3 wait time expires for relay, **PA_B9_TL_B8_B9_DR**. The state machine transitions to `waiting2` state.	Associated observer also updates it's state to `waiting2`.	
2.002	Since the fault in line **TL_B7_B8** has already been isolated, the relay, **PA_B9_TL_B8_B9_DR** moves back to `idle` state.		
2.031		*t3* wait time expires for the observer tracking **PA_B9_TL_B8_B9_DR**. The observer does not detect *cmd_open* event and conclude the absence of zone 3 fault. It produces *h_OR'* alarm and moves back to `idle` state.	The number of spurious detection fault reduces to 8 in H0.
5.000	Due to increased current flowing through the conductor, the transmission line, **TL_B8_B9**, sags and comes in contact with the nearby vegetation.		
5.001	Relays, **PA_B8_TL_B8_B9_DR**, **PA_B9_TL_B8_B9_DR** detect zone 1 fault conditions. These relay transitions to `tripped` location and produce *Z1* and *cmd_open* events. Breakers, **PA_B8_TL_B8_B9_BR**, **PA_B9_TL_B8_B9_BR** acknowledge the relay command and transition to `opening` state.	The observers associated with **PA_B7_TL_B7_B8_DR**, **PA_B8_TL_B7_B8_DR**, first transition to `chkZ1` state and then jumps to `tripped` state. They produce *h_Z1* to conclude presence of zone 1 fault conditions. The breaker observers move to `opening` state after detecting *cmd_open* event.	Number of hypotheses increases to 4. H0: Failure count increases from 8 to 10 H1: Alarms added to inconsistent set, Robustness, Plausibility and Failure count are 75%, 75% and 3 respectively. H2(New Hypothesis): Lists physical fault in **TL_B8_B9** as primary fault and active alarms related to fault **TL_B7_B8** are added to inconsistent set. Robustness, Plausibility and Failure count are 25%, 28.57% and 8 respectively. H3(New Hypothesis):Extension of H1, lists fault in **TL_B8_B9** as a secondary fault. Robustness, Plausibility and Failure count are 100%, 100% and 2 respectively.
5.051	Breakers, **PA_B8_TL_B8_B9_BR** and **PA_B8_TL_B8_B9_BR** change their state to `open` and produce *st_open* events.	On detecting state change events, the corresponding observers also transition to `open` state and produce mode change (*h_open*) and alarm state change (*h_stuck_close'*) events.	

A transmission line is labeled according to the buses that are connected at it's two ends. For instance, **TL_Bi_Bj** is a transmission line connected between two buses Bi, Bj, such that $i, j \in \mathbb{Z}$, $i \neq j$ and TL implies the component type. Similarly, a protection assembly is named as per the labels of the adjacent bus and the transmission line. For instance, **PA_Bi_TL_Bi_Bj** is a protection assembly connected between bus **Bi** and transmission line **TL_Bi_Bj**. Distance relays, over-current relays and breakers are labeled by appending *_DR*, *_OR* and *_BR* to the label of their respective protection assemblies.

Table 3. System Events - Contd.

Time Stamps (secs)	Cyber-Physical System Events	Observer Events	Reasoner Hypotheses
11.051	Wait time, *WT* of over-current relays associated with lines, **TL_B4_B6**, **TL_B5_B7**, **TL_B8_B9** expires. The relays move from `chk` to `waiting` state.	Wait time for corresponding observers expires and their states are updated to `waiting`	
11.052	Overloading condition persists only in line **TL_B5_B7**. Relays, **PA_B5_TL_B5_B7_OR** and **PA_B7_TL_B5_B7_OR** update their state to `tripped` and produce *cmd_open* events. While the over-current relays associated with lines, **TL_B4_B6** and **TL_B8_B9** move back to `idle`. The breakers, **PA_B5_TL_B5_B7_BR** and **PA_B7_TL_B5_B7_BR** update their state to `opening`.	The observers tracking **PA_B5_TL_B5_B7_OR** and **PA_B7_TL_B5_B7_OR** relays update their state to `tripped`. The observers associated with **PA_B5_TL_B5_B7_BR** and **PA_B7_TL_B5_B7_BR** update their state to `opening`.	
11.081		*t3* wait time expires for the observer tracking relays, **PA_B8_TL_B8_B9_OR**, **PA_B9_TL_B8_B9_OR**, **PA_B4_TL_B4_B6_OR** and **PA_B6_TL_B4_B6_OR**. Observers transition to `idle` and produce h_OR' event to indicate absence of overloading conditions.	Robustness, Plausibility and Failure count of Hypothesis H3 are updated to 87.5%, 100%, 2 respectively and Failure count in H0 reduces to 6
11.102	Breakers, **PA_B5_TL_B5_B7_BR** and **PA_B7_TL_B5_B7_BR** change their state to `open` and produce *st_open* events.	On detecting state change events, the corresponding observers also transition to `open` state and produce mode change (*h_open*) and alarm state change (h_stuck_close') events.	The mode change event causes a change in the hypothesis H1, H2 and H3. Overload alarms are expected from protection assemblies associated with line **TL_B4_B5** instead of **TL_B4_B6**
11.103	The over-current relays, **PA_B4_TL_B4_B5_OR** and **PA_B5_TL_B4_B5_BR** produce *OR* alarms to signal overload conditions and update their state to `chk`.	The observers tracking the behavior of these over-current relays transitions to `chk` state after detecting *OR* event. The observers produce *h_OR* to conclude overloading conditions.	The hypothesis metrics of H3 are updated to 100%, 100%, 2 and failure count in H0 increases to 8.
21.103	Wait time, *WT* of over-current relays associated with line, *TL_B4_B5* expires. The relays move from `chk` to `waiting` state and produce *cmd_open* event.	Wait time for corresponding observers expires and their states are updated to `waiting`	
21.104	Due to persistent overloading conditions, relays, **PA_B4_TL_B4_B5_OR** and **PA_B5_TL_B4_B5_OR** produce *cmd_open* event and transition to `tripped`. The breakers, **PA_B4_TL_B4_B5_BR** and **PA_B5_TL_B4_B5_BR** update their state from `opening`.	The observers tracking relays **PA_B4_TL_B4_B5_OR** and **PA_B5_TL_B4_B5_OR** relays also update their state to `tripped`. The observers associated with breakers, **PA_B4_TL_B4_B5_BR** and **PA_B5_TL_B4_B5_BR** update their states to `opening`.	
21.154	Breakers, **PA_B5_TL_B5_B7_BR** and **PA_B7_TL_B5_B7_BR** change their state to `open` and produce *st_open* events.	On detecting state change events, the corresponding observers also transition to `open` state and produce mode change (*h_open*) and alarm state change (h_stuck_close') events.	**Most Probable Hypothesis is H3:** **Robustness = 100%** **Plausibility = 100%** **Failure Count = 2 (*F_TL_B7_B8*, *F_TL_B8_B9*)**

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

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.

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

Chhokra, A., Dubey, A., Mahadevan, N., & Karsai, G. (2017). Hierarchical reasoning about faults in cyber-physical energy systems using temporal causal diagrams. *International Journal of Prognostics and Health Management, Submitted to*. Retrieved from `http://www.isis.vanderbilt.edu/sites/default/files/IJPHM_chhokrad_rev_0.pdf/`

Di Fazio, A. R., Erseghe, T., Ghiani, E., Murroni, M., Siano, P., & Silvestro, F. (2013). Integration of renewable energy sources, energy storage systems, and electrical vehicles with smart power distribution networks. *Journal of Ambient Intelligence and Humanized Computing*, *4*(6), 663–671.

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

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

Group, W. (1973, Nov). Common format for exchange of solved load flow data. *IEEE Transactions on Power Apparatus and Systems*, *PAS-92*(6), 1916-1925. doi: 10.1109/TPAS.1973.293571

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

Hare, J., Shi, X., Gupta, S., & Bazzi, A. (2016). Fault diagnostics in smart micro-grids: A survey. *Renewable and Sustainable Energy Reviews*, *60*, 1114–1124.

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

Jones, L. E. (2014). *Renewable energy integration: practical management of variability, uncertainty, and flexibility in power grids*. Academic Press.

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

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

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

NERC. (2005). Evaluation of criteria, methods, and practices used for system design, planning, and analysis response to nerc blackout recommendation 13c [Computer software manual].

NERC. (2013). Transmission system planning performance requirements - nerc standard tpl-001-4 [Computer software manual].

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

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.

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).

Venkatesh, C., & Swarup, K. S. (2012). Investigating performance of numerical distance relay with higher sampling rate. In *North american power symposium (naps), 2012* (pp. 1–6).

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

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

APPENDIX

A temporal causal diagram is a behavior-augmented fault propagation graph. It comprises of a directed graph that captures the fault 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 the cyber components.

Temporal Fault Propagation Graphs: A temporal fault 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 fault 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. Formally, the TFPG is represented as a tuple $\{F_{physical}$, $D_{physical}$, *E, M, ET, EM, ND*$\}$, where

- $F_{physical}$ is a nonempty set of fault nodes in physical system. A fault node can be in two states either present denoted by ON state or absent represented by OFF state. A fault node represents a failure mode of the system or a component, and its state represents whether the failure mode is present or not. In the subsequent discussion we will use the terms fault node and failure mode interchangeably.
- $D_{physical}$ is a nonempty set of discrepancy nodes related to fault effects of physical faults.
- $E \subseteq V \times V$ is a set of edges connecting the set of all nodes $V = F_{physical} \cup D_{physical}$.
- *M* is a nonempty set of system modes. At each time instance t the system can be in only one mode.
- *ET* : $E \to I$ 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 fault propagation over the edge.
- *EM* : $E \to M$ 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* : $E \to \{True, False\}$ 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.

Discrete Behavior Models: The behavior of discrete devices is modeled using extended time triggered automaton (Krčál, Mokrushin, Thiagarajan, & Yi, 2004). The extension includes sets of failure modes and failure mode guards. Mathematically, an extended time triggered automaton is represented as tuple *(Σ, Q, q_0, Q_m, F_{cyber}, D_{cyber}, $\mathbb{M}$, $\alpha(F)$, Φ, 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 the sets of observable and unobservable discrepancies.
- **Failure Mode Set:** F_{cyber} is a finite set of unobservable failure modes associated with the component. Similar to a fault 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

  $$\omega_t := \delta_t(f) \quad | \quad \neg\delta_t(f) \quad | \quad \omega_{1,t} \quad \wedge \quad \omega_{2,t} \qquad (1)$$

  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 fault 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*. If no event is mentioned, then the transition is valid only if the failure mode constraint evaluates to true as per the timing constraints.