Failure Diagnosis about Cyber-Physical Energy Systems Using Temporal Causal Models

By

Ajay Dev Chhokra

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

September 30, 2021

Nashville, Tennessee

Approved:
Dr. Gabor Karsai, PhD
Dr. Sherif Abdelwahed, PhD
Dr. Mitchel D Wilkes, PhD
Dr. Theodore Bapty, PhD
Dr. Abhishek Dubey, PhD

To my respected parents Parveen Kumar Chhokra, Sunanda Chhokra

and

my caring sister Vasudha Chhokra

ACKNOWLEDGMENTS

Pursuing PhD has been a life changing experience and I have many people to thank for their continuous guidance, support and encouragement. First and foremost, I would like to extend my deepest gratitude towards my advisor, Dr. Gabor Karsai for his invaluable time, guidance, and unwavering support over the years. Dr. Karsai believed in me and provided me with every opportunity to succeed. He was always available for research discussions and his advise was critical in shaping the outcome of my research work. I am also thankful to Dr. Sherif Abdelwahed, Dr. Mitchel D Wilkes, Dr. Theodore Bapty and Dr. Abhishek Dubey for serving on my dissertation committee and their valuable feedback. I am extremely grateful to Dr. Dubey for insightful discussions and incisive critique that helped making my research work stronger. I also wish to thank Nagabhushan Mahadevan for the many hours he invested in discussing the diagnosis problem with me. I would like to acknowledge the assistance of other members of CPS-POWER project, Dr. Rishabh Jain, Dr. Saqib Hasan, Dr. Srdjan Lukic and Dr. Anurag Srivastava for helping me to model various power system components and sharing the simulation traces. Finally, I thank National Science Foundation (grants 1329803 and 1840052) for their financial support without which this research work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

Introduction

Recent advances in sensor networks, embedded systems, information and communication technology have steered the interest of the scientific community towards the development of a new paradigm, *Cyber-Physical System* (CPS) [3]. CPS is defined as a generalization of embedded systems where a collection of computing devices communicate with one another as well as interact with physical processes via sensors and actuators and the system's functionality emerges from these interactions. Strong 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 [4].

A generic structure of a CPS application is shown in Figure 1.1. The block, **physical plant** models the *physical* part of the CPS. A physical plant mainly consists of mechanical, chemical or biological processes and is realized without digital computers or networks. The *cyber* part is composed of **computational platforms** and a **digital network**. A CPS application can have one or more computational platforms, comprising of sensors, actuators and one or more compute nodes hosting different operating systems along with a network fabric, that provides the mechanisms for the compute nodes to communicate.

One of the emerging applications of CPS is the modern power system, commonly referred to as *Cyber-Physical Energy System* (CPES). CPES is the amalgamation of power grid technology with intelligent control, coordination, and communication between demand and supply side to deliver electricity efficiently [5]. Physical components in power systems include transmission lines, generators, transformers, etc. Cyber components include protection devices such as relays to safeguard physical components from faults such as earthing and short circuit, *Supervisory Control and Data Acquisition* (SCADA) system that collects data from various sensors in remote locations and then transfer to a central node to be processed, *Energy Management Systems* (EMS) to control and

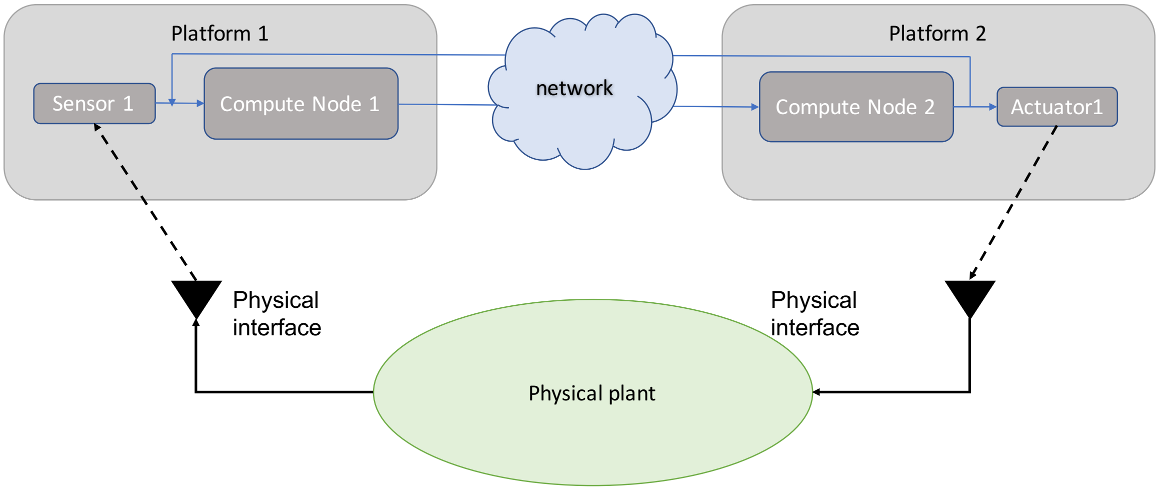


Figure 1.1: Generic Structure of a *Cyber-Physical System*

optimize the performance of power system.

According to [6], some of the key concerns in designing CPS are safety, reliability and fault tolerance. Physical components in power systems are continuously exposed to dynamic environments resulting from varying demand, changing operational requirements and component degradation which can lead to failures. A failure of a component or a system is the loss of its ability to perform a required function under specific operating conditions. Apart from failures due to aging, there could be faults within a component, such as short circuit and winding faults that can alter the behavior of the component. These faults manifest in observable anomalies, i.e., deviations from expected behavior; or, they can remain unobservable. A fault can cause system-level failure(s) and due to the highly connected nature of CPES, a failure of one component can also lead to secondary faults in connected components, possibly leading to a cascade of failures.

To make the system resilient against faults, several localized protection mechanisms are deployed throughout the system to detect and isolate faults. These protection systems include 1) *fault detection devices*, such as, fast-acting numerical (digital) relays that are designed to detect abnormal changes in line current and bus voltage and 2) *fault mitigation devices*, such as breakers, that can be triggered to remove the faulty component from the electrical network. These protection devices are useful in detecting and isolating faults in specific regions of a system, but 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 on system stability. Apart from the lack of system-wide perspective, these protection devices have detection faults. These faults include *Missed* detection, which causes a protection device not to act and *Spurious* detection, which results in unwanted relay operation. The change in system state due to misoperation of the protection devices can eventually increase stress on other parts of the system and thus can spread the fault effects, i.e., fault propagation or can cause secondary faults in other parts of the system. To isolate these secondary faults, other protection devices will trigger and can start a cascading effect which can lead to total system collapse. A closer investigation by *North American Electric Reliability Corporation* (NERC) [7] demonstrated that nearly all recent major blackouts, excluding those caused by severe weather, have had a protection system or automatic control misoperation contributing to cascading failures [8].

Fast and accurate fault diagnosis is essential in power systems as it helps system operators to reduce the system restoration time after blackouts by identifying faulty components that need to be replaced. It also helps utility operators in managing an ongoing cascading outage by providing useful information regarding malfunctioning equipment and enable them to make better decisions for controlling the failure effects. By fault diagnosis, we imply isolating the root cause(s) of the observed anomaly, i.e., the component that is the source of the problem. However, diagnosing faults manually by matching an observed set of defects against a pre-compiled rule book of fault patterns is a tedious, error-prone and impractical task. Moreover, this technique fails when many alarms occur within a short time period, overwhelming the utility operators, barring them from finding the root cause of the problem in a timely manner.

Furthermore, operators have to consider the possibilities of misoperation of the protection system. This problem is often compounded due to loss of information from the protection system due to failure in the field. Inability to timely diagnose the source(s) of failures combined with the potential side-effects of automated protection actions lead to impending fault cascades, which can be avoided. NERC blackout report [8] also lists lack of real time diagnosis information as one of the major contributing factor for the 2003 blackout. Thus, to achieve resiliency and reliability, an autonomous online management tool for CPES is necessary that provides hypotheses about the current state of the system while considering the complex behavior of the protection system.

We propose a model based fault diagnosis approach that can perform timely diagnosis of failures caused by faults in physical components and misoperation of protection systems using available information from the physical and the cyber components of this system. The approach presented here is unique in that it models the fault propagation in the physical system while considering possible failures in the cyber components of the system. The key idea in this work will be to consider the physical and logical connections of the subsystems, and the time required for a fault to propagate from one component to another. That is, we will capture the salient attributes of the fault propagation without explicitly modeling the complexities of an electrical CPS.

The remainder of this dissertation is organized as follows. The chapter 2 provides basic knowledge about the structure and different components of power systems, followed by an overview of different faults in power systems and their associated effects. This chapter also describes the fault cascade phenomenon in networked systems and presents a summary of events that lead to blackout in 2003 along the USA-Canada border. Chapter 3 reviews state of the art in the field of fault diagnostics related to power systems. Chapter 4 introduces the proposed model based fault diagnostics approach and lists three research goals, namely, 1) Development of a novel fault modeling language (chapter 5), 2) Automatic synthesis of fault models for CPES (chapter 6) and 3) Designing a diagnosis system using the fault models based on the newly developed modeling language (chapter 7). A simple prognostics and failure mitigation application based on the response of the proposed diagnosis system is presented in chapter 8. Chapters 9 and 10 conclude the dissertation and list the publications in peer reviewed conference and journals.

Chapter 2

The Electric Power System

The electric power system is one of the largest and most complex human-made systems. It is composed of thousands of generators, transformers, transmission lines along with an extensive infrastructure of measurement, communication and control equipment [9]. It has been integrated into one big fixed-frequency synchronous system covering a large geographic area, such as mainland U.S.A. The electricity supply chain can be categorized based on three functions: *Generation*, *Transmission*, and *Distribution* as follows:

1. *Generation*:The Electrical power generation is the process of generating electric power from sources of primary energy. It is the first stage in the delivery of electricity to end users. Typically, production is carried out in power plants using electro-mechanical *generators*, driven by heat engines that are fueled by combustion or nuclear fission or by other means such as the kinetic energy of flowing water and wind. Additional energy sources include solar photovoltaic cells and geothermal power.
2. *Transmission*: The Electrical power transmission is the bulk movement of electrical energy from a generating site, such as a power plant, to a local distribution system. The energy is transferred through overhead cables called *transmission lines*. These lines interconnect with each other to join various regions in a redundant fashion forming a transmission network . These networks are designed to transport energy over long distances with minimal power losses which is made possible by stepping up voltages at specific points using *transformers*. Transmission lines typically operate at 765, 500, 345, 230, and 138 kV.
3. *Distribution*: The Electrical power distribution system is the final stage in the delivery of electric power, carrying electricity out of the transmission system to individual customers. Distribution systems can link directly into high-voltage transmission networks to deliver power to transmission customers (138 kV), or be fed by sub-transmission networks (69 kV). Primary distribution circuits, also known as *feeders*, carry medium-range voltage to additional distribution transformers that are located in closer proximity to load areas. Distribution trans-

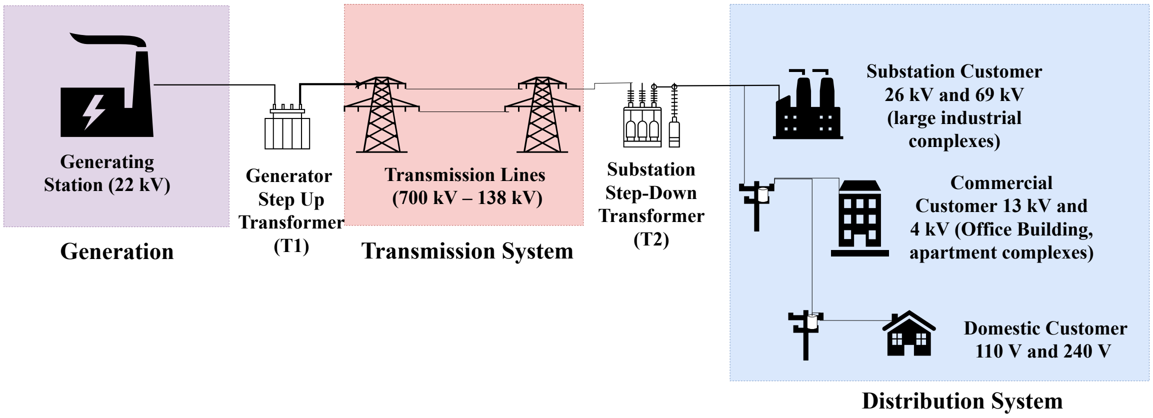

Generating Station (22 kV)
Generation
Generator Step Up Transformer (T1)
Transmission Lines (700 kV – 138 kV)
Transmission System
Substation Step-Down Transformer (T2)
Substation Customer 26 kV and 69 kV (large industrial complexes)
Commercial Customer 13 kV and 4 kV (Office Building, apartment complexes)
Domestic Customer 110 V and 240 V
Distribution System

Figure 2.1: Electrical Power System
https://www.overleaf.com/project/5cfe88749036727b6b608a17

formers step down medium range voltages to low voltage (13 - 4 kV) and route the power over distribution power lines to commercial (13 - 4 kV) and residential customers (120 V) as illustrated in Figure 2.1.

Generation, transmission and distribution systems are linked through key assets known as *substations*. Substations not only provide crucial links for connecting these systems to customers but also assist in monitoring and controlling the quality of power delivered. A substation generally contains transformers, protective equipment, switches for controlling high-voltage connections and electronic instrumentation to monitor system performance. Some important functions carried out at substations are monitoring active and reactive power flows, voltage control and reactive power compensation. While a substation can provide a number of distinct system functions, most utilize transformers to adjust voltage along the supply chain. Listed below are different types of substations in the bulk power system, along with a brief description:

1. *Step-Up Substation*: It links a generation plant to the transmission system. Alternating Current (AC) power plants typically generate voltage below 35 kV, and generator step-up transformers increases voltage to 765 kV in order to transmit power over long distances.
2. *Step Down Substation*: It connects a high-voltage transmission system to a sub-transmission system.
3. *Distribution Substation*: It connects transmission or sub-transmission network to low voltage distribution networks.
4. *Distribution Transformer*: It connects the distribution system to end user customers. These

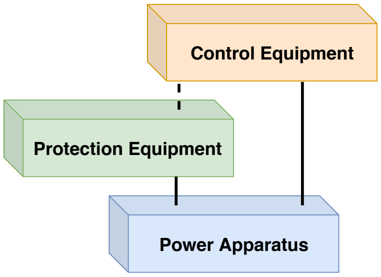


Figure 2.2: Three Layered Structure of Power System

are typically not referred to as substation because they are modular and lack most of the equipment found in a large, high-voltage substation.

5. *Converter Substation*: It connects non-synchronous AC transmission networks to a high-voltage direct current transmission (HVDC), or vice-a-versa.[1]
6. *Switching Substation*: It acts as a circuit breaker in transmission and distribution networks. These substations meant for switching purposes only i.e. disconnecting and connecting a part of the network in order to facilitate maintenance work and do not contain transformers.

A power system is made up of interconnected equipment that belongs to one of the three layers from the point of view of the functions performed as illustrated in Figure 2.2. At the basic level is the *power apparatus* that generates, transforms, and distributes the electric power to end users. The second layer consists of *protection devices* which protect the apparatus from the effects of faults and abnormal operating conditions. The protection equipment can be further categorized into *apparatus* protection and *system integrity* protection. Next, there is a layer of *control equipment*. This layer helps to maintain the power system at its normal voltage and frequency, generates sufficient power to meet the load, and maintains optimum economy and security in the interconnected network. The control equipment is organized in a hierarchy of its own, consisting of local and central control functions. The response time of control functions is generally slower than that of the protection functions as shown in Figure 2.3.

Both protection and control functions change the operational state of the power system but there is a subtle difference in the nature of the actions. Protection functions act to open and close Circuit

[1] The electrical grid that powers mainland North America is divided into four regions or interconnections (two minors and two majors). An interconnection, also known as a wide area synchronous grid, is a region of interconnected AC power systems operating at the same frequency and phase with one another, though not with other interconnections. These interconnections are connected via HVDC transmission lines and converter substations. For more information regarding mainland US electrical grid interconnections please refer to Appendix A.

Breakers (CBs), thus changing the structure of the power system, whereas the control functions act continuously to adjust system variables, such as the voltages, currents, and power flow on the network. Thus protection functions can be defined as functions that lead to operation of power switches or CBs as instructed by protection relays, while actions that change the operating state (voltages, currents, and power flows) of the power system without changing its structure fall under the domain of control functions. Since the scope of this work is tied very closely to the power system protection, a detailed description of architecture and type of devices used is provided in the following subsection followed by a sub-section describing the cascade phenomenon in power systems. For interested readers, a brief overview of the power system control is also provided in the appendix F.

2.1 Power System Protection

Power system equipment is constantly exposed to dynamic environments caused due to changing loading conditions, physical degrading of the components and external faults such as earthing and winding faults. The safety of the system is ensured by a large infrastructure of *protection system assemblies*. A protection system assembly is composed of instrument transformers, protection relays, and high-voltage circuit breakers. Relays sample the scaled-down voltage and current signals from instrument transformers and based on embedded relay logic ascertain the presence of a fault. On detecting the presence of faulty conditions, the relay sends a tripping signal to the breaker which isolates the faulty component from the system. The main components of protection system assemblies are described as follows:

1. *Instrument Transformers*: Instrument transformers are designed to transform voltage and cur-

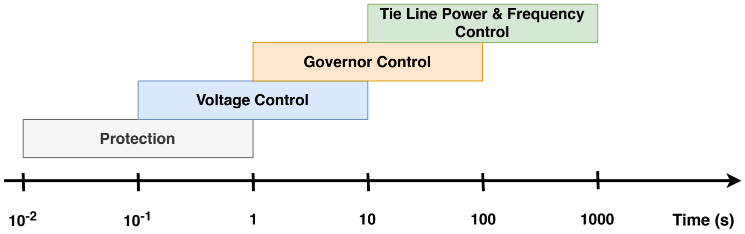


Figure 2.3: Different time scales of power system controls

rent from high values in the transmission and distributions systems to the low values that can be utilized by small voltage metering devices. The amount of scaling of voltage and current is dictated by *turn ratio* between primary and secondary windings of the transformer as indicated by Equation 2.1, where n is the turn ratio, N_p,N_s are the number of primary and secondary coil winding, V_p,V_s are primary and secondary voltages, I_p, I_s are the primary and secondary currents. There are three primary applications for which instrument transformers are used: metering; protection control and load survey. There are two types of instrument transformers:

$$n = \frac{N_p}{N_s} = \frac{V_p}{V_s} = \frac{I_s}{I_p} \tag{2.1}$$

(a) *Current Transformer* (CT) is a type of instrument transformer that is designed to produce an alternating current in its secondary winding which is proportional to the current being measured in its primary. Current transformers reduce high voltage currents to a much lower value and provide a convenient way of safely monitoring the actual electrical current flowing in an AC transmission line using a standard ammeter. The CT is typically described by its current ratio from primary to secondary. A 1000:5 CT will provide an output current of 5 amperes when 1000 amperes are flowing through its primary winding. Standard secondary current ratings are 5 amperes or 1 ampere, compatible with conventional measuring instruments.

(b) *Voltage Transformer* (VT), also called *Potential Transformer* (PT), is a parallel connected type of instrument transformer. They are designed to present a negligible load to the supply being measured and have an accurate voltage ratio and phase relationship to enable accurate secondary connected metering. The PT is typically described by its voltage ratio from primary to secondary. A 600:120 PT will provide an output voltage of 120 volts when 600 volts are impressed across its primary winding. Standard secondary voltage ratings are 120 volts and 70 volts, compatible with conventional measuring instruments.

2. *Protection Relays*: The function of protective relaying is to cause the prompt removal from service of any element of a power system when it suffers a short circuit, or when it starts to operate in an abnormal manner that might cause damage or otherwise interfere with the effec-

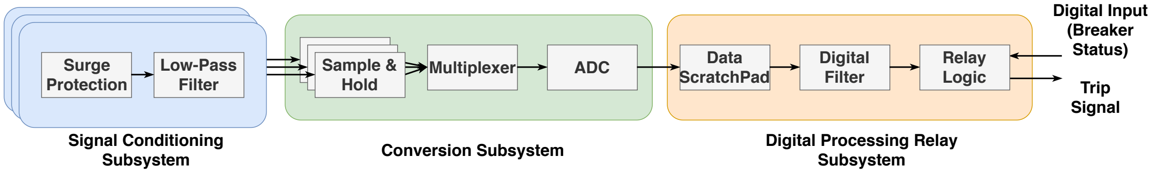


Figure 2.4: Digital Relay Block Diagram

tive operation of the rest of the system. The early relay designs utilized actuating forces that were produced by electromagnetic interaction between currents and magnetic fluxes. However, due to the increased complexity of modern power system and inherent disadvantages associated with electromagnetic relays such as high response and reset times, decrease in reliability due to gradual deterioration of relay components along with the advent of high-performance microprocessors have lead to the rapid adoption of digital and numerical relays. Any computer-based relay can be thought of as comprising of three fundamental subsystems as shown in Figure 2.4:

(a) **Signal Conditioning Subsystem**: This sub-system is responsible for manipulating the analog signal from instrument transformers such that it can be efficiently transformed into its corresponding digital representation. This sub-system is common to all digital and numerical relays and consists of surge protection devices and low pass or anti-aliasing filters. Surge protection devices safeguard the system against high magnitude current and voltage transients (surges and spikes). A low pass filter with a cut off frequency of $F_s/2$ is added to remove high frequency aliasing components of the signal, where F_s is the sampling frequency used by Analog to Digital Converter. Typical sampling and cut off frequencies used in modern digital relays are 8 kHz and 3 kHz respectively [10].

(b) **Conversion Subsystem**: This subsystem is responsible for sampling and converting the filtered analog signal to a discrete signal. The main components of this system are Sample and Hold Circuit, Multiplexer, and Analog to Digital Converter. Sample and hold circuit is an analog device that samples continuously varying voltage and current signals and hold their value at a constant level so that analog to digital conversion is complete. Due to the multi-phase nature of power systems, one sample and hold circuit is used for each voltage and current signal. A multiplexer selects these channels sequentially and routes the output of Sample and Hold circuit to Analog to Digital Converter. Most digital relays use Successive Approximation Register type Analog to Digital Converters that convert the analog quantity to its digital counterpart via a binary search through all possible quantization levels before finally converging upon a digital output. Typical conversion time is of the order of 15-30 μs [10].

(c) **Digital processing relay subsystem**: This is the heart of any computer relay. The main components of this subsystem include high-speed CPU, memory and digital I/O. Modern digital relays consist of two types of microprocessors 1) Digital Signal Processor, for computing phasor representation from the digital samples. A phasor is a complex number representing a sinusoidal function. It is an analytical representation that decomposes a sinusoid into a product of a complex constant and a factor that encapsulates frequency and phase. 2) A general purpose microprocessor, for executing different relay programs, maintenance of various timing functions and communication with different peripherals.

Modern computer based relays allow inter-relay and operator-relay communication over serial and ethernet interfaces. A variety of protocols are available including Mirrored Bits communications [11], Distributed Network Protocol (DNP3) [12] and ModBus [13] that enable fast and reliable communication in direct and multidrop network settings.

3. Circuit Breaker (CB): Circuit breaker is a mechanical switching device capable of obstructing the flow of power through a conductor. Well coordinated action of relays and breakers is essential for successful fault clearing. Currently, a high voltage circuit breaker can interrupt fault currents of the order of 10^5 A at system voltages up to 800 kV [14]. Circuit breakers are classified based on rated voltage levels as 1) Low Voltage ($\leq$ 1 kV), 2) Medium Voltage (1-72 kV) and 3) High Voltage ($\geq$ 72.5 kV) circuit breakers. Low voltage circuit breakers are used for domestic, commercial and industrial applications, whereas Medium and High voltage circuit breakers are deployed at power distribution and transmission substations.

   In general, a circuit breaker consists of fixed and moving contacts called electrodes. Under normal conditions, these contacts remain closed and are opened after receiving a signal from

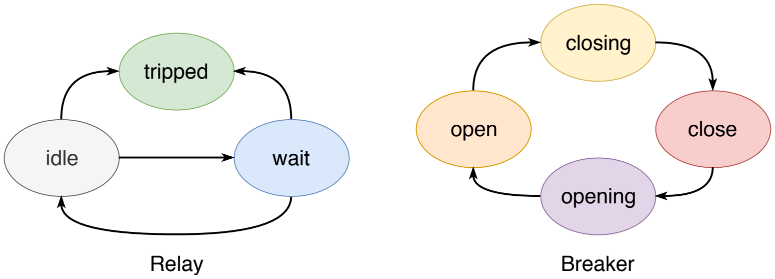


Figure 2.5: Abstract state machines of protection system components (relays and breakers)

protection relays. At the instant when contacts begin to open, the contact area decreases rapidly resulting in increased current density which in turn increases the temperature. The heat produced can ionize the medium between the contacts. This ionized medium acts as a conductor and an arc is struck which provides a low resistance path and consequently the current remains uninterrupted as long as arc persists.

Due to the alternating nature of currents in power systems, the arc is momentarily quenched at current zero (every half cycle). However, when the current across the contact of the circuit breaker is zero, a high frequency transient voltage, called re-striking voltage, develops due to the sudden distribution of energy between the electric and magnetic fields. Under the influence of potential gradient created by the re-striking voltage, ionization of the medium can occur and arc can re-strike. If immediately after current zero, the dielectric strength of the medium between contacts is built up more rapidly than the voltage across the contacts, the arc fails to re-strike and the current will be interrupted. The dielectric strength of medium between the contacts is increased by injecting Sulfur hexafluoride (Sf_6), a high dielectric strength gas and rapidly increasing the separation between the contacts. High and Medium Voltage Circuit breakers can be further categorized based on the arc interrupting medium as 1) Vacuum 2) Sf_6 3) Oil 4) Air Blast circuit breakers. Fault clearing can be done as quickly as the first current zero after the initiation of fault, although it more often interrupts at the second or third current zero [14]. Rated operating time for commercial grade high voltage circuit breakers is 2-3 cycles [15]

Protection system can be classified into *apparatus protection* and *system integrity protection*. Apparatus protection deals with the detection of physical faults (short-circuit and grounding faults) in the apparatus and their respective protection. Distance relays, over-current relays and differential relays [16] are used for apparatus protection. The decision of these relays is based on local information. These relays are designed to protect the equipment in a predetermined region of the power system, known as the *zone* of protection. Relay operation is considered to be secure if it responds only to faults within its specified area of protection or zones. These zones are described using magnitude and phase of bus voltage and line flows.

There is a possibility that a protection system may fail to operate and, as a result, fail to clear a fault. It is thus essential that provision be made to clear the fault by some alternative protection system or systems. These alternative protection system(s) are referred to as duplicate, backup, or secondary protection systems, whereas the main protection system for a given zone of protection is called the primary protection system. A deliberate delay is introduced in the action of any backup relay to clear the fault. This delay allows the primary protection assembly to operate first and in case it fails, only then a backup relay will act.

Figure 2.5(Left) shows an abstract state machine capturing a behavior of the relay element. Initially, the relay is in `idle` state and on detecting fault can transition to `tripped` or `wait` states depending upon the location of the fault. If the fault is in the primary zone, the relay moves to `tripped` state and instructs breaker to open. On the other hand, if the fault is in secondary or tertiary zones, relay jumps to `wait` state and waits there for a specified amount of time depending on the zone. When the wait time expires, the relay can either transition to `idle` state, if the fault has been cleared or `tripped` state if the fault still persists. The abstract state diagram of a breaker, as shown in Figure 2.5(Right) consists of 4 states `open`, `opening`, `close` and `closing`. Initially, breaker is in `close` state and moves to `opening` state after receiving relays command to open. `opening` and `closing` states model the time delay incurred due to arch quenching and mechanical nature of the breaker. After a specified amount of time, the state machine moves to `open` state.

The principal task of apparatus protection is to disconnect equipment or subsystems from the overall power system. Normally, this is the appropriate action. Systems are designed to be resilient, that is, to withstand the removal of one or several elements without unduly stressing the overall system (*N-k security criteria*, see section F.0.2.2). However, if the system is already stressed as a result of multiple equipment outages, heavier than normal loads, or extreme weather, the corrective action may exacerbate the situation and result in wide-area outages. Thus, a different protection scheme is required which would apply to the overall power system or a strategic part of it in order to preserve system stability, maintain overall system connectivity, and/or avoid severe equipment damage during major events. System integrity protection deals with the detection of proximity of the system to the unstable operating region and consequent control actions to restore stable operating point and/or prevent damage to equipment. Under-frequency relays, out-of-step protection, rate of change of frequency relays, reverse power flow relays [16] and [17] are used for system protection. The following subsection describes different types of faults and their effects in power systems along with protection relays that detect and isolate these faults.

2.1.1 Faults

A fault is defined as a defect within a component of the system, cyber or physical that can manifest in observable anomalies: deviations from expected behavior; or, it can remain unobservable.

1. Physical Faults: These faults are associated with physical components such as transmission lines, transformers, etc. In electrical system, there are mainly two types of physical faults, *short circuit* and *ground* (*earth*) faults. A short circuit fault occurs when the insulation of the component fails resulting in low impedance path between energized phases, whereas a ground fault is the result of reduction in the insulation strength between an energized phase conductor and earth or any earthed part of the electrical system. These faults cause excessively high currents to flow through the system and or decrease in bus voltages, that can damage equipment. Due to the multi-phase nature of electrical power systems, these faults are further classified as:

   (a) *Symmetrical Faults* involve all the phases and affect each of them equally. It can be of two types: (1) phase-phase-phase-ground fault (3-ϕ-G fault) or (2) phase-phase-phase fault (3-ϕ). The behavior of 3-ϕ-G fault and 3-ϕ fault is identical due to their balanced nature. Symmetrical faults are very severe faults. However, such faults seldom occur and only about 5% of the system faults are symmetrical faults [14].

   (b) *Unsymmetrical Faults* don't affect all the phases equally leading to disruption in the balanced state of the system. The most common type of unsymmetrical fault is single phase to ground fault (ϕ-G fault) that accounts for 60% to 75% of faults in power systems [14]. The other types of unbalanced faults are phase to phase faults (2-ϕ faults) and phase to phase to ground faults (2-ϕ-G faults). About 15% to 20% faults are 2-ϕ-G faults and 5% to 15% are 2-ϕ faults [14].

2. Abnormal Operating Conditions: Abnormal operating conditions pertain to deviation from the safe operating point. The major cause of these anomalies is disruption in generation-consumption balance caused due to switching of power system equipment such as generators, transmission lines and loads, etc. as a result of a planned outage or a fault isolation action. These conditions can manifest in the form of line overload or bus voltage variation or alteration in system frequency or power swings. If the equipment continues to operate in this state for a long time, it can lead to permanent damage or reduction in life of the equipment and may cause failure.

3. Cyber Faults (Protection System Faults): These faults are associated with the misoperation of the protection system itself. A protection system element misoperates when it either fails to operate as designed or operates unintentionally outside of its scope of protection. Misoperations contribute to outages of generation and transmission facilities that can place the system in an insecure state. Information about relay misoperations is collected through *Misoperation Information and Data Analysis System* (MIDAS) [18]. The average rate of misoperations in the last five years is close to 10% [19], i.e., more than 90% of the time protection assemblies operated correctly for isolating all faults as depicted in Figure 2.6 (Top-Left). According to MIDAS, misoperations are categorized into following classes:

    - **Failure to Trip - During Fault**: A failure of a composite protection system to operate for a fault condition for which it is designed.

    - **Failure to Trip - Other than Fault**: A failure of a composite protection system to operate for a non-fault condition for which it is designed, such as a power swing.

    - **Unnecessary Trip - During Fault** : An unnecessary composite protection system operation for a fault condition of another element.

    - **Unnecessary Trip - Other than Fault** : An unnecessary composite protection system

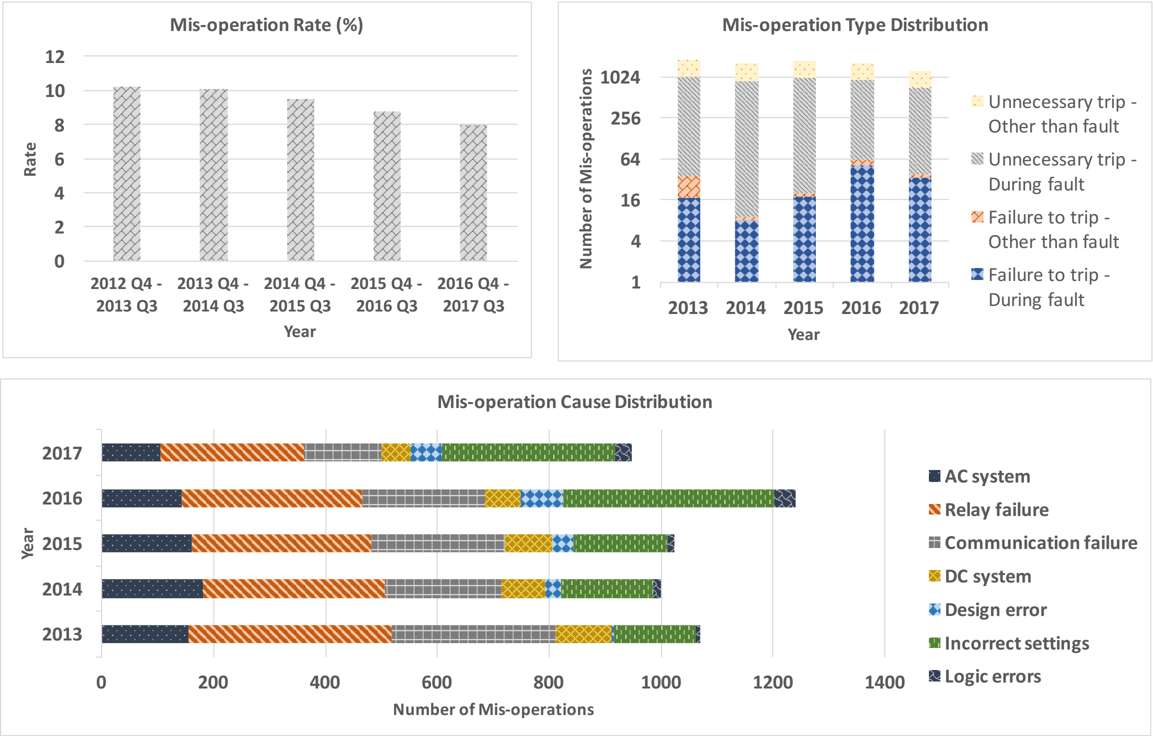


Figure 2.6: **Top Left** figure shows misoperation rate in last 5 years; **Top right** figure highlights the trends in type of misoperatuon faults occurring in last 5 years; **Bottom** figure depicts cause distribution of misoperations

operation for a non-fault condition.

The first two categories of failures generalize to *Missed Detection* faults, i.e., the relay failed to detect the anomaly whereas the last two are part of *Spurious Detection* fault category, i.e the relay incorrectly identified the presence of defects. The spurious detection faults are 75 times more likely to happen than missed detection faults [20] as shown in Figure 2.6 (Top-Right). A variety of reasons causes the protection system misoperations. But NERC's 2018 State of Reliability Report [21] lists incorrect setting, relays malfunctions and communication failures as three most common causes of protection system misoperations, contributing to over 50% of the total number of misoperations in the last five years as shown in Figure 2.6 (Bottom). More details about the causes of the protection system misoperations and their categories are presented in the appendix C.

Relay Characteristic	**Equation**
Standard Inverse	$t(I) = \frac{TD}{7}\left\{\left(\frac{0.0515}{I_r^{0.02}-1}\right)+0.114\right\}$
Very Inverse	$t(I) = \frac{TD}{7}\left\{\left(\frac{19.61}{I_r^{2}-1}\right)+0.491\right\}$
Extreme Inverse	$t(I) = \frac{TD}{7}\left\{\left(\frac{28.2}{I_r^{2}-1}\right)+0.1217\right\}$

Table 2.1: IEEE inverse definite time relay characteristic equations [1]

2.1.2 Relay Operating Principle

2.1.2.1 OverCurrent Relay

Overcurrent relays are used to detect and isolate ground faults in overhead transmission lines [14]. The protection scheme is based on the intuition that short circuit and earth faults lead to currents much above the nominal load currents. An overcurrent relay periodically samples current (scaled down) flowing through a component, if its magnitude is more than a pre-determined threshold (pickup current), presence of fault is indicated. There are mainly three kinds of overcurrent relays 1) *Instantaneous* 2) *Time definite* 3) *Inverse time definite*.

Instantaneous relays instantly send a trip signal after detecting an increase in current beyond the specified threshold. The operating time of an instantaneous relay is of the order of a few milliseconds [1]. Time definite over current relays wait for a pre-defined time after detecting current has passed the pickup threshold. The wait time is independent of the magnitude of the fault current. Inverse time definite over current relay has inverse time characteristics where the wait time is inversely proportional to the ratio of measured current to the pick up current. The inverse definite time relays are further subclassed into three classes depending upon the slope of the T-I characteristics as (1) Very inverse time (2) Extreme inverse time (3) Standard inverse time. Table 2.1 summarizes the IEEE standard equations governing inverse characteristics. Modern over-current relays are also equipped with a directional element to improve selectivity. This feature allows an overcurrent relay to distinguish between upstream and downstream faults and react accordingly.

2.1.2.2 Distance Relay

Distance relays are typically applied to detect transmission line faults (phase and ground) by inspecting the apparent impedance (V/I) [14]. It can also be applied as a backup protection relay for transformers and generators. When a ground or phase fault is introduced in a transmission line, the current flowing through the conductor increases and voltage at the bus terminals drops resulting in a decrease in impedance seen by the distance relay. Distance relays, depending upon the value of the impedance detected, determine the location of the fault.

Typically, distance relays are configured to operate in three zones [22], where a zone is a segment of a transmission line. These zones are defined by the reach of the relay as shown in Figure 2.7. 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, the 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, the 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.

However, the apparent impedance seen by the relay does not follow the transmission line impedance due to the presence of fault resistance, (mainly resistive). Instead, it would lie in the region vicinity as shown in Figure 2.8(a), where line AB represents the impedance phasor of the transmission line. Different shapes of relay characteristics have been used to detect faults as illustrated in Figure 2.8(b) to (e). Usually, distance relay characteristics are visualized by drawing the relay characteristics in R-X plane. If the apparent impedance seen by the relay falls inside the trip region (enclosed region), then relay declares a fault and issues a trip decision.

Distance relays provide fast protection up to 80% of the primary line length. However, primary protection for remaining 20% is deliberately slowed down by zone 2 wait time (co-ordination time). Pilot protection is used for lines to provide the high-speed simultaneous detection and isolation of phase and ground faults for 100% of the primary line. The basic idea behind these schemes is to obtain the response of the distance relay element at the other end to expedite the decision making

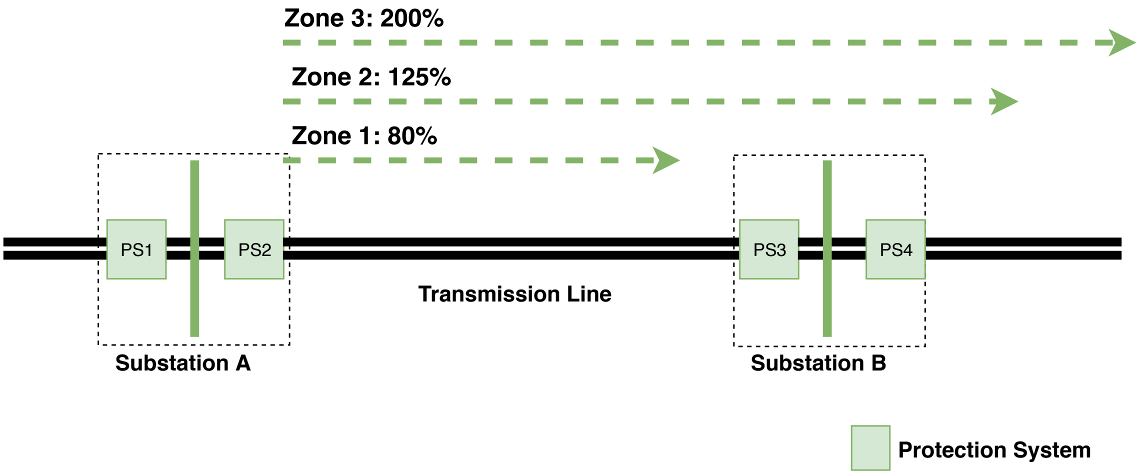


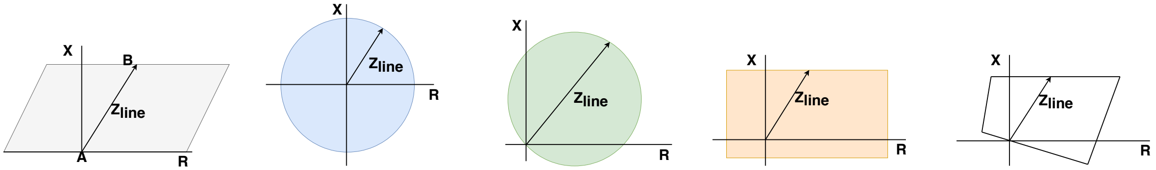
Figure 2.7: Distance Relaying Zones

and bypass the zone 2 wait time.

2.1.2.3 Differential Relay

Differential relays take a variety of forms, depending on the equipment they protect. They are the primary form of protection for transformers against winding earth faults and buses against ground faults [14]. Differential protection is based on the fact that any fault within an electrical equipment would cause the current entering it, to be different, from the current leaving it. Thus by comparing the two currents either in magnitude or in phase, a fault can be detected, if the difference exceeds a predetermined set value. The response time of differential elements is of the order of 16-33 ms [23]. There are mainly two types of differential relay depending upon the principle of operation.

1. *Current Balance Differential Relay*: In a current differential relay, two *Current Transformers* (CTs) are fitted on either side of the equipment to be protected. The secondary circuits of CTs are connected in series in such a way that they carry current in the same direction. If any fault occurs in the external to the zone covered by the CTs, faulty current passes through primary of both current transformers and thereby secondary currents of both current transformers remain same as in the case of normal operating conditions. But if any fault occurred inside the protected equipment, two secondary currents will be no longer equal. In that case, the differential relay is being operated to isolate the faulty equipment.



(a) Transmission Line Impedance Characteristics

(b) Impedance Relay Element Characteristics

(c) Mho/ Admittance Relay Element Characteristics

(d) Reactance Relay Element Characteristics

(e) Quadrilateral Relay Element Characteristics

Figure 2.8: Types of distance relay characteristics

2. *Voltage Balance Differential Relay*: In this arrangement, CTs are connected to either side of the equipment in such a manner that Electromotive Force (EMF) induced in the secondary of both CTs will oppose each other. That means the secondary of the current transformers from both sides of the equipment are connected in series with opposite polarity. In normal operating conditions and also in external fault conditions, the EMFs induced in both of the CT secondary are equal and opposite of each other. But as soon as an internal fault occurs in the equipment under protection, these EMFs are no longer balanced, leading to circuit breaker tripping.

2.1.2.4 Power Swing and Loss of Synchronization Relay

Power swings are variation in three phase power flow which occur when the generator rotor angles are advancing or retarding relative to each other in response to changes in load magnitude, line switching, loss of generation, faults, and other system disturbances. Power swings are classified as Stable and Unstable. A power swing is considered stable if the generators do not slip poles and the system reaches a new acceptable operating state. Whereas during unstable power swings a generator or group of generators experience pole slipping. Pole slipping is a condition whereby a generator, or group of generators, terminal voltage angles go past 180 degrees with respect to the rest of the connected power system.

Large power swings, stable or unstable, can cause unwanted relay operations at different network locations, which can aggravate the power-system disturbance and cause major power outages or power blackouts. A Power Swing Block (PBS) function is implemented in modern relays to prevent unwanted relay operation during power swings. The main purpose of the PBS function is to differentiate between faults and power swings and block distance or other relay elements from operating during a power swing. However, faults that occur during a power swing must be detected and cleared with a high degree of selectivity and dependability.

Unstable power swings can cause large separation of the rotor angles between groups of generators and eventual loss of synchronism. When two areas of a power system, or two interconnected systems, lose synchronism, the areas must be separated from each other quickly and automatically to avoid equipment damage. The systems should be separated in predetermined locations to maintain a load-generation balance in each of the isolated areas. The loss of synchronism condition is also referred to as Out-Of-Step (OOS) condition. The Out-of-Step Tripping (OST) function accomplishes this separation. The primary purpose of the OST function is to differentiate stable from unstable power swings and initiate system separation at the predetermined network locations at the appropriate source-voltage phase-angle difference between separating systems. Sometimes system separation is followed by controlled tripping of circuit breakers in separated areas in case load-generation balance cannot be achieved.

Many different methods are used to detect power swings, each with its strengths and drawbacks [24]. The most common method is based on rate of change of impedance that is, during power swings, impedance travels in a complex plane at a relatively slow pace as compared to physical fault conditions. The two-blinder scheme shown in Figure 2.9 is based on the same principle of measuring the time needed for an impedance vector to travel a certain delta impedance. The time measurement starts when the impedance vector crosses the outer blinder (RRO) and stops when the inner blinder (RRI) is crossed. If the measured time is above the setting for delta time, a power swing situation is detected. An unstable power swing is detected if the impedance trajectory crosses the leftmost blinder (RLO) after a power swing has already been concluded. The set time delay is adjusted so that it will be greater than the time interval measured during a fault and smaller than the time interval measured during the impedance travel at maximum speed. Extensive stability studies with different operating conditions must be performed to determine the fastest rate of possible power swings. PBS functions are implemented in all relays, but OST functions are implemented in some relays at specific network locations. The selection of network locations for placement of OST systems can best be obtained through transient stability studies covering many possible operating conditions.

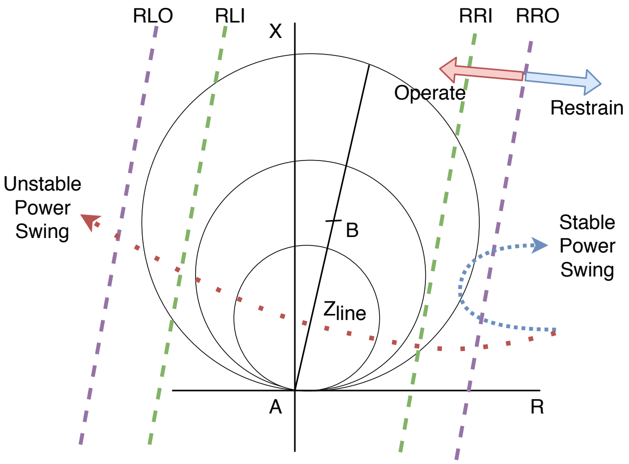


Figure 2.9: Two blinder Scheme

2.1.2.5 Overload Relay

Operation under overload conditions for an extended period causes the component's temperature to increase beyond safe limits and cause damage to the component. The larger the severity of the overload, the lower the time required for overheating to cause damage to the component. Overload protection can be achieved by first detecting when the current flowing through the protected component and then disconnecting the component from the power source before overheating causes damage to the component. Overload protection can be achieved using a variety of means: fuses, low-voltage circuit breakers like miniature circuit breakers and over-current relays used in conjunction with high-voltage circuit breakers.

2.1.2.6 Frequency Variation Relay

The deviation of the frequency from the rated system frequency indicates unbalance between the generated power and the load demand. Over frequency conditions occur when the available generation is large compared to the consumption by the load connected to the power system. Whereas, an excess of load demand creates under-frequency conditions. Generating units cannot operate for an extended period of time in abnormal frequency conditions, as the mechanical resonance (vibrations) will damage the turbine blades.

Over frequency conditions are relatively easier to correct by reducing the output power of a generating unit with the help of governors as compared to under frequency conditions. Under-frequency protection is applied at both generation and load points (sub-stations) using under frequency relays [25]. At load side, after detecting under frequency conditions, substation feeders are tripped to decrease system load in 60-90 milliseconds. If the system frequency is not returned to normal within a certain time period, the generator side under frequency protection scheme kicks in to protect the generator. The wait time is in the range of 0-20 secs.

2.1.2.7 Reverse Power Flow Relay

The reverse power relay is a directional protective relay that prevents power from flowing in the reverse direction [26]. The relay is used in installations where a generator runs in parallel with the utility or another generator to prevent power from the bus bar or another generator from flowing back to the active generator when its output fails.

The relay monitors the power from the generator and in case the generator output falls below a preset value, it quickly disconnects the generator coil to avoid power from flowing into the stator coil. Most of the reverse power relays have adjustable settings to allow the operator to do the settings according to the installed equipment. The trip point is usually adjustable to between 2 and 20 percent of the input current while the time delay is adjustable from 0 to 20 seconds [14].

2.2 Cascading Outages and Blackouts in Power Systems

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 NERC, a cascading failure is defined as an uncontrolled loss of any system facilities, because of thermal overload, voltage collapse, or loss of synchronism, occurring as a result of fault isolation.

There are lots of factors that make the power system prone to cascading outages. These factors can be roughly classified into two groups: a) *non-technical factors*, such as change in operating procedures due to deregulation, aging infrastructure, lack of investment, and inadequate personnel training for new operating conditions, b) *technical factors*, such as operating difficulties, increased system complexity, more difficult protection setting coordination, inadequate traditional security analysis, lack of understanding of the cascades and unavailability of effective support tools.

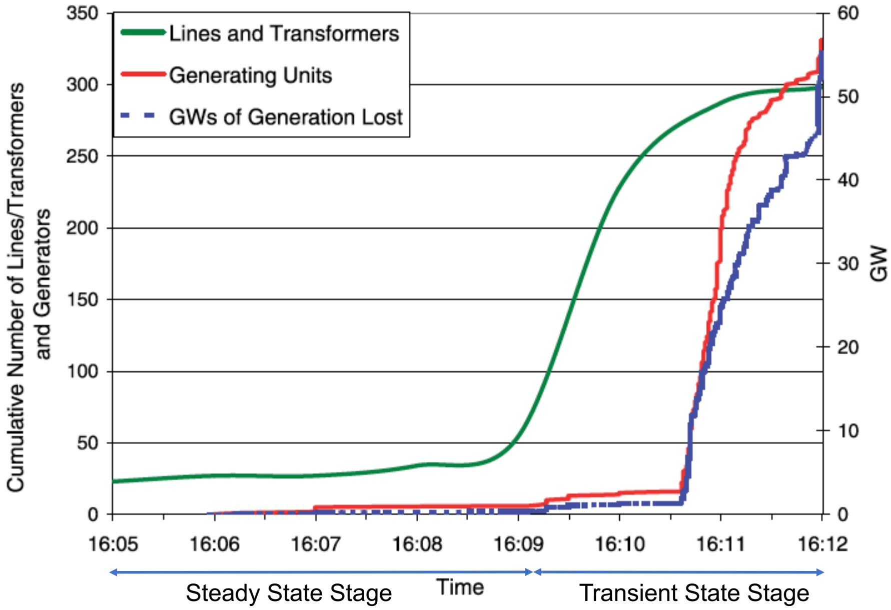


Figure 2.10: Rate of Line and Generator Trips During the US-Northeastern Blackout (Aug 14, 2003)

There are two stages associated with cascading outages. The first stage is the steady state progress stage: a period of slowly evolving successive events. During this stage, system operators evaluate system condition against different security criteria to identify vulnerable contingencies and take some control actions to increase the security level and prevent the possible cascading outages. The control cost is minimal compared with the massive cost of cascading outages. The timescale associated with the evolution of events in this period is of the order of minutes to hours. The second stage is the transient progress stage: a fast transient process resulting in cascading outages and finally the collapse of the entire system. If disturbances are not handled within their Critical Clearing Time (CCT), they will cause the transient stability problem. Some generators may run faster and others may run slower leading to power swings and loss of synchronism and can result in uncontrolled trippings throughout the network. The timescale associated with this period is of the order seconds to minutes. System dynamics need to be carefully considered in this stage by performing transient stability analysis. However, once the system enters this stage, there is not enough time to perform these expensive analysis.

2.2.1 US-Northeastern Blackout of 2003

On August 14, 2003, large portions of the Midwest and the Northeast United States and Ontario, Canada, experienced an electrical blackout. The outage affected an area with an estimated 50 million people and 61,800 MW of electric load in the states of Ohio, Michigan, Pennsylvania, New York, Vermont, Massachusetts, Connecticut, New Jersey and the Canadian province of Ontario. The blackout began a few minutes after 16:00 EDT, and power was not restored for four days in some parts of the United States. Parts of Ontario suffered rolling blackouts for more than a week before full power was restored.

The task force investigating the blackout published an interim report on November 19, 2003 [27] describing the sequence of the events, causes of the blackout and future recommendations. The task force divided the events into seven sequential phases. Software failures dominated events in first and second phase along with planned outage of a generator in northern Ohio and a 345 kV transmission line in southern Ohio (Stuart-Atlanta). The applications that failed were Midwest Independent System Operator's (MISO) state estimator and contingency analysis software as well as utility company, FirstEnergy's (FE) alarm logging program. Third phase consists of two 345 kV line outages (Harding-Chamberlin & Hanna-Junipe) as a result of sagging and coming in contact with vegetation. These line outages pushed the system to N-1 insecure state[2]. In fourth phase, two 138 kV lines (Pleasant Valley-West Akron, East Lima-New Liberty) were tripped as a result of sagging and contacting the underlying distribution line. The outage of these lines exacerbated the system state leading to heavy overloading and poor voltage stability conditions. The distance relay associated with Sammis-Star 345 kV line incorrectly inferred these conditions to be a zone 3 fault and tripped. The outage of Sammis-start at 16:05 EDT triggered the cascade. The cascade progressed at a slow pace till the outage of 345 kV East Lima-Fostoria Central line (due to relay misoperation) at 16:09 EDT, after that big power swing occurred, leading to tripping of multiple lines and generators in the next two minutes as shown in Figure 2.10. Complete summary of time stamped events are listed in appendix E.1.

According to the report, the initial phase of the blackout was caused by deficiencies in specific practices, equipment, and human decisions by various organizations. In total, NERC identified 7

[2] N-1 insecure state implies the tripping of another system facility can lead to cascading outages

violations that caused the blackout and categorized them into 4 groups :

- **Group 1:** Utility company, First Energy (FE), failed to assess and understand the inadequacies of its system, particularly with respect to voltage instability and the vulnerability of the Cleveland-Akron area, and FE did not operate its system with appropriate voltage criteria.
- **Group 2:** Inadequate situational awareness, i.e., FE did not recognize or understand the deteriorating condition of its system.
- **Group 3:** FE failed to manage adequately the tree growth in its transmission rights-of-way.
- **Group 4:** Failure of the interconnected grids reliability organizations to provide effective real-time diagnostic support.

The investigation team found that if 1000 MW load were shed in Cleveland-Akron area before the Star-South Canton line tripping, it would have prevented the subsequent tripping of the Sammis-Star line and if 1500MW load were shed within Cleveland-Akron area before the loss of Sammis-Star 345-kV line, the blackout could have been prevented. Loss of the Sammis-Star line was the critical event leading to the widespread cascade in the Northeastern system.

2.3 Summary

The power system is a complex system that is composed of a large number of physical components such as generators, transmission lines, transformers, etc. that generate, transfer and distribute electricity to end users. The power system is managed through a huge infrastructure of metering, protection and control equipment organized hierarchically. Misoperation of protection system components (relays and breakers) have been known to cause cascading outages and complete or partial blackouts. Fault diagnosis in large interconnected systems such as power system is very important to maintain system reliability and resiliency. NERC has also indicated ineffective diagnostic support, lack of situational awareness and improper communication between different regional coordinators as major causes of blackouts. Online fault diagnosis systems with capabilities of predicting impending failures play an important role in the management system of large networked systems such as power systems. The next chapter provides a detailed survey of the different methodologies developed for diagnosing faults in the power system.

Chapter 3

Related Research

3.1 Introduction

Fault diagnosis is a two-step process, where the first step includes the *detection* of the time origin of abnormality in the system followed by *isolation* of the faulty component, where isolation implies classification and location of the event that occurred [28]. The proliferation of monitoring systems together with the exponential increase in compute power has stimulated the development of technologies dedicated to diagnosing electrical power system failures [29]. Apart from technical innovation, the monetary incentive of faster and efficient restoration of faulty equipment has also motivated research in power systems fault diagnosis. As a result, the problem of fault analysis and diagnosis in power systems has been a constant subject of technical literature over the last 60 years. More than 2100 documents including journals, conference proceedings, and book chapters, have been published since 1939 [30].

In the 1980s, the task of fault diagnosis was mainly delegated to the human operators, resulting in considerable time consumption. The major reason for manual diagnosis was lack of sensor data and compute power to perform fault classification [30]. In the early 1990s, several expert system based diagnosis systems were developed to perform automated diagnosis and aid operators in quickly restoring the power system to its normal operational state. The widespread use of digital relays in the late 1990s made automated local diagnosis possible. However, these intelligent electronic devices may fail in their operation and are primarily focused on equipment protection, with a secondary commitment to the fault diagnosis and reporting.

The fault diagnosis task in power systems is the combination of three sub-tasks: *detection*, *classification*, and *location*. The detection phase is necessary for the identification of the instant the fault was injected into the system. The timing information is necessary to partition the gathered sensor information into two spaces: pre-fault and post fault data. The analysis of these subsets is done by the classification and location routines to classify the type of the fault, i.e., single, phase-to-phase, etc. and its location (equipment and distance from the substation). These sub-tasks are discussed in the following sub-sections.

3.1.1 Fault Detection

The use of IEDs and SCADA data is necessary to implement the detection stage. Data from IEDs consists of recorded and time-sampled events. These events provide information about state transitions in the system, internal flags of the protection relays and tripping signals sent to breakers. Methodologies for detection use the fault records generated by Digital Fault Recorders (DFR) or output of digital protection relays. The detection phase can also be used to verify the behavior of the protection system as illustrated in [31].

3.1.2 Fault Localization

Fault localization sub-task has two objectives, 1) Identify the equipment that has caused the interruption 2) Estimate the fault section, i.e., fault's distance from the substation. Time-stamped data from relays and other IEDs can be processed to estimate the origin and location of the fault, but the amount of data sent by substation IEDs is large which necessitates the development of fault location system as highlighted in [32]. Some of the advanced relays already have this feature [33].

3.1.3 Fault Classification

Fault classification sub-task is responsible for processing the acquired data to estimate the fault type. This stage generally includes the following kinds of faults:

- Single-phase to ground faults and which phase is affected (three possibilities);
- Two-phase faults, either involving the ground or not (six possibilities);
- Three-phase faults, either involving the ground or not (two possibilities).

3.2 Fault Diagnosis Architectures

The main architectures used to implement fault diagnosis methodologies are: (1) central, (2) decentralized and (3) distributed. The architectures are described in more detail as follows:

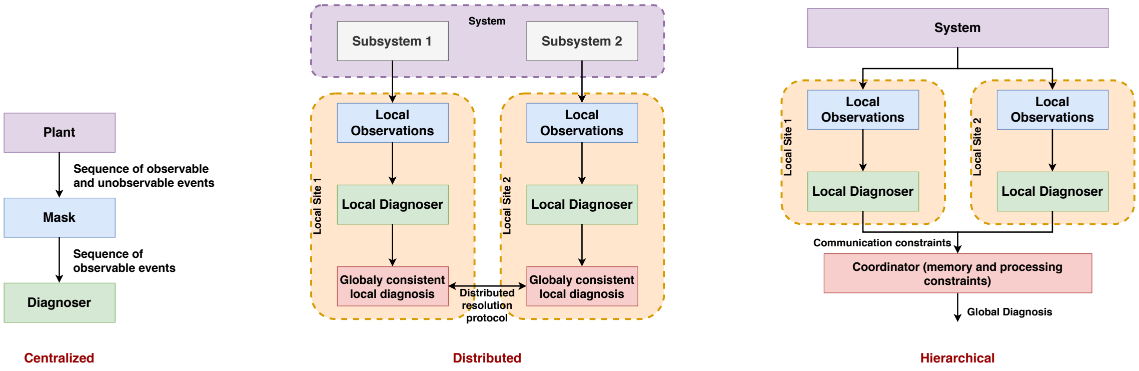


Figure 3.1: Different Diagnosis Architectures

3.2.1 Central diagnosis

In the centralized architecture, the diagnosis is performed in a global, monolithic diagnoser that reason about the system using global fault model of the system. Figure 3.1 **(Left)** depicts the structure of centralized diagnoser. Mask represents the observation function that filters out the un-observable events of the plants and provides a sequence of observed events to the diagnoser.

The main advantage of centralized diagnosis approach is their diagnosis precision and conceptual simplicity. However, its major disadvantages are prohibitive computational complexity, low maintainability, and week robustness.

3.2.2 Distributed diagnosis

Distributed approaches achieve diagnosis using a set of local fault models without referring to a global system model as illustrated in Figure 3.1 **(Middle)**. The aim is to improve the scalability and robustness of diagnostic methodologies. Each subsystem knows only its part of the global model and has its local diagnoser to perform diagnosis locally. This diagnosis computation is based on the local model and the information communicated directly to it by the other local diagnosers through a communication protocol. The information exchanged among local diagnosers is used to update their own information and compensates their partial observation.

A communication protocol must be defined to ensure consistency among local diagnosers in case of conflicts between their local decisions. The challenge of distributed diagnosis is how to perform local diagnosis that is equivalent, if possible, to the global one, using a scalable communication protocol (with respect to the number of component modules), and without the need to use a global model. Secondly, to reach a consensus about faults in a timely manner in the presence of ambiguous, missing and delayed information.

3.2.3 Hierarchical diagnosis (with Coordinator)

In the hierarchical structure as shown in Figure 3.1 **(Right)**, the system is partitioned into several sites. Each site knows the entire system fault model, has local observations and uses a local diagnoser that computes a local diagnosis decision based on its partial observation of the whole system. A coordinator provides the final diagnosis decision as a function of the local diagnosis decisions that are communicated to it. The coordinator is constructed using a global model of the system. The local diagnosers may not communicate directly with each other, and usually only limited communication among them through the coordinator is permitted.

The main problem to address in these architectures is coming to a consensus about the occurrence of a fault, knowing that the available information from local diagnosers can be ambiguous, incomplete, delayed, and possibly erroneous. The coordinator should, therefore, have some memory and processing capabilities to coordinate the required exchange of information between the local diagnosers to resolve the ambiguities of the local decisions. However, these capabilities should be constrained; otherwise, the centralized structure could be replicated at the coordinators site by communicating all observations to it, which would defeat the purpose of the hierarchical structure.

3.3 Fault Diagnosis Methodologies

There are several proposed methods in the literature for fault diagnosis in transmission systems with the presence of uncertainties, partially missing data or historical data. These diagnostic methods use a wide range of techniques, starting from expert systems to analytical optimization methods. Figure 3.2 summarizes the various categories of fault diagnosis techniques applied in power systems. The review of these techniques is presented in the following sections.

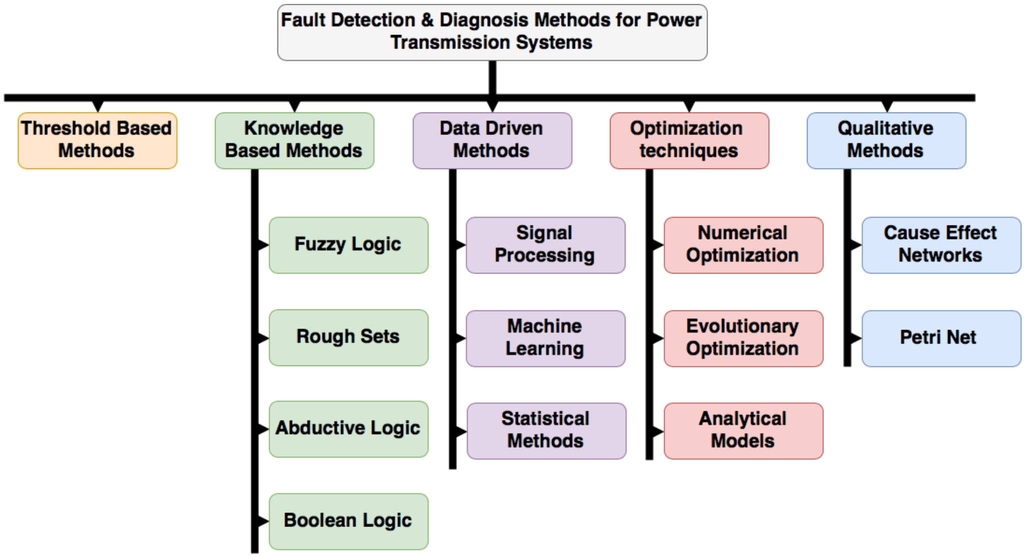


Figure 3.2: List of fault diagnosis techniques applied to power transmission systems

3.3.1 Threshold Based Methods

These methods aim at identifying characteristics is sensor measurements that allow the separation of a healthy and faulty component using simple thresholds or confidence bounds. When these measurements exceed pre-determined thresholds or fall outside confidence bounds, an event is triggered that signals the presence of fault(s). Sometimes the thresholds are made adaptive according to the operating conditions. Although these methods are easily implementable, hard thresholds are prone to high false positive and negative rates due to uncertainties arising from noisy measurements.

3.3.2 Knowledge Based Methods

Knowledge-based systems, also known as expert systems, are methodologies in the area of artificial intelligence that seek to represent the available knowledge about a given task analytically, aiming to infer conclusions. An expert system is a computer program that mimics the cognitive behavior of a human expert for solving problems in a particular domain. These systems are composed of two main blocks: the *knowledge base*, formed by a set of rules of cause and consequence (IF-THEN) responsible for representing all the available knowledge about the task at hand; and an *inference engine* responsible for processing the knowledge base to obtain answers.

For fault analysis in transmission lines, information about the power system is stored in a knowledge base using a set of rules that represent the fault conditions. An inference system is used to compare the data obtained in real time with the knowledge base. A literature review of the main applications of expert systems in power systems, including alarm processing and fault diagnosis, can be found in [34].

To compose the knowledge base and mainly develop the inference engine, it is necessary to use a specific logic, thus allowing the use of mathematical inference rules for building and processing the knowledge base. The first proposals in expert systems for fault diagnosis in transmission lines used propositional or boolean logic [35, 32, 36, 37, 38, 39, 40, 41, 42, 43, 44]. The use of Boolean logic for knowledge representation and processing of information presents some limitations, such as the difficulty in dealing with uncertainty and ambiguity, which encourages the use of other types of logic, such as the theory of rough sets [45], abductive logic [46], and fuzzy logic [47].

3.3.2.1 Fuzzy Logic

As only the information about alarms is known, it is challenging to infer the fault section through deductive reasoning, making the fault diagnosis process an inductive process in which the possibility of missed and spurious alarms from mis-operating protection devices should be considered. In fault diagnosis, uncertainties arise because the premises (causes) of the rules must be inferred using information from parts related to conclusions (effects) in situations in which the malfunction of components may occur. In this context, fuzzy logic is a promising alternative as it can handle imprecise information for making sound decisions.

Fuzzy logic has been used in power system to solve various problems ranging form planning and operations [48]. In [49], the approach based on fuzzy logic is used to identify the type of faults. However, the approach is limited to asymmetrical ground faults. Fuzzy associations among system components, protective relays, and circuit breakers were established with the assistance of experts in protection of power systems. Sagittal diagrams [50] were constructed and different parametric families of operators were tested to obtain the degrees of membership of alarm patterns in each class of the problem.

In [51], fault section estimation is performed considering the network part under the influence of a switching breaker. A three-dimensional matrix is used to represent the network topology and protection system. To deal with the uncertainties of the protection system, fuzzy logic is employed to examine the relationship between the protection devices that operated and fault section candidates. Authors in [52] proposed fuzzy logic based transmission line fault classification technique utilizing three phase currents.

The accuracy of protection systems based on fuzzy logic cannot be guaranteed for large variations in system conditions, because the inference system is built based on the experience of experts regarding certain operating scenarios. Due to the absence of any understanding of the underlying physics of the system, processing missing and conflicting data about the system state and its sequencing of events is a challenge in the knowledge-based system. Consequently, a more reliable algorithm, which considers possible variations of loading and topology in a power system, is necessary for the classification of faults in real time.

3.3.3 Data Driven Methods

Data driven methods are based on extracting useful knowledge from large amounts of historical data. These methods are further divided into three groups:

- *Signal processing techniques*: Signal processing techniques consist of a set of mathematical tools developed for estimation analysis and detection of signals. These methods use statistical concepts related to stochastic processes. They are employed for fault diagnosis in power transmission lines with three main objectives: analysis of oscillography data, feature extraction, and information compression.

- *Statistical methods*: Statistical methods use traditional statistical concepts for knowledge extraction and inference about possible faults.

- *Machine learning techniques*: These techniques involve the development of classification and regression models for detection and isolation of faults. Numerous models based on supervised and unsupervised learning have been proposed to diagnose faults in both transmission and distribution network.

Major contributions in these lines of research is presented in the following sub-sections.

3.3.3.1 Signal Processing

The use of signal processing techniques for faults analysis in the transmission system is widespread in the literature [30]. Most of the studies apply wavelet transform to voltage and current waveforms. The analysis of the obtained coefficients is carried out by a heuristic algorithm to detect the occurrence of the fault event [31, 53, 54]. After the fault is detected, analytical or machine learning based models are used to identify type and location of the fault [55, 56, 57, 58, 59, 60, 61]. In [62, 63], information theory is applied in conjunction with wavelet transform for detection and classification of faults in transmission lines.

Authors in [64] use adaptive Wavelet transform, together with a Bayesian classifier to classify faults in transmission lines. This method uses measurements of only one phase to classify faults involving any phase. The proposal presented in [65] provides a model for combining wavelets for three-phase current signals and the final coefficients are used to detect a fault based on a heuristic algorithm. After the fault is detected, a decision tree, Classification and Regression Tree (CART) is used for fault-type classification. In [66, 67], the authors use the traveling wave theory with Wavelet transform for fault location.

In addition to the direct analysis of the result produced by wavelet transform for diagnosis purposes, the signal processing technique is also utilized as an initial data processing and feature extraction step for machine learning based diagnosing system. As sometimes faults occurring in distribution and transmission networks are challenging to diagnose with raw sensor data especially under the influence of noise in the raw sensor data. Thus, the data processing step becomes necessary to identify certain features that can enhance separability in different fault types. After identifying correct features, data from various faults are arranged in separable clusters in the feature space, thus enabling the classifier trained on this feature space to diagnose faults. These machine learning based classification techniques are described in the next section.

3.3.3.2 Statistical Methods

Like other data-driven methods, these techniques aim to retrieve information from the data collected in the field or generated by simulation. In [68], authors use the median of the sliding window on voltage and current data to detect the fault. Reference [69] computes statistics related to wavelet transform of the current oscillographies for fault detection and provide an indication about where the fault has occurred on a medium-voltage system. The Kalman filter is another statistical technique widely used in power systems, including in the area of diagnosis of transmission lines [30]. Authors in [70] use a Kalman filter with hypothesis testing for event detection in transmission lines. In [71], the Kalman filter is used to estimate voltage and current, and a set of statistics is used for estimating fault occurrence probability at a given instant of time of the oscillography. After the event is detected, its probability of belonging to each of the fault types is calculated.

3.3.3.3 Machine Learning

Artificial Neural Networks: Owing to its ability to approximate nonlinear functions and adaptive learning, neural networks have been used for solving a variety of problems such as load forecasting [72, 73, 74, 75], prediction of flow [30] and fault diagnosis. A literature survey of neural network applications to power systems, including alarm processing and fault diagnosis can be found in [76]. In the context of fault diagnosis, most of the proposed models are derived using supervised learning. The use of artificial neural networks for fault diagnosis in transmission networks lines requires the development of classification and regression models. Classifiers estimate the decision boundary that separates the subspace of patterns belonging to a particular class from other classes, i.e. differentiating among faults. Lastly, regression models aim to estimate the continuous function that relates the input patterns with the desired output, i.e. the distance from the substation to the point of fault. The technical literature presents several types of artificial neural networks to address fault detection, classification and location, such as, Multi-Layer Perceptron (MLP) and Radial basis Function (RBF) networks, Recurrent Neural Network (RNN) and Extreme Learning Machine (ELM).

Most of the neural network based diagnosis systems are aimed for ex-post analysis, and thus assume the detection time of the fault event is provided by the protection system with oscillographies of voltage and/ or current containing samples collected before and after the event. In [77], authors uses RBF networks for fault classification using current and voltage samples. Reference [78] uses MLPs for the same task. Reference [79] proposes a RBF network for fault classification by utilizing five pre and post fault samples of three phase currents and zeros sequence components. Apart for depending upon the accurate detection, this methodology also assumes the information regarding the fault is of the ground type. [80, 81] predict the location of the fault in transmission lines based on the current and voltage sample where samples are collected from one end in [80], whereas [81] considers samples from both ends. Due to the computational effort needed to train neural networks on voltage and current waveforms, authors in [82], proposes a MLP based approach that uses breaker status and protection relay state instead of oscillographies to predict the location of the fault.

Previously mentioned proposals directly analyze the oscillographies from digital fault recorders without any pre-processing. However, there are a lot of methodologies that transform the data using signal processing techniques to extract important features which are learned by neural network models. Authors in [83, 84] propose the use of wavelet transform to process the current signals, generating features to be presented to MLP neural model for fault classification. References [85, 86, 87, 88, 89] compares different classification methodologies based on neural networks that utilize wavelet transform to extract features. The wavelet transform is not the only signal processing technique used in fault classification and location methods. Authors in [90] use Fourier transform as a tool for generating input patterns. The transform is applied to the leakage current, i.e. difference of the current measurements from both ends of the line. The extracted features are used by neural network for fault location. Reference [91] proposes the use of hyperbolic S-transform to process three phase voltage and current signals. The extracted features are used to train RBF models for fault classification and location.

Reference [92] presents a methodology combining wavelet transform and extreme learning machines (ELM) for fault diagnosis. ELMs are neural networks with a single hidden layer, but the weights connecting the inputs to the hidden layer are randomly chosen. The main advantage of ELMs is the smallest computation time as compared to MLP and RBF networks.

Support Vector Machines: Artificial neural network models require a large amount of data for training and the resulting model can suffer from the problem of over-fitting. Moreover, the nonlinear functions approximated by neural networks are optimized by local gradient methods that can lead to inaccurate predictions. These problems are partially addressed by Support Vector Machines (SVM). SVM tries to estimate the optimal decision boundary between data points of different classes, i.e. it aims in determining the separation boundary between the classes that provides the greatest distance between them.

Similar to neural network based proposals, most of the SVM based proposals also rely on fault detection information provided by the protection system. In [93], SVM based fault classification technique is presented that uses samples of three phase currents and their zero sequence components per cycle. References [94] proposes a SVM based methodology for classification and location of the fault. Authors used Electro-Magnetic Transients Program (EMTP) [95] for generation of fault scenarios , simulating different types of faults with variable location by altering the fault distance from substation.

Transitioning to the methodologies that use signal processing for feature extraction. Reference [96] presents such an approach of transforming the voltage data using the wavelet transform and presenting it to SVM for classification and location. Authors in [97] also use the same method but use three phase current signals instead. Whereas, proposal [98] considers both voltage and current signals. SVM and wavelet transformation based methods are used in estimating the location of the fault using voltage and current waveforms in [99, 100, 101]. However, in the first method, location problem is reduced to a binary classification problem, i.e., the output indicates the presence of fault before or after the capacitor bank. Reference [102] presents a hybrid methodology in which the wavelet transform is used for feature extraction followed by SVM model for classification and RBF network for location. Other feature extraction techniques are also used in combination with SVM models. In [103], authors use a technique based on determinants of data collected by digital event recorders, aiming to generate input patterns of SVMs for fault classification and location.

Other Models: Few unsupervised learning based models have been proposed for fault detection, classification and location. The work presented in [104] use k-nearest neighbor for fault classification in transmission lines. Authors in [105], describe another unsupervised learning technique based on adaptive resonance theory for both classification and isolation of faults.

There are a number of approaches presented in the power systems diagnosis literature that combines machine learning and signal processing methods to detect, classify and locate a fault. But there are two fundamental challenges associated with data driven methods. Firstly, the problem arises due to the nature of the machine learning based models, i.e., performance of the learned model is contingent upon the quality of data that is used to train. Most of the models are trained using simulated data and or limited actual operational data. Thus the performance of these models cannot be guaranteed in the wake of changes in the topology that can lead to new patterns for the already learned faults. Secondly, most of the machine learning methods depend upon the quality of the protection system's fault event detection functionality and the effectiveness of the diagnosis system can be compromised in the presence of protection system misoperations.

3.3.4 Optimization Techniques

The diagnosis methods based on optimization tools seek to compare the recorded voltage or current oscillographies with those obtained by a parameterized simulation model of the system. The parameters of the simulation model are related to the occurrence of the fault, namely: type of defect; distance from the substation to the fault; angle of incidence and fault resistance. Through manipulation of these variables, optimization techniques are used to find a solution such that the simulated system oscillographies show greater resemblance to the actual recorded in the field. In [106], authors use methods of numerical optimization $Nelder-Mead$ and $Broyden-Fletcher-Goldfarb-Shanno$ methods with the SIMULINK simulation [107] software to locate faults in transmission lines.

In addition to the methods of numerical optimization, the literature on fault diagnosis in transmission lines presents applications that use genetic algorithms [108]; Particle Swarm Optimization (PSO) [109, 110] and simulated annealing [111]. A literature review on the application of evolutionary computation in power systems, including methodologies proposed for alarm processing and fault diagnosis, can be found in [112].

Authors in [113] use harmony search to detect the instant of occurrence of the fault, the fault-type classification, and estimation of the distance from the substation to the point of failure. The SIMULINK simulation software is used for the simulation of the transmission line under study and optimization tools developed in MATLAB software are used. Using the same approach, in [114] the authors use genetic algorithms as optimization tools.

Besides the comparison between recorded and simulated oscillographies, other criteria can be optimized to characterize the fault type associated with a particular oscillography. For instance, a criterion that quantifies discrepancies between real and expected states of circuit breakers and protection relays of the system can be built. An optimization method is then used as a search tool for the solution that minimizes that criterion. In [115], an evolutionary algorithm is proposed for that optimization phase. In [116, 117], a method with an analytic model is proposed. This model incorporates the time characteristic of alarm sequencing and protection relays coordination. Another analytical method using a hierarchical structure can be seen in [118]. Analogous to expert systems, the methods based on analytical models also have the difficulty of dealing with the presence of multiple faults.

3.3.5 Cause Effect Methods

Diagnosis is the inverse of simulation. Simulation is concerned with the derivation of the behavior of a process given its structural and functional aspects. Diagnosis, on the other hand, is concerned with deducing structure from the behavior. This kind of deduction needs reasoning about the cause and effect relationships in the process. Cause-Effect graphs explicitly represent the relationships between events that define the behavior of a system. Often, there is a notion of temporality in a sequence of events derived from a chain of cause-effect relations. Therefore, following a sequence of events derived from the causal graph provides an abstract notion of the systems dynamic behavior.

Since causal graphs show the effects of faults (causes) they can be used for alarm management applications in two ways: 1) recovering all the consequences of failures using a top down approach, and 2) determining the causes using a bottom-up approach starting from a consequence (i.e., an alarm). Signed Directed Graph (SDG) [119], Temporal Causal Graph (TCG) [120], fault trees [121], and causal probabilistic models [122] use cause/effects relations to represent the fault model and have been used in manufacturing and chemical plants for fault diagnosis. These methods employ qualitative reasoning methods, thus avoiding the complexity of numerical calculations.

Cause-effect networks have also been used to diagnose faults in power systems [35, 123]. A cause-effect network consists of nodes and edges where nodes represent faults or anomalies and edges represent failure propagation. Authors in [124, 125] presented a cause-effect network based fault diagnosis system to detect and locate multiple faults. However, this technique doesn't distinguish between faults and is unable to justify the protection system actions, whether a relay has acted correctly or not. Reference [126, 127] presents a robust hybrid cause-effect and fuzzy rule-based fault diagnosis methods while considering uncertainties in protection system actions. Above mentioned approaches are designed for distribution substations with a radial configuration, so they are not applicable to transmission networks. Therefore, the authors of [128] proposed an improved cause-effect network to extend the applicability of matrix-based fault reasoning to transmission networks. Although the approach used in [128] can detect fault sections for different types of transmission networks, it suffers from a decreased reasoning ability, since it uses a Boolean inference mechanism. In addition, the computational efficiency of these approaches is highly affected by the size of the rule matrix. Since the matrix is built with respect to the entire power system, the method is suitable for a system with a limited number of components. Reference [129] proposes a fuzzy digraph based technique for diagnosing faults in large scale transmission networks. This approach uses a component based approach and constructs faults models for each transmission line separately and then uses fuzzy logic to generate fault candidates.

3.3.6 Petri Nets

Petr Nets (PN) based fault diagnosis techniques have been proposed in scientific literature [130] . Authors in [131] use timed and color PNs to model the behavior of relay and circuit breakers to diagnose faults correctly. However, this method does not consider incomplete or inconsistencies in alarm information. Authors in [132, 133, 134] combine PNs with fuzzy logic to create a robust diagnosis engine that produces satisfactory results even in the presence of missing alarm information. The major features of PN-based methods are graphical knowledge representation and parallel information processing. However combinatorial explosion of state space in large power networks are its weaknesses.

3.4 Summary

Fault diagnosis in power system is an active area of research. Different data-driven and model-based methodologies have been proposed to detect, classify and locate faults in both distribution and transmission networks. However, in the last decade, a large number of proposals related to qualitative and data-driven approaches have been presented. There are some limitations associated with these methods, namely, slow convergence rate and lack of generalization in case of machine learning methods; high computation complexity and inability to handle multiple faults in analytical methods; insufficiency of cause-effect models to appropriately model failure propagation due to behavior of protection system; and lastly state space explosion in Petri-net based models. Moreover, most of the approaches depend upon protection relays for data and do not consider unavailability of data or fault in their operation.

Although various techniques have been developed for on-line system diagnosis in power networks, the actual on-line implementations are minimal. Primarily, the difficulty in most systems is caused by the lack of detailed relay operations in the SCADA system and the system conditions under which these relays operate. Most of the online diagnosis implementations follow the centralized design. A fully centralized reasoning approach is not well-suited for the on-line diagnostics of extremely large-scale systems that are made up of many sub-systems. In such cases, a single centralized diagnostics engine might not scale to provide the desired response time. In a centralized system, data from all substations and control centers are routed to one processing node. Decentralized and distributed diagnosis is better-suited for power systems as it will allow standard protocol of exchanging local hypotheses.

Chapter 4

Fault Diagnostics in CPES using Temporal Causal Diagrams

A post blackout report by the NERC, described the need for more data and research in the field of diagnosis so that critical fault cascades can be prevented in time [8]. According to the report, *A deeper investigation into the root causes of protection system misoperations which contribute to dependent and common mode events is a high priority* [8].

Analyzing fault cascades in the electric grid is challenging because the current state of the art does not capture these fault cascades in a way that could enable automated system-wide fault diagnostics. Effect of failures in protection system components, protection settings, software tools, and human decision on power system physical components are not captured either. In the absence of a system-wide integrated fault model, faults are identified by directly observing the associated anomaly or a set of anomalies as part of a pattern. However, this technique fails when many alarms occur within a short time period. It has been noted that in case of transmission systems this leads to a situation where the utility operators are typically overwhelmed with alarms. Considering that efforts are being made to increase the visibility and reach of the system operators to analyze wide area transmission system or distribution system, fusing all data available will become even more challenging.

Furthermore, operators have to consider the possibilities of misoperation of the protection system. For example, a false fault signature in transmission protection systems can be triggered due to the impression of a fault. Distance relays have been known to incorrectly initiate tripping due to apparent impedance that fall into the zone settings of line relays caused by heavy load and depressed voltage conditions. Protection malfunction and its correlation with major blackouts require a careful rethinking of its system-wide effects. This problem is often compounded due to the loss of information from relays or RTU failure in the field. Such hidden relay failures are hard to locate and are responsible for cascades.

Thus, to meet the required resiliency and reliability, efficient online management of CPES is necessary to operate safely within specified parameters, even in the presence of faults. In order to improve the manageability of CPES, there is a need for technology that can be used to (a) model cascade effect of failure propagation across various levels of these cyber-physical systems, and (b) diagnose and prognosticate the components failure/ faults in the system based on direct or indirect sensor measurements in real-time. However, automatic fault diagnostics in large CPS such as smart grids is challenging. The foremost challenge is to create a model that can be used to analyze the dynamics of fault propagation by taking into account both continuous and discrete dynamics of the underlying system and also capture the effect of misoperations in protection system components. Another challenge is caused by the geographical size of the system. This can often lead to communication delays or communication outages between different subsystems. Any failure analysis methodology should be able to account for these delays. Furthermore, the analysis has to be fast enough so that the operators have sufficient time for mitigation actions before the fault propagation impacts a critical component or functionality. Lastly, the heterogeneity of subsystems necessitates the use of techniques that are domain-independent and applicable in different physical domains including electrical, mechanical as well as software.

We propose a graphical model based approach for diagnosing faults in both physical and protection system of the power system. The graphical model is composed of 1) discrete abstraction of the physical system by using qualitative temporal causal models that capture fault propagation in power apparatus, 2) discrete models capturing nominal and faulty behavior of protection system IEDs and, 3) interaction between the two. The diagnosis engine would consume alarms and other observable internal flags of digital relays to produce and refine hypotheses. The overall research objective of developing an online diagnosis tool is broken into four smaller research goals, illustrated as:

Goal 1: Developing a modeling language (graphical) for capturing different faults in physical and cyber components in CPES followed by their propagation.

Goal 2: Developing methodology for modeling and possibly synthesizing models from the specification of CPES and the connectivity diagram of the cyber components.

Goal 3: Develop efficient diagnosis algorithms and tools that can be used to isolate failures in a scalable fashion.

These goals along with their respective problem statements, challenges, evaluation strategies and solution approaches are described in following chapters.

Chapter 5

TCD Modeling Formalism for CPES

5.1 Problem Statement

Develop a modeling formalism, $\mathscr{L}$, that should represent 1) faults, discrepancies and fault effect propagation across the physical system. 2) nominal operation of the protection system in terms of the observed effects, the control action and its influence on the modes that control the state of the system. 3) failure modes associated with protection system and their effect on the operating modes of the system.

5.2 Challenges

Major challenges in developing modeling formalism are 1) To provide sufficient language features to appropriately capture the effects of fault in both physical and cyber components. 2) To define and verify the fault effect propagation semantics within and across component boundaries.

5.3 Solution Approach

Our approach for modeling faults and their propagation is based on *Temporal Fault Propagation Graph* (TFPG) [135] that captures the causality and temporal characteristics of fault effect propagation in dynamical systems. The classical TFPG is a discrete-event model that captures the causal and temporal relationships between faults (causes) and discrepancies (effects) in a system, thereby modeling the fault cascades while taking into account propagation constraints imposed by operating modes and timing delays. In this graphical model, nodes can be faults or discrepancies, and edges represent the direction of causality. The edges have two attributes which capture the conditions (mode and temporal delays) under which fault effect propagates, see Figure 5.1.

The TFPG based fault model is generic or domain agnostic that has been applied to represent faults and their propagation in various physical domains [136]. However, it cannot capture the effect

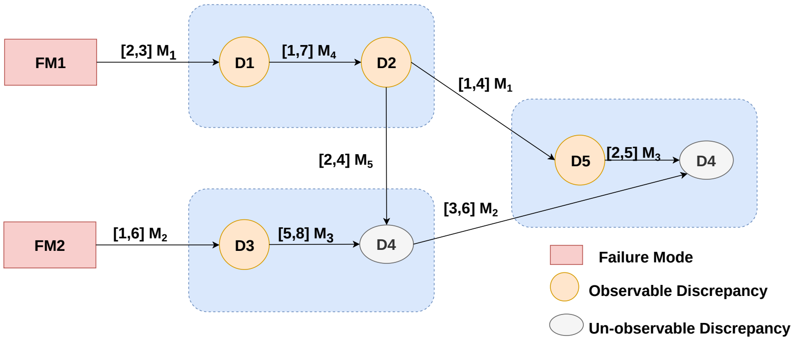


Figure 5.1: TFPG Model with two Failure Modes (FM1, FM2); six Discrepancies (D_1,..,D_6) and fault propagation links. Labels on edges indicate delay (min, max) values and system modes under which edges are active (M_1,...,M_3)

of the built-in automatic fault-protection mechanisms in the system. Such local fault protection components are designed to mask the effect of faults and arrest the fault cascades. Additionally, these fault protection components introduce failure modes that are specific to the operation or lack of operation of the protection components.

We propose a new graph based formalism, *Temporal Causal Diagram* (TCD), $\mathscr{L}_{TCD}$, that is based on a discrete-event abstraction of the physical system, and is also capable of representing faulty and nominal behavior of protection system components (relays and breakers). The formalism, $\mathscr{L}_{TCD} \supset \mathscr{L}_{TFPG} \cup \mathscr{L}_{TTA}$, is a super-set of TFPG language as it contains extended set of nodes and edges to model behavior of protection system components using *Time Triggered Automata* (TTA) [137].

A TTA is essentially a timetable for digital devices, describing what a device should do at a given point of time. Similar to finite automata, TTA are finite labeled directed graphs where input alphabet for the automata is a finite set of events. Each transition of TTA is labeled with an event and a time constraint from $\phi = \{[n], (n) | n \in \mathbb{N}_{\geq 0}\}$, where $[n]$, denotes an *instantaneous* constraint, read as *at n* and (n) denotes a *periodic* constraint, read as *at every n unit of time*. Figure 5.2 shows a TTA with four locations, (S1, S2, S3, S4) with S1 being the initial location. In location, S1, after every time unit, the automaton checks for the occurrence of event *a* and if *a* is observed, then the automaton jumps to S2. In S2, if *c* is observed at time 3 (since automaton entered S2), then automaton transitions to location S3 whereas, if *b* is observed then automaton jumps back to S1. In location S3, if event *b* is observed at 2, the automaton moves to location S1 whereas, if no event is

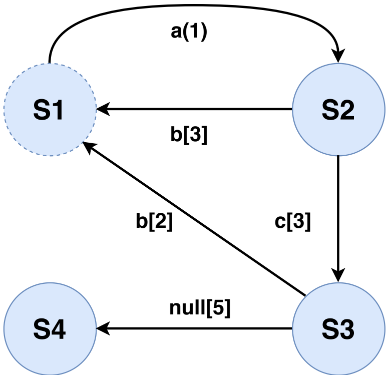


Figure 5.2: Time-Triggered Automata

observed by 5 then automaton transitions to `S4`. Special event labeled as, *null*, represents absence of events.

5.3.1 Extensions to TFPG and TTA

Apart from $\mathscr{L}_{TFPG}$ and $\mathscr{L}_{TTA}$, $\mathscr{L}_{TCD}$ consists of number of extensions to model fault effect propagation between TFPG and TTA sub-models. These extensions are described as follows

- A mode-location map that relates the state of TTA sub-models to system operating mode, thereby influencing the fault effect propagation in TFPG sub-models.
- An node-event map that relates activation or de-activation of nodes in TFPG sub-models to events in TTA, thereby modeling synchronization between fault effect propagation (TFPG) and response of protection devices (TTA).
- An edge certainty map that stores non-deterministic propagation of fault effect along an edge in TFPG sub-models.

5.3.2 Temporal Causal Diagrams

A TCD model is a behavior augmented fault propagation graph where behavior of protection system components is explicitly modeled using TTA. Formally, a TCD model, $\mathscr{G}$, can be defined as a tuple,

$$\mathscr{G} = (F, D, E, M, ET, EM, ND, Q, Q_0, \Sigma, \Psi_{act}, \Psi_{ina}, \Phi, \Omega, T)$$

where

- F is a nonempty set of faults in the system. F is partitioned into two disjoint sets, F_{phy} and F_{cyb}, where the first set represents the faults in the physical components and the latter shows faults associated with cyber or protection system.

- D is a nonempty set of observable discrepancies. It is a combination of two disjoint sets, D_{phy} and D_{cyb}, where the set D_{phy} represents fault effects related to F_{phy} and D_{cyb} are discrepancies related to cyber faults.

- $E \subseteq V \times V$ is the set of directional fault propagation edges connecting two nodes, where $V \subseteq F \cup D$ such that (1) there are no self loops, (2) a physical node, $v_i \in F_{phy} \cup D_{phy}$, is not connected to cyber node, $v_j \in F_{cyb} \cup D_{cyb}$ and (3) fault nodes cannot be destination node.

- M is a nonempty set of system modes. At each time instant, t, the system can be in only one mode.

- $ET : E \rightarrow \mathbb{R}^2_{\geq 0}$ is a map that associates with every edge, $e \in E$, a time interval $[t_{min}, t_{max}] \in \mathbb{R}^2_{\geq 0}$, such that $t_{max} \geq t_{min}$, where t_{min}, t_{max} are the minimum and maximum time for fault propagation to occur along the edge, i.e. given the edge is active oe enabled it takes *at least* t_{min} and *at most* t_{max} amount of time for fault effect to propagate from the source to the destination node.

- $EM : E \rightarrow \{M^n \cup \varnothing\}$ is a map that associates with every edge $e \in E$, a set of modes under which the edge is active. For mode independent edges i.e. active in all system modes, $EM(e) = \varnothing$.

- $ND : E \rightarrow \{\top, \bot\}$ is a map that associates with an edge, $e \in E$, $\top$(True) or $\bot$(False), where $\bot$ implies the propagation along the edge, e **will** happen, whereas $\top$ implies the propagation is uncertain and **can** happen.

- Σ is a finite set of event labels. We categorize events into two types, observable (Σ_{obs}) and unobservable (Σ_{unobs}). Observable events include alarms related to observable discrepancies, commands or messages exchanged between cyber components etc. Whereas, unobservable events are related to injection of faults in the system.

- Q is a finite set of locations associated with time triggered automata, and Q_0, is the initial location set.

- $\Omega : M \rightarrow f(Q^n)$ is a map that relates a system mode $m \in M$ with a boolean function defined over locations $q \in Q$. A boolean function, $f : Q^n \rightarrow \{\top, \bot\}$ can be viewed as a constraint on the locations of cyber sub-system automata. At any time t_1, a system mode m represents the actual operating conditions if the corresponding boolean constraint is satisfied by the locations of cyber sub-system automata, Q_{t_1}, i.e. $f(Q_{t_1})\big|_{t=t_1}$ evaluates to `True`, where $f = \Omega(m)$.

- $\Psi_{act} : F \cup D \rightarrow \Sigma$ is a map that relates activation of nodes, $v \in F \cup D$, in TFPG sub-model with events labels, $\sigma \in \Sigma$, in TTA.

- $\Psi_{ina} : F \cup D \rightarrow \Sigma$ is a map that relates de-activation of nodes, $v \in D$, in TFPG sub-model with events labels, $\sigma \in \Sigma$, in TTA[1].

- Φ is a set of timing constraints, $\Phi = \{[n] \mid (n)\}$ for $n \in \mathbb{N}_+$, 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 the presence of events.

- $T \subset Q \times \Sigma \times \Phi \times \Sigma^n \times Q$ is a finite set of transitions between any two locations. Each transition of the time triggered automaton is labeled with an event request, a timing constraint and output event(s). For example the tuple, $(\, q_1,\, \sigma_1,\, [n],\, \sigma_2,\, q_2 \,)$ represents a transition from location $q_1 \in Q$ to $q_2 \in Q$ where $\sigma_1, \sigma_2 \in \Sigma$ are input and output events respectively and $[n]$ is a instantaneous time constraint. The transition is enabled only iff the event σ_1 is valid at time, $t = t_1 + n$, where t_1 is the time when automaton entered location q_1. An event, $\sigma \in \Sigma$ is valid at time t if $t - t_{occ} \leq \varepsilon$, where t_{occ} is the time of occurrence of event σ.

The TCD model of an arbitrary CPS is described in Table 5.1. The example system consists of two protection devices (`PD1`, `PD2`) that detect anomalous behavior in the underlying physical processes and an actuator (`ACT`) to mitigate or isolate the fault effects. The TCD model consists of four fault nodes, $F = \{F1, Fmiss1, Fmiss2, Fstuck\}$, where $F1$ is a physical fault and $Fmiss1, Fmiss2$, $Fstuck$ are cyber faults related to `PD1`, `PD2` and `ACT` respectively. The fault, $F1$ leads to an aberrant behavior indicated by the observable discrepancies, $D = \{D1, D2\}$. The fault effect from $F1$

[1] Since the mappings, Ψ_{act} and Ψ_{ina} are bijective in nature, we use Ψ_{act}^{-1} and Ψ_{ina}^{-1} to map events to nodes.

Table 5.1: Example TCD Model

TCD Element	**Example Model**
Faults (F)	$\{F1, Fmiss1, Fmiss2, Fstuck\}$
Discrepancies (D)	$\{D1, D2\}$
Edges (E)	$\{(F1,D1),\, (D1,D2)\}$
System modes (M)	$\{m1, m2\}$
Fault propagation duration (ET)	$ET(F1,D1) = [2,5]$, $ET(D1,D2) = [1,3]$
Edge-Mode Map (EM)	$EM(F1,D1) = m1$, $EM(D1,D2) = m1, m2$
Edge Uncertainty (ND)	$ND(F1,\, D1) = \perp$, $ND(D1,\, D2) = \top$
Events (Σ)	$\{d1, d2, d1', d2', fstuck, fmiss1,$ $fmiss2, fm1, c, sc\}$
Automaton locations (Q)	$\{S1, S2, S3, S4, S5, S6, S7, S8, S9, S10\}$
Intial Locations (Q_0)	$\{S1, S5, S7\}$
Location-Mode map (Ω)	$\Omega(m1) : (S1 \vee S3)$, $\Omega(m2) : (S2 \vee S4)$
Activation event map (Ψ)	$\Psi(D1) : d1$, $\Psi(D2) : d2$, $\Psi_{act}(F1) : f1$, $\Psi_{act}(Fmiss^1) : fmiss1$, $\Psi_{act}(Fmiss2) : fmiss2$, $\Psi_{act}(Fstuck) : f_{stuck}$
De-activation event map (Ψ_{act})	$\Psi_{act}(D1) : d1'$, $\Psi_{act}(D2) : d2'$
Timing constraints (Φ)	$\{(r)\}$
Transitions (T)	Illustrated in Figure 5.3

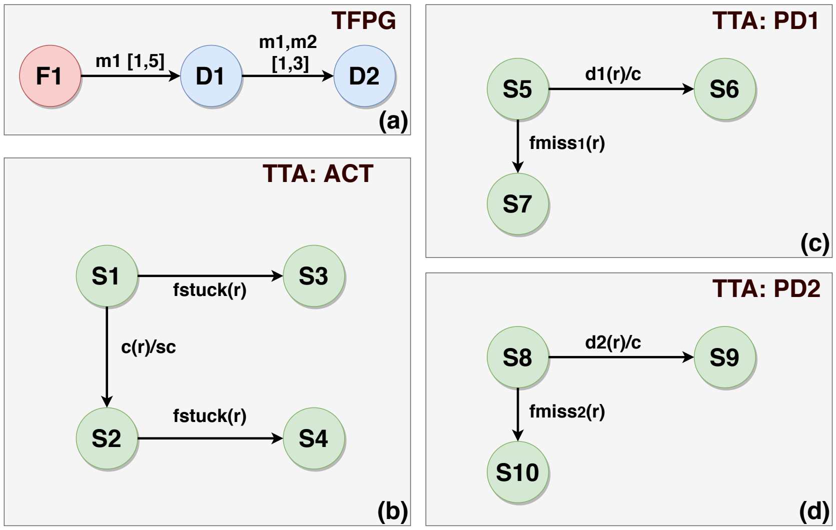


Figure 5.3: Example TCD Model

manifests as $D1$ after the activation of $F1$, indicated by the event $f1$ in system mode $m1$. The manifestation of $D1$ is signaled by the event $d1$ that succeeds $f1$ by a duration [2,5] as highlighted by the markers associated with the edge between $F1$ and $D1$ in Figure 5.3(a). The system mode $m1$ implies the location of the actuator, `ACT` to be either $S1$ or $S3$. Under the same system mode, the fault effect propagates from $D1$ to $D2$ in [1,3] units of time, producing an event $d2$.

The TTA associated with `PD1` consists of three locations with `S5` being the initial location. The automaton models both nominal and faulty operation of the protection device. A missed detection fault, $Fmiss1$ affects the operation, by forcing the automaton to skip the detection of anomalous behavior indicated by the event $d1$. While in `S5`, the automaton checks for the presence of events $d1$ and $fmiss1$ every r units of time. The periodic checking of events is enforced by the timing constraint associated with all outgoing transitions from `S5`. If $fmiss1$ is present then the automaton transitions to `S7`. On the other hand, the presence of the event $d1$ causes the automaton to jump to `S6` and generates an actuation command, indicated by the event c.

The TTA, `ACT`, consists of four locations, with `S1` being the initial location. The automaton models the operation of an abstract actuator that changes the state of the physical process after receiving commands from protection devices. The change in actuator location, signaled by the event, sc leads to change in the system mode affecting the fault propagation. The automaton also captures the behavior of the actuator under the influence of the stuck fault, $Fstuck$ that forces an actuator to ignore commands from the protection devices. While in `S1`, the automaton periodically checks the presence of events, c or $fstuck$. The event, $fstuck$, indicates the presence of the stuck fault. The presence of c forces the actuator to move to `S2` and generate sc state change event. On the other hand, the automaton transitions to `S3` from `S1` or to `S4` from `S2` if $fstuck$ is observed as shown in Figure 5.3(b).

5.3.3 Fault Propagation Semantics

In this section, we describe how fault effect propagates in a TCD model $\mathscr{G}$, while adhering to modal and temporal constraints. To define these constraints, we use two maps $\mathscr{N} : I \times F \cup D \rightarrow \{ON, OFF\} \times \mathbb{R}_+$, and $\mathscr{E} : I \times E \rightarrow \{ON, OFF\} \times \mathbb{R}_+$ to store the state of node and edges, where I is the indexed set of all timestamped events, $F \in \mathscr{G}$ is the set of fault nodes, $D \in \mathscr{G}$ is the set of discrepancies, $E \in \mathscr{G}$ is the set of fault propagation edges. A timestamped event, k is a tuple, (σ, t_1) such that $\sigma \in \Sigma$ is the event label and $t_1 \in \mathbb{R}_+$ is the time instant at which the event occurs. We define two functions, $\mathscr{T} : I \rightarrow \mathbb{R}_+$, $\mathscr{L} : I \rightarrow \Sigma$, that relates an event $k \in I$ to its time of occurrence and event label.

The state of a node is deemed ON, if the fault effect has reached the node, otherwise, remains OFF. Similarly, the state of an edge, $e \in E$ is considered ON if $m \in EM(e)$, where m is the current system mode and EM is the edge mode map. The maps ($\mathscr{N}$, $\mathscr{E}$) also track the time at which the state of each node and edge is changed. For improving the readability, we refer to the state and time attributes associated with a node n after k^{th} event as $\mathscr{N}_k(n)$.`State` and $\mathscr{N}_k(n)$.`time` respectively. Similarly for an edge e, $\mathscr{E}_k(e)$.`State` and $\mathscr{E}_k(e)$.`time` denote state of edge after event k and the time it was last changed.

Every fault effect propagation trace in a TCD model starts with a fault node activation event k, such that $\Psi_{act}^{-1}(\mathscr{L}(k)) \in F$, where $\mathscr{L}(k)$ is the label associated with the time-stamped event $k \in I$. If the associated fault node is cyber in nature, i.e., $\Psi_{act}^{-1}(\mathscr{L}(k)) \in F_{cyb}$ then it can cause transition in one or more protection system automaton. A transition in a protection device can result into generation of synchronizing events that can cause state transitions in other protection devices. These synchronizing events can also lead to generation of actuation commands that change the location of one or more actuator automata. An event, indexed as $l \in I$, related to an actuation command can alter the current system mode from m_i to m_j such that $\Omega(m_i) \rightarrow \perp \big|_{t=\mathscr{T}(l)} \wedge \Omega(m_j) \rightarrow \top \big|_{t=\mathscr{T}(l)}$. The new system mode m_j can enable or disable an edge $e \in E$ after l^{th} event as per Equations (5.1), (5.2) respectively.

$$\mathscr{E}_l(e) \leftarrow (ON, \mathscr{T}(l)) \mid \mathscr{E}_{l-1}(e) = OFF \wedge m_j \in EM(e) \tag{5.1}$$

$$\mathscr{E}_l(e) \leftarrow (OFF, \mathscr{T}(l)) \mid \mathscr{E}_{l-1}(e) = ON \wedge m_j \notin EM(e) \tag{5.2}$$

On the other hand, if the initiating event is related to a physical fault node i.e. $\Psi_{act}^{-1}(\mathscr{L}(k)) \in F_{phy}$ then discrepancy node activation events can take place. The fault effect can propagate from fault node, $n = \Psi_{act}^{-1}(\mathscr{L}(k))$ to a discrepancy node $d \in D$ if the constraint specified in Equation (5.3) holds true

$$\mathscr{N}_k(n).State = ON \wedge \mathscr{N}_k(d) = OFF \wedge (n,d) \in E \wedge \mathscr{E}_k((n,d)).State = ON \tag{5.3}$$

Equation (5.3) ensures the fault effect will propagate to destination node only if the edge between the source and destination nodes is active and the state of the destination node is inactive. A discrepancy activation event with index $p \in I$ will[2] happen in the interval defined in Equation (5.4)

$$\mathscr{T}(p) \leftarrow [ET(n,d).t_{min}\,, ET(n,d).t_{max}] + \max(\mathscr{E}_k(n,d).time\,, \mathscr{N}_k(n).time) \tag{5.4}$$

The activation of discrepancy node can lead to further activation of other discrepancy nodes as per the constraints mentioned in Equations (5.3) and (5.4). It can also cause transitions in one or more protection device automata that can generate new events leading to more transitions in protection devices and actuators. As stated previously, actuator location change can alter the system mode, which can disable existing active edges or enable new edges according to Equations (5.1) and (5.2). Finally, the change in the state of edges can alter the future activation of discrepancy nodes as per Equation (5.4).

[2] If $ND(n,d)$ is $\perp$ then p is guaranteed to be observed within the duration mentioned in Equation (5.4) otherwise its uncertain

5.3.4 Execution Semantics

In this section, we describe the execution semantics of a TCD model using a discrete model of time, i.e., the time advances in discrete steps. To define the execution rules, we translate TFPG sub model and all TTA to a common model of computation, Timed Automata [138], such that a TCD model, $\mathscr{G}$ is transformed into a network of timed automata, $\mathscr{G}_{X^r}$, where each automaton is synchronized to an external clock source, *Ticker*.

A network of timed automata is the parallel composition, $A_1 \,|\, A_2 ... \,|\, A_n$ of a set of timed automata, A_1 , A_2 , ... , A_n. Communication between the individual automaton occurs in two ways via (1) handshake synchronization using actions and (2) shared variables. To model handshake synchronization between automata, an event $\sigma \in \Sigma$ associated with a transition is replaced by an action pair (σ?, σ!) where σ? implies event generation action and σ! denotes consumption action. The synchronization is achieved by forcing generation and consumption actions to occur simultaneously i.e transitions in different automata with action labeled σ! and σ? are taken simultaneously. The second method uses two functions, `register_occurrence(event_id)` and `check_occurrence(event_id)`, where the former explicitly stores the presence and the later checks for the occurrence of an event with a unique identifier, *event_id*. We consider handshake communication as a *strict* form of synchronization since it happens instantaneously without any time delay whereas communication via shared variables can be relayed in the next cycle depending upon the order of execution and therefore deemed as *loose*.

Apart from clock source, transformed TCD model, $\mathscr{G}_{X^r}$ requires *Injector* automaton to introduce external events such as fault activation events, *Mode Calculator* automaton to implement Ω and finally node *De-activator* automaton to signal the discrepancies that should no longer be active in the current system mode. Figure 5.4 shows the structure of the translated TCD model, $\mathscr{G}_{X^r}$ and its interfacing with external automata. The external clock source uses strict communication mode to synchronize time with $\mathscr{G}_{x^r}$ and injector automata. A fault edge automaton uses handshake synchronization to convey activation of discrepancies to other fault edge automata but relies on shared variable to relay the same update to protection system. Similarly, protection system automata (protection device and actuator) uses loose synchronization mode to send and receive updates among each other but synchronizes with De-activator and Mode Calculator through handshake actions.

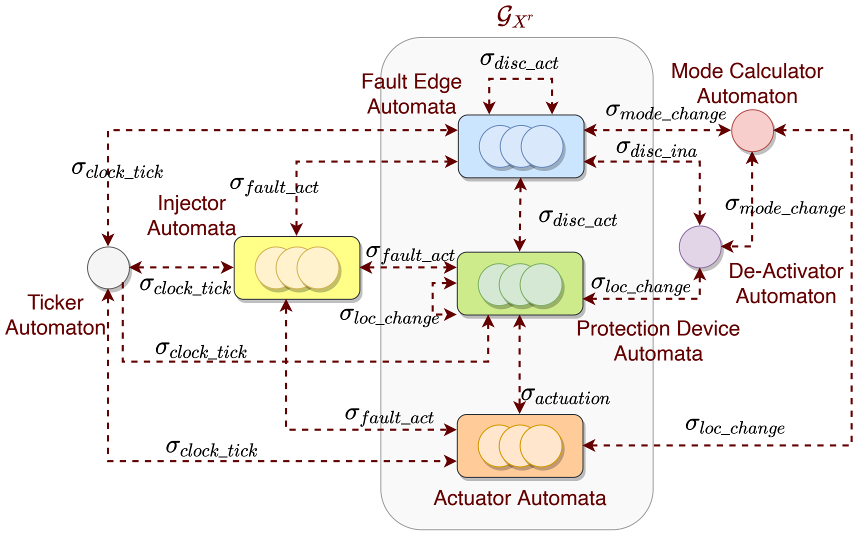


Figure 5.4: Structure of translated TCD model and its interfacings

The following sub-sections describe the UPPAAL [139] timed automata (TA) templates for a fault edge, clock source, mode calculator and de-activation automata along with translation procedure for converting an arbitrary TTA model to UPPAAL TA. An UPPAAL TA is an extended implementation of TA model of computation that allows transition guards to be defined on discrete variables as well as updating state variables when a transition is taken from one location to other. The UPPAAL transition tr from location q_i to q_j is a tuple, $< q_i, g, \sigma_k! \, \text{or} \, \sigma_k?, r, q_j >$ where g is a boolean constraint defined over discrete variables and (or) clock variables, $\sigma_k!$ or $\sigma_k?$ is the synchronization action (generation or consumption) and r is the set of assignment statements over discrete and (or) clock variables.

5.3.4.1 Fault Edge Automaton

The TFPG sub-model is a specification that put modal and timing constraints on the propagation of fault effect. To actually simulate or verify the fault effect propagation, we represent each edge as an UPPAAL TA. Figure 5.5 shows the timed automaton template of an arbitrary fault propagation edge, $e \in E$ with $EM(e)$ be the set of system modes in which the edge is active and (t_max, t_min) = $ET(e)$ is the duration of upper and lower bound on the propagation time. The corresponding node activation and de-activation events for source and destination nodes are $(src\_act$, $src\_ina)$ and $(dst\_act$, $dst\_ina)$ respectively. Other template arguments include a unique edge identifier, $edge\_id$,

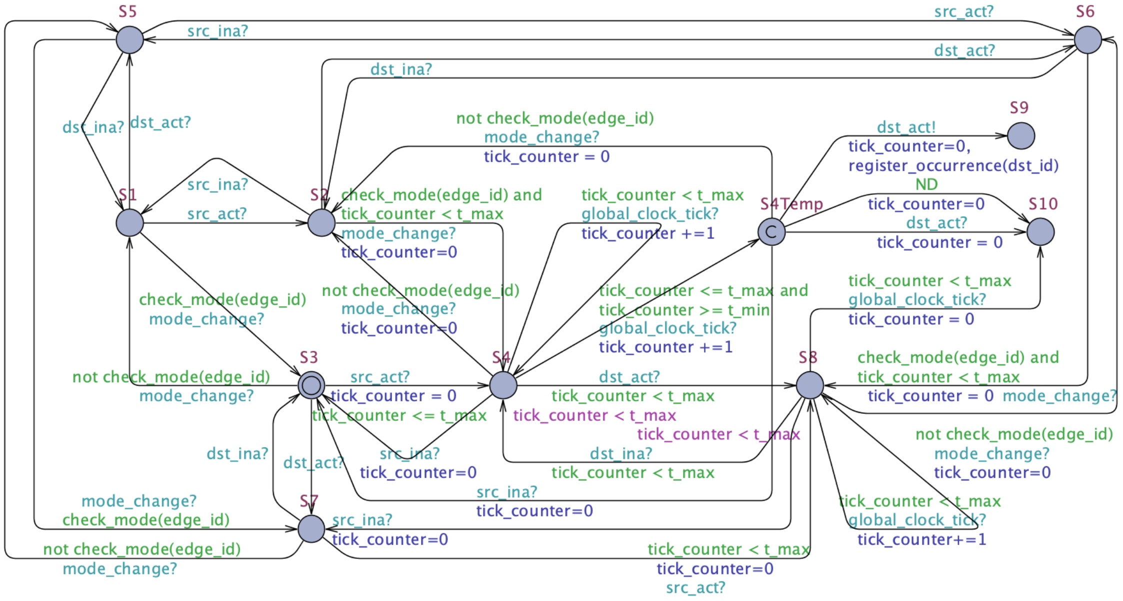


Figure 5.5: UPPAAL Timed automaton templates for Fault Edge

boolean parameter *ND* to capture uncertainty associated with the edge and lastly, an event identifier associated with activation of destination node, *dst_id*.

The fault edge automaton consists of 11 locations with S3 being the initial location, based on the assumption that the initial system mode belongs to the set $EM(e)$, i.e., the edge is active and state of all TFPG nodes is inactive[3]. The automaton can transition to locations S7 or S4 or S1 depending upon the event observed as shown in Figure 5.5. The transition, S3 → S7 is taken if the destination node becomes active i.e. the event *dst_act* is observed, whereas if source node becomes active, the automaton moves to S4. The location S1 is selected if the edge becomes inactive as a result of mode change event, *mode_change*. Whenever, the current system mode is changed, Mode Calculator generates a *mode_change* event and every fault edge automaton responds to the event by calling a function check_mode(edge_id) to ascertain if the edge remains active in the new system mode. The function return true if the current mode is listed in $EM(e)$ otherwise false.

Similarly rest of the locations in the automaton reacts to these events and transition to different locations as highlighted in Figure 5.5. Table 5.2 summarises the physical meaning of each location based on four conditions, (1) Is destination node active?, (2) Is edge active?, (3) Is source node active? and (4) Has the edge fired? We have assumed persistent faults in this study, which implies

[3] If the assumption is not valid for system, then an appropriate location can be selected based on Table 5.2

Table 5.2: Fault edge automaton location interpretation

Location	Edge Fired?	Dest. Active?	Edge Active?	Source Active?
S1	False	False	False	False
S2	False	False	False	True
S3	False	False	True	False
S4	False	False	True	True
S5	False	True	False	False
S6	False	True	False	True
S7	False	True	True	False
S8	False	True	True	True
S9	True	X	X	X
S10	True	X	X	X

a fault edge can fire only once. As a result of this assumption, the locations `S9` and `S10` have no outgoing transitions. According to Table 5.2, both locations represent the edge has fired. However, `S9` denotes the edge has fired while signalling the activation of destination node, whereas `S10` does not.

While in `S4(S8)`, the automaton counts the number of ticks received from external clock source, *Ticker* using a bounded local integer variable, *tick_counter* as shown in Figure 5.5. While the value of *tick_counter* is less than *t_min*, the automaton takes the self transition and increments the counter at every tick. However, after *tick_counter* becomes equal to *t_min* then the transition `S4` → `S4Temp` (`S8` → `S10`) also becomes enabled and automaton randomly decides whether to take the self transition or move to `S4Temp` (`S10`) at every clock tick. The self transition is feasible till the location invariant, $tick\_counter < t\_max$, associated with `S4` (`S8`) is valid. At the next tick, the automaton has to take the transition to `S4Temp` (`S10`). The location `S4Temp` is an intermediate committed location, that is used to check certainty parameter before generating *dst_act* event i.e either transition to `S9` or `S10`.

5.3.4.2 TTA to TA Translation

The central idea of translating a TTA to an UPPAAL TA is to add a set of intermediate locations that imply satisfaction of timing constraints. These locations allow automaton to check the enabling condition of an outgoing transition at discrete time steps while adhering to the timing constraints. Algorithm 1 outlines sequence of steps required to translate a location s in TTA model to its equivalent UPPAAL TA representation. The algorithm expects the label of the location, s along with a set of all outgoing transitions T. The output of the algorithm is a tuple, where the first element is a set of locations, S that contains the original location label, s and $p+1$ intermediate locations where p is the number of unique timing constraints associated with transitions in T. We use `tr.dst`, `tr.constraint`, `tr.ip_event`, `tr.op_event` to refer to destination node label, timing constraint, input and output synchronization event labels associated with a TTA transition, $tr \in T$. For a timing constraint k associated with a transition $tr \in T$, the expression `k.type` and `k.value` refer to type and value attributes of k. We also define two functions, `Transition`() and `Location`() where the former creates an UPPAAL transition that takes source location, guard condition, synchronization action, update statements and destination location as input arguments. The function `Location` creates a location of type *normal*, *urgent* or *committed*, where type parameter is passed as an input argument.

The algorithm begins by adding s to S (Line 3) and initializing three maps, Ctr, $Flag$ and Loc where key is the timing constraint (Lines 3-9). The map Ctr stores variable labels, $Tick\_counter\_$`<i>` that are used to count clock ticks for evaluating timing constraints. The map Loc stores reference to boolean variables $CheckFlag\_$`<i>` that are used to restrict evaluation of instantaneous timing constraints and the Loc map stores state labels of newly created urgent locations for every unique timing constraint. These urgent locations represent the satisfaction of timing constraints. One more urgent location is created and referred as $temp$ (Line 10) which implies the reception of clock tick from the global clock source. After adding these intermediate locations to the output set S (Lines 11-14), transitions between these locations are created (Lines 15-17) using function defined in algorithms 2, 3 and 4.

The transition from s to $temp$ happens at the reception of synchronizing event $''global\_clock\_tick?''$ and increments all clock tick counting variables as shown in (Line 12 in algorithm 2). Whereas the transition, $s \leftarrow temp$ occurs only if none of the timing constraint are satisfied at the current time instant (Line 13 in algorithm 2). A pair of transitions are added between $temp$ and every location in Loc (Lines 15-16) such that the transition from $temp \rightarrow Loc[k]$ implies the timing constraint k has satisfied whereas the transition in reverse direction represents the un-satifiability of

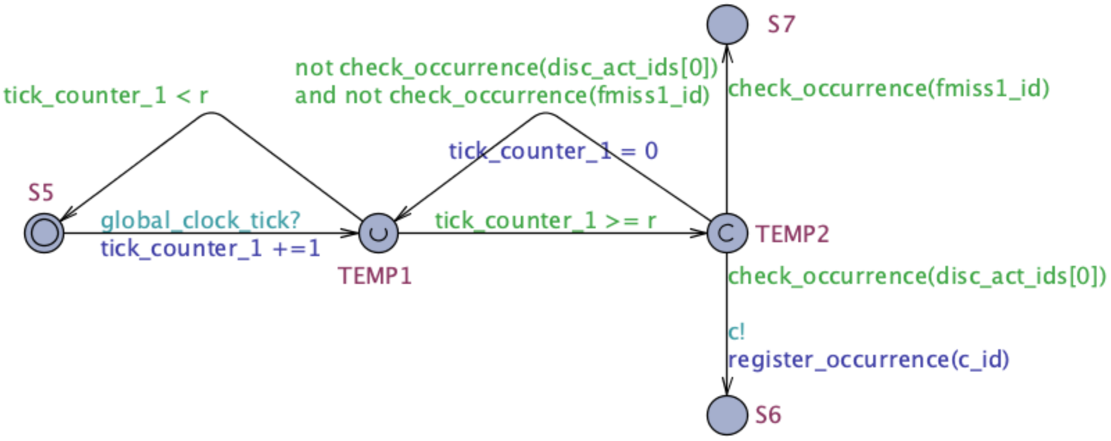

S7
not check_occurrence(disc_act_ids[0])
and not check_occurrence(fmiss1_id)
check_occurrence(fmiss1_id)
tick_counter_1 < r
tick_counter_1 = 0
S5
global_clock_tick?
tick_counter_1 +=1
TEMP1
tick_counter_1 >= r
TEMP2
check_occurrence(disc_act_ids[0])
c!
register_occurrence(c_id)
S6

Figure 5.6: UPPAAL TA model of protection device, `pd1`

transition $tr \in T$ with constraint k . The un-satisfaction of the transition is captured by the function call `check_occurrence`(event_id) that checks for observance condition of event with that unique identifier and resets the clock tick counter in case the function returns false. For instantaneous timing constraints, an extra condition related to boolean variable, $Flag(k)$ is added to the transition $temp \leftarrow Lock(k)$ as described in algorithm 3 (Lines 11-14). Finally, for every transition $tr \in T$, a corresponding transition is added between the intermediate location $Loc(tr.\texttt{constraint})$ and $tr.$`dst` with guard condition related to observance of the $tr.$`ipevent` and updating all clock tick counters to 0 along with generating $tr.$`op_event` synchronizing event as highlighted in algorithm 4 (Lines2-10). Figures 5.6, 5.7 and 5.8 shows the translated UPPAAL TA associated with TTA model of protection devices (example TCD model) described in previous section.

5.3.4.3 Clock Source Automaton

Figure 5.9 shows an UPPAAL TA of the global clock source, referred as *Ticker* with a single location `S1`. Ticker periodically resets clock variable, *time*, and broadcasts a synchronizing event, *global_clock_tick*. The automaton stays in the location till, $time < 1$ and at $time = 1$, the self transition is enabled and the location invariant, $time \leq 1$ enforces the automaton to take the enabled transition resulting in a broadcasting event, resetting clock variable and increasing the global clock counter, *global_counter* by calling function `increment_global_counter()`.

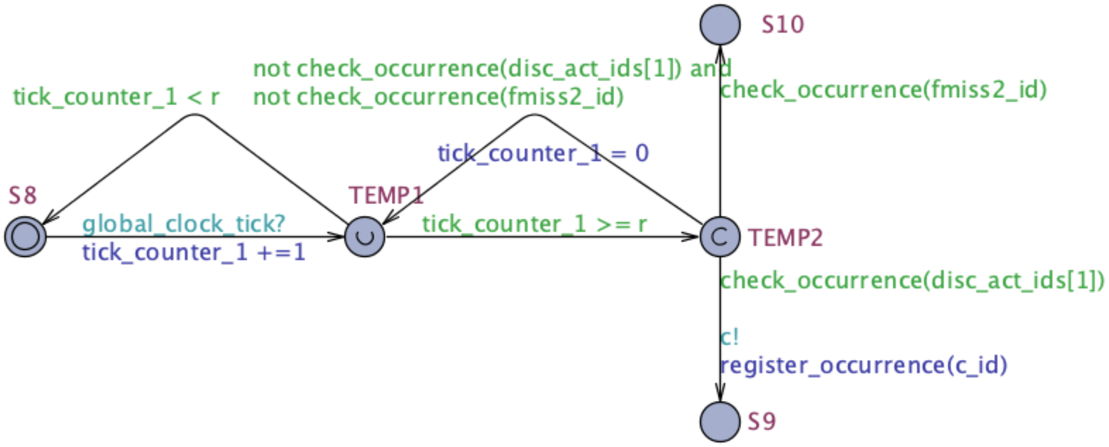

S10
not check_occurrence(disc_act_ids[1]) and
not check_occurrence(fmiss2_id)
check_occurrence(fmiss2_id)
tick_counter_1 < r
tick_counter_1 = 0
S8
TEMP1
global_clock_tick?
tick_counter_1 +=1
tick_counter_1 >= r
TEMP2
check_occurrence(disc_act_ids[1])
c!
register_occurrence(c_id)
S9

Figure 5.7: UPPAAL TA model of protection device, `pd2`

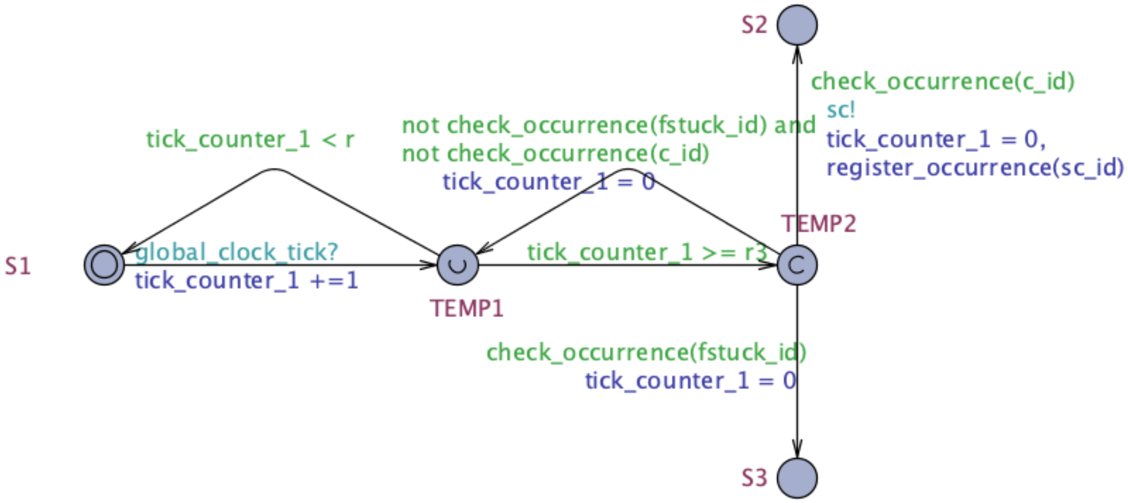


Figure 5.8: UPPAAL TA model of actuator, `act`

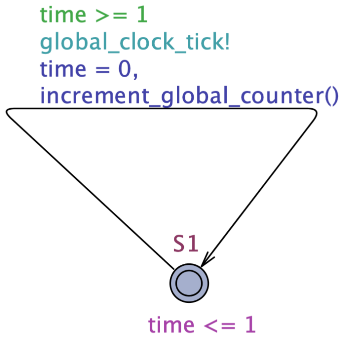


Figure 5.9: Clock source UPPAAL TA model

Algorithm 1: TTA to UPPAAL TA translation

```
Input: $s$, $T$
Output: $S$, $T'$
Initialize: $S \leftarrow s$, $Ctr \leftarrow \varnothing$, $Flag \leftarrow \varnothing$, $Loc \leftarrow \varnothing$
begin
    $i \leftarrow 1$
    foreach $tr \in T$ do
        $Ctr[tr.constraint] \leftarrow'' Tick\_counter\_\$i\$''$
        $Flag[tr.constraint] \leftarrow'' CheckFlag\_\$i\$''$
        $Loc[tr.constraint] \leftarrow$ Location(”urgent”)
        $i \leftarrow i + 1$
    end
    $temp \leftarrow$ Location($''urgent''$)
    $S \leftarrow S \cup temp$
    foreach $k \in Loc$ do
        $S \leftarrow S \cup Loc[k]$
    end
    $T_1 \leftarrow$ createTickerTransitions($s, temp, Ctr, Flag$)
    $T_2 \leftarrow$ createTimeConstTransitions($temp, Loc, Ctr, Flag$)
    $T_3 \leftarrow$ createDestinationTransitions($T, Loc, Ctr, Flag$)
    $T' \leftarrow T_1 \cup T_2 \cup T_3$
end
```

Algorithm 2: Function: createTickerTransitions

```
Function createTickerTransitions
    Input: $s$, $temp$, $Ctr$, $Flag$
    Output: $T'$
    begin
        $u, g, T' \leftarrow \varnothing$
        foreach $k \in Ctr$ do
            $u[k] \leftarrow '' \$Ctr[k]\$ += 1''$
            $c \leftarrow '' \$Ctr[k]\$ < \$k.\mathtt{value}\$''$
            if $k.\mathtt{type} == ''Instantaneous''$ then
                $c$.append($''$ or not $\$Flag[k]\$''$)
            end
            $g \leftarrow ''(\$c\$)$ and $\$g\$''$
        end
        $T' \leftarrow T' \cup$ Transition($s, \varnothing,$ “global_clock_tick?”$, u, temp$)
        $T' \leftarrow T' \cup$ Transition($temp, g[: -3], \varnothing, \varnothing, s$)
    end
end
```

Algorithm 3: Function: createTimeConstTransitions

```
Function createTimeConstTransitions
    Input: $temp$, $Loc$, $Ctr$, $Flag$
    Output: $T'$
    begin
        foreach $k \in Ctr$ do
            $c1 \leftarrow ``\$Ctr[k]\$ \geq \$k.\texttt{value}\$''$
            $u2 \leftarrow ``\$Ctr[k]\$ = 0'' \; c2 \leftarrow \varnothing$
            foreach $tr \in T$ do
                if $k == tr.constraint$ then
                    $c2.\texttt{append}(``\textit{not check\_occurrence}(\$tr.\texttt{ip\_event}\$)\ \textit{and}'')$
                end
            end
            if $k.\texttt{type} == ``\textit{Instantaneous}''$ then
                $c1.\texttt{append}(``\textit{and}\ \$Flag[k]\$'')$
                $u2.\texttt{append}(``\$Flag[k]\$ = false'')$
            end
            $T' \leftarrow T' \cup \texttt{Transition}(temp, c1, \varnothing, \varnothing, Loc[k])$
            $T' \leftarrow T' \cup \texttt{Transition}(Lock[k], c2[: -3], \varnothing, u2, temp)$
        end
    end
end
```

Algorithm 4: Function: createDestinationTransitions

```
Function createDestinationTransitions
    Input: $T$, $Loc$, $Ctr$, $Flag$
    Output: $T'$
    begin
        foreach $tr \in T$ do
            foreach $k \in Ctr$ do
                $u[k] \leftarrow ``\$Ctr[k]\$\ =\ 0''$
            end
            $u.\texttt{push}(``\textit{register\_occurrence}(\$tr.\texttt{op\_event}\$)'')$
            $c \leftarrow ``\textit{check\_occurrence}(\$tr.\texttt{ip\_event}\$)''$
            $T' \leftarrow T' \cup \texttt{Transition}(Loc[tr.\texttt{constraint}], c, tr.\texttt{op\_event}, u, tr.\texttt{dst})$
        end
    end
end
```

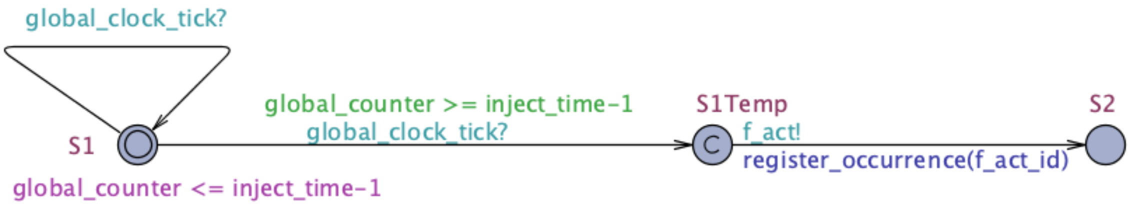

global_clock_tick?
global_counter >= inject_time-1
global_clock_tick?
S1Temp
S2
S1
f_act!
register_occurrence(f_act_id)
global_counter <= inject_time-1

Figure 5.10: Certain Injector UPPAAL TA

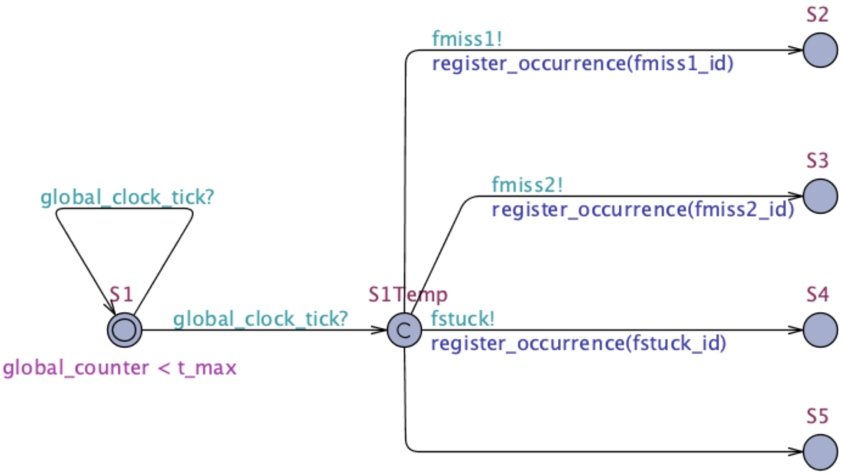


Figure 5.11: Un-certain Injector UPPAAL TA

5.3.4.4 Injector Automaton

Injector automaton is responsible for injecting external events in the system . These events can be related to introduction of physical and cyber faults. We have used two types of injector automata, 1) Certain Injector and 2) Un-certain Injector. The certain injector, as the name implies, produces the specified event at a given time instant as shown in Figure 5.10 shows a UPPAAL TA template for an injector automaton that injects fault f_act at $inject\_time$ clock ticks. The un-certain injector automaton may inject an event out of collections at any time in the duration [0, t_max]. The automaton shown in Figure 5.11 injects events related to one of the cyber fault in the example TCD model at any time in the range [0, t_max].

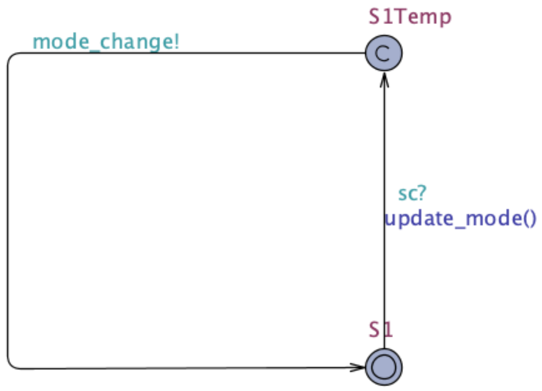


Figure 5.12: Mode Calculator UPPAAL TA

5.3.4.5 Mode Calculator Automaton

This automaton responds to the events related to change in the location of the actuators. Figure 5.12 shows mode calculator automaton for the example tcd model, consisting of two locations with `S1` being the initial location. When the automaton receives an actuator state change event, *sc_*`<i>`, it moves from `S1` to `S1Temp` while updating the system mode variable through a function call, `update_mode`(). The function iterates over every possible system mode and selects the mode, m for which $\Omega(m)$ evaluates to true. The automaton transitions back to `S1` after updating the system mode variable and generating *mode_change* event.

5.3.4.6 Discrepancy De-activator Automaton

The protection devices cause the actuators to change their location in response to the observed fault effects. The actuator location change can alter the system mode resulting in masking of the fault effects leading to de-activation of discrepancy nodes. Figure 5.13 shows a generic discrepancy de-activator automaton template consisting of two locations with `S1` being the initial location. The automaton jumps to committed location, `S1Temp` after observing a mode change event. While in `Temp1`, the automaton iterates over every discrepancy node and generates de-activation event, σ, for a discrepancy, $\Psi_{ina}^{-1}(\sigma)$, if all fired fault edges leading to that node have become in-active. The condition is checked by the function call, `check_disc_status`() as shown in Figure 5.13.

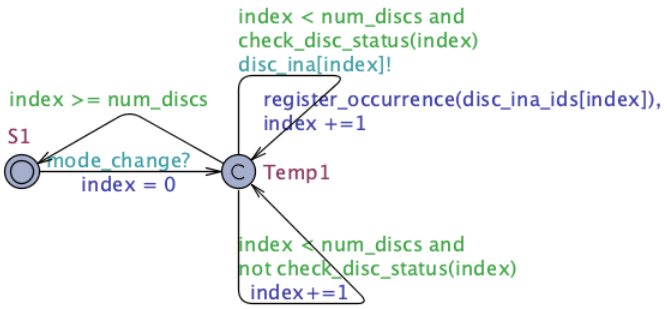


Figure 5.13: De-activator UPPAAL TA

5.4 Evaluation

In this section, we evaluate the correctness of execution and fault propagation semantics of the TCD model with respect to certain requirements. These requirements are necessary conditions that stem from the semantics of the TCD modeling formalism. We encode these requirements in the form of a Timed computation tree logic (TCTL) formulaes or properties and verify these properties using concrete examples with the help of symbolic model checking tool, UPPAAL (see Appendix G for TCTL grammar definition). In the following sub-sections, we describe the requirements for the fault edge automaton and TTA translation followed by the respective TCTL formulaes. In the end, we show multiple simulation traces of the CPS example, described in Table 5.1, produced using UPPAAL's inbuilt simulator.

5.4.1 Fault Edge Automaton

The fault edge automaton must satisfy the following requirements to guarantee the correctness of the fault propagation semantics as described in equation (5.4).

(R1) The minimum (maximum) signaling time of a discrepancy associated with a certain edge is equal to the sum of minimum (maximum) time for the fault effect to propagate over the edge and the latest time by which both edge and source node became active.

(R2) An active discrepancy can not signal more than once.

(R3) A discrepancy associated with an uncertain edge is not always expected to signal.

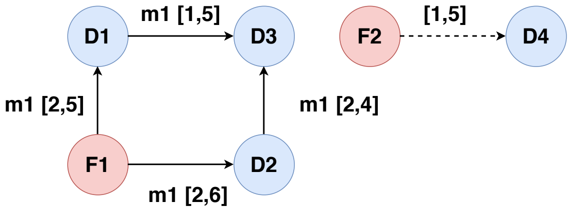


Figure 5.14: TFPG with five edges: e1-e5

(**R4**) There is no deadlock.

These requirements are evaluated in context of an arbitrary TFPG, with two fault nodes, (F1, F2), four discrepancy nodes, (D1, D2, D3, D4) and five fault edges as shown in Figure 5.14. The complete system under test includes, two fault injector automata (p1, p2), a mode change event injector (mc), five fault edge automata (e1-e5) and a global clock ticker to which rest of the system is synchronized. The fault injector automaton activates a fault node at a fixed time, `inject_time`. However, the mode change injector changes the system mode at any time instant bounded by the template parameter, `t_max`.

We define seven TCTL formulaes for the given system that translates to the requirements mentioned above. These formula-es are described in more details as follows

(**P1**) `A[] not deadlock`:

The satisfaction of this formula assures the fulfilment of the requirement **R4**, i.e. there is no deadlock in the system.

(**P2**) `A<> (e5.S9 or e5.S10)`:

The satisfaction of this formula assures the fulfilment of the requirement **R3**, where e5 is an uncertain edge, i.e. ND(e5) = true, and the location S9 (S10) implies the associated discrepancy, D5, is (not) signaled. The property shows that the fault edge automaton reaches S9 or S10 location eventually in all paths.

(**P3**) `e4.S9 --> e3.S10`:

This property is related to the requirement **R2**, which implies if edge e4 signals the discrepancy, then e3 cannot.

(**P4**) `e3.S9 --> e4.S10`:

This property is also related to the requirement **R2**, which states if edge `e3` signals the discrepancy, then `e4` cannot.

(P5) `A[] not(e4.S9 and e3.S9)`:

The satisfaction of this property signifies the fulfilment of the requirement **R2** which implies that there does not exist a state in which both edges signaled the discrepancy, `D5`.

(P6) `mode_change_time > faults_time[0] and mode_change_time > 0 --> disc_act_times[0] >= mode_change_time + e1.t_min and disc_act_times[0] <= mode_change_time + e1.t_max`:

This property signifies that if `F1` is injected before the edge becomes active, then D1 will become active in the range, `mode_change_time` + [`e1.t_min`, `e1.t_max`], where `mode_change_time` is the time at which mode change injector automata (`mc`) changes the mode of the system to *m1*, `fault_time[0]` corresponds to the time at which `F1` is injected in the system by fault injector automaton (`p1`), `disc_times[0]` stores the time at which `D1` becomes active and (`e1.t_min`, `e1.t_max`) denote the minimum and maximum fault propagation time for edge `e1`. It partially fulfils the recommendation **R1** i.e. covers only one case when edge becomes active after the fault is injected.

(P7) `faults_time[0] > mode_change_time and mode_change_time > 0 --> disc_act_times[0] >= faults_time[0] + e1.t_min and disc_act_times[0] <= faults_time[0] + e1.t_max`:

This property covers the second case of the requirement **R1** i.e. edge becomes active before the fault is injected.

5.4.2 TTA to UPPAAL TA Translation

A translated UPPAAL TA should adhere to the following requirements to satisfy the execution semantics of TTA.

(R1): An instantaneous timing constraint is evaluated only once.

(R2): A periodic timing constraint is evaluated at a fixed rate (specified).

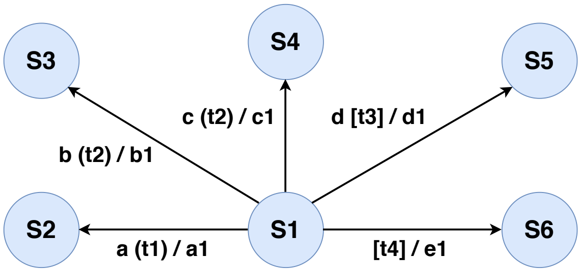


Figure 5.15: Time triggered automaton: `g`

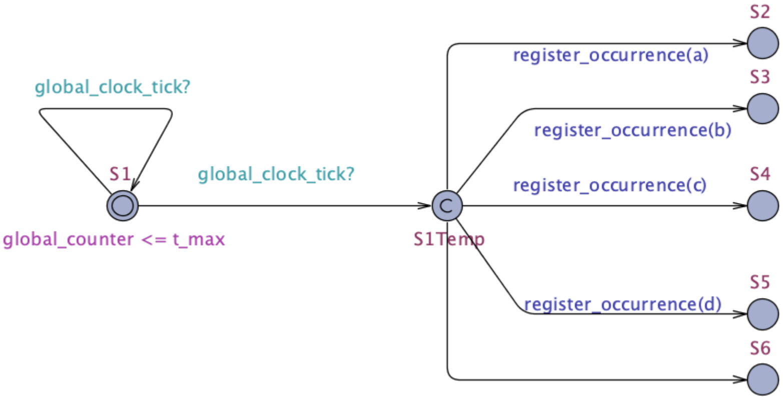


Figure 5.16: Event Injector Automaton: `p1`

(**R3**): The time difference between a transition becoming active and actually taken is bounded by the timing constraint.

(**R4**): There is no deadlock.

Similar to propagation semantics of fault edge, execution semantics are evaluated in context of an arbitrary TTA, `g` as illustrated in Figure 5.15. The close system to be analysed consists of translated UPPAAL TA model (`gxr`) and an event injector (`p1`) as shown in Figures 5.16 and 5.17. The event injector automaton may inject one of the four events *a*, *b*, *c* or *d* at any clock tick between duration [`0`, `t_max`].

We define four TCTL formulae or properties for the system at hand which translates to requirements mentioned above. These properties are described in more detail as follows:

(**P1**): `A[] not deadlock`:

The satisfaction of this formula assures the fulfilment of the requirement **R4**, i.e. there is no deadlock in the system.

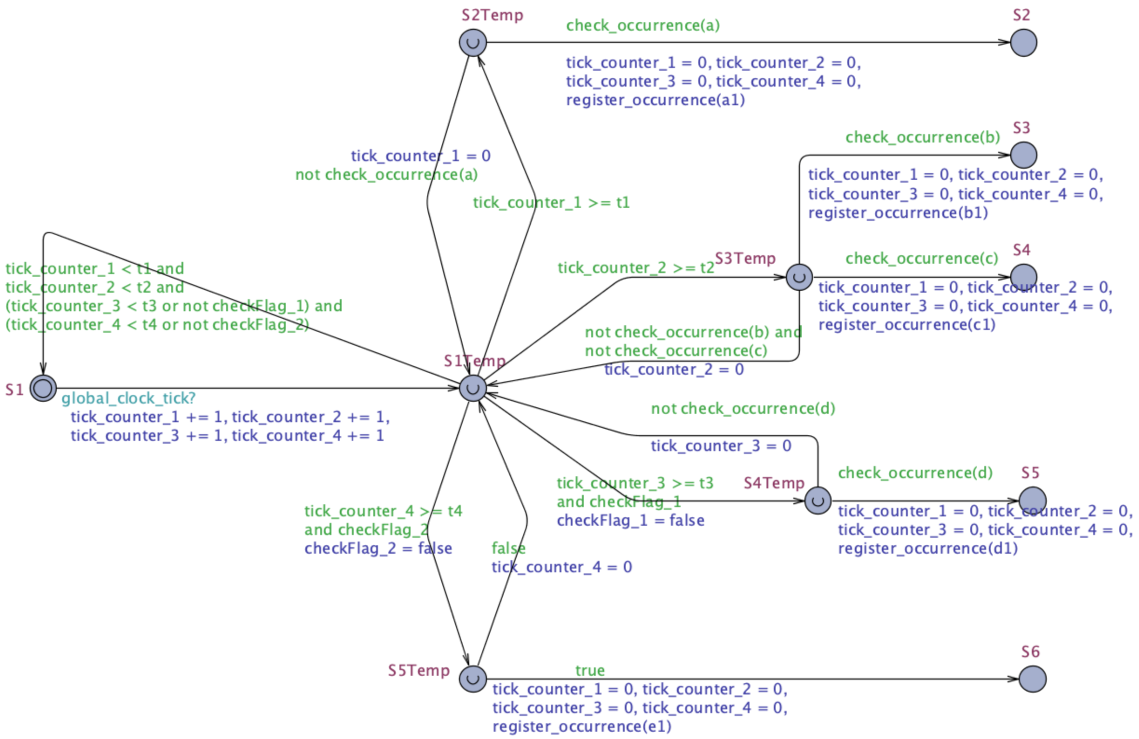


Figure 5.17: Translated UPPAAL TA: `gxr`

(P2): `p1.S5 and disc_act_times[3] > t3 --> gxr.S6 and alarm_act_vars[4] and alarm_act_times[4] == t4`

This property is related to requirements **R1** and **R4**. It states that if event d is injected after time `t3` then `gxr` will transition to location `S6` and generate event $e1$ by time `t4`. The pair of arrays `disc_act_vars`, `disc_act_times` indicate the state and activation time of events a, b, c, d, indexed from 0 to 3. Similarly, `alarm_act_vars`, `alarm_act_times` corresponds to the events $a1$, $b1$, $c1$, $d1$ and $e1$ indexed from 0 to 4. This property implies that the transition, $< S1, d, t3, d1, S5 >$ is evaluated only once. Since the event is produced after `t3`, the transition from location `S1` to `S5` never happens.

(P3): `p1.S5 and disc_act_times[3] <= t3 --> gxr.S5 and alarm_act_vars[3] and alarm_act_times[3] == t3`

This property is related to requirements **R1** and **R4**. It states that if event d is injected by time `t3` then `gxr` will transition to location `S5` and generate event $d1$ by time `t3`.

(P4): `p1.S4--> gxr.S4 and alarm_act_vars[2] and alarm_act_times[2]`

```
- disc_act_times[2] < t2 and alarm_act_times[2] -
disc_act_times[2] >= 0
```

This property is related to **R2** and **R3**. It states that if event c is injected at any time in the system then TTA will transition to `S4` and generate event $c1$ by the time bound `t2`. This property implies the periodic constraints are checked at fixed rate.

5.4.3 Simulation Results

Now, we show simulation traces of the example TCD model described in section 5.3.2 involving physical and cyber (missed detection) faults. The complete system consists of 10 timed automata, a global clock ticker `gc` (Figure 5.9), mode change calculator `mc` (Figure 5.12), discrepancy de-activator `de` (Figure 5.13), physical fault injector `ip` (Figure 5.10), cyber fault injector `ic` (Figure 5.11), two fault edges `f1_d1`, `d1_d2` (Figure 5.5), two protection devices `pd1`, `pd2` and an actuator `act` (Figures 5.6, 5.7 and 5.8). The injector automaton, `ip` is of deterministic type i.e. it injects the fault `F1` at specified clock tick while the other injector automaton, `ic` is non-deterministic as it may inject `Fmiss1`, `Fmiss2` or `Fstuck` at any clock tick between duration [0, `t_max`]. The value of template parameters `r`, `t_max` and `inject_time` used in the simulation is 1, 10 and 5 respectively. Tables 5.3 and 5.4 summarise the traces of three different scenarios that are described in more detail in the following sub-sections.

5.4.3.1 Scenario 1: Physical Fault Only

Initially the system mode is $m1$ and the state of the system is represented by a tuple $<$ S1, S1, S1, S1, S1, S3, S3, S5, S8, S1 $>$ as indicated in the first row of Table 5.3. At t=2, the injector automaton, `ic`, transitions to S5 without injecting any cyber fault. After 3 clock ticks, i.e., at t=5, the injector automaton `ip`, transitions to S2 while generating f_act event implying the injection of the physical fault F1. The event f_act causes the automaton `f1_d1` to change its state from S3 to S4, signifying both source node and edge are active. At t=7, the edge signals the activation of discrepancy D1 by transitioning to S9. The generation of event, $disc\_act[0]$ i.e. $d1$ forces the succeeding edge, `d1_d2` to jump to location S4. In the same time step, the protection device `pd1` observes the occurrence of $d1$ and transitions to S6 while producing actuation command c. The

Table 5.3: Simulation trace for scenario 1: Only physical fault

Global counter	System mode	gc	mc	de	ip	ic	f_d1	d1_d2	pd1	pd2	act
0	m1	S1	S1	S1	S1	S1	S3	S3	S5	S8	S1
1	m1	S1	S1	S1	S1	S1	S3	S3	S5	S8	S1
2	m1	S1	S1	S1	S1	S5	S3	S3	S5	S8	S1
3	m1	S1	S1	S1	S1	S5	S3	S3	S5	S8	S1
4	m1	S1	S1	S1	S1	S5	S3	S3	S5	S8	S1
5	m1	S1	S1	S1	S2	S5	S4	S3	S5	S8	S1
6	m1	S1	S1	S1	S2	S5	S4	S3	S5	S8	S1
7	m2	S1	S1	S1	S2	S5	S9	S1	S6	S8	S2
8	m2	S1	S1	S1	S2	S5	S9	S2	S6	S8	S2

actuator act detects the presence of c and transitions to S2 and creates a change of status event sc. The alteration in the status of the actuator forces the mode calculator automaton mc to update the system mode to $m2$ and generate a mode change event, $mode\_change$. This mode change event forces the $d1\_d2$ edge to change its location from S4 to S2. The discrepancy de-activator also responds to the mode change event and creates a $disc\_ina[0]$ i.e., $d1'$ event, forcing the d1_d2 to finally change the location to S1 implying both source node and edge are in-active. Table 5.3 shows the final state of all the automatons at each clock tick. However, it does not list all intermediate locations each automaton goes through to reach the final location due to limitation of space. For complete list of state transitions, the interested reader can download the complete trace files and associated Gnatt chart from the github repository [140].

5.4.3.2 Scenario 2: Physical and Cyber Faults

The initial system mode and state of the system is m1, $<$ S1, S1, S1, S1, S1, S3, S3, S5, S8, S1 $>$ as highlighted in Table 5.4. Contrary to previous trace, in this scenario, ic injects a missed detection fault, $Fmiss1$ in pd1 and transitions to location S2. In the same time step, pd1 detects the occurrence of event $fmiss1$ and transitions to location S7. AT t=7, the edge automaton $f1\_d1$ jumps to S9 while emitting $d1$ ($disc\_act[0]$) event, signaling the activation of D1 forcing the $d1\_d2$ to jump to S4. Since the protection device pd1 is in S7, it ignores the discrepancy activation event. At t=9,

Table 5.4: Simulation trace for scenario 2: Physical and Cyber Faults

Global counter	System mode	`gc`	`mc`	`de`	`ip`	`ic`	`f_d1`	`d1_d2`	`pd1`	`pd2`	`act`
0	m1	S1	S1	S1	S1	S1	S3	S3	S5	S8	S1
1	m1	S1	S1	S1	S1	S1	S3	S3	S5	S8	S1
2	m1	S1	S1	S1	S1	S2	S3	S3	S7	S8	S1
3	m1	S1	S1	S1	S1	S2	S3	S3	S7	S8	S1
4	m1	S1	S1	S1	S1	S2	S3	S3	S7	S8	S1
5	m1	S1	S1	S1	S2	S2	S4	S3	S7	S8	S1
6	m1	S1	S1	S1	S2	S2	S4	S3	S7	S8	S1
7	m1	S1	S1	S1	S2	S2	S9	S1	S7	S8	S1
8	m1	S1	S1	S1	S2	S2	S9	S2	S7	S8	S1
9	m1	S1	S1	S1	S2	S2	S9	S9	S7	S9	S2

the edge `d1_d2` signals the activation of D2 by registering the event *disc_act*[1] and transitions to S9. The protection device `pd2` observes the occurrence of the activation event in the same time step and moves to S9 while producing actuation command *c*. The actuator `act` responds to the event by generating the status change event *sc* and transitioning to S2. The event *sc* forces the mode calculator to update the system mode to $m1$ and create a mode change event as illustrated in Table 5.4

5.4.4 Scalability Analysis

For a TCD model, $\mathscr{G}$ with N TFPG nodes, M fault propagation edges and P TTA automata with a maximum of K unique timing constraints per automata then the resulting transformed TCD model, $\mathscr{G}^{Xr}$ will consist of (M + P) TAs with 2N communication channels, (M + PK + $|\Sigma|$) bounded integer variables and 1 clock variable. The M UPPAAL TAs are fault edge automata that correspond to M TFPG fault edges, P TAs are the translated UPPAAL TA from the P TTA models, 2N communication channels are needed to signal activation and de-activation of N TFPG nodes, with M+PK integer variables for counting ticks and $|\Sigma|$ variables are used for loose handshake signaling between translated TTAs. The number of locations are fixed in each fault edge automata i.e 11 but the translation of TTA to TA adds extra intermediate locations equal to the number of unique timing constraints associated with the outgoing transitions, i.e. for a given location with k unique timing constraints the translation will add k + 1 intermediate locations.

5.5 Summary

In this chapter, we described the new graph based language, $\mathscr{L}_{TCD}$ to model the fault propagation in CPS. It uses TFPG to represent faults, their effects and propagation in physical components. The formalism defines the response of the protection devices (cyber components) to the observed fault effects using TTA. The TTA based models are also capable of capturing the behavior of protection devices in the presence of cyber (detection and stuck) faults. Lastly, the maps Ω and Ψ threads the two sub-systems together by modeling the interplay between fault effects in physical system and actions of protection system. The chapter also describes and verifies the fault propagation and execution semantics of a TCD model with the help of UPPAAL TA. In the end, simulation traces of two scenarios involving physical and cyber faults in a concrete example are shown.

5.6 Contributions

The TCD language is first introduced in the publication titled, *Using temporal causal models to isolate failures in power system protection devices* [141] and then formally re-defined along with execution semantics in the publication titled, *Qualitative Fault Modeling in Safety Critical Cyber Physical Systems* [142]. All UPPAAL TA models and Python utility to translate TTA models are available in Github repository [140].

Chapter 6

TCD Fault Model Synthesis

6.1 Problem Statement

Develop a transformation, $\mathscr{A}_1 : \mathscr{M}^{topology} \rightarrow \mathscr{M}^{fault}$ that maps a given model of the power system network specified using standard data formats to a TCD based fault model.

6.2 Challenges

Synthesizing fault models for power networks is a challenging task mainly due to the size of the power system and the associated heterogeneity. We identified the following three major challenges that a transformation pipeline has to overcome:

- Generating system fault models using graph theoretic approach that does not rely on power system simulators.
- The next challenge is to identify and capture the modal dependencies, i.e., depending on what mode the system is in, the fault effect propagation must change.
- Large size of power network can lead to huge monolithic fault models and managing such fault models can be very challenging.

6.3 Solution Approach

The apparatus protection in power system is performed with high degree of redundancy, where each equipment has dedicated primary and shared secondary protection relays. Modern numerical relays such as SEL-421 [16] consists of large number of internal flags that indicate events related to the detection of fault conditions, actuation commands and change in the status of a breaker. Thus, an unique signature can be constructed for physical faults in power equipment based on the state of these flags. Moreover, the modern relays have the capability to track changes in their internal flags as well as transmitting concise event reports in the form of summaries over serial port.

We propose a component based approach, where each component TCD fault model captures the effect of physical fault and abnormal conditions pertaining to a power system equipment as a function of observable state of protection assemblies while considering cyber faults i.e. missed and spurious detection in relays and stuck fault in breakers. Figure 6.1 highlights a generic system level fault model composed of component TCD fault models. We identify two different classes of interactions between component fault models, defined as:

- **Intra-component Interactions**: These interactions occur between TFPG and TTA sub-models of the same component model, $\mathscr{G}^{C1}$, where C1 is the component label. These interaction can signify

  1. Change in the location of a protection behavior automaton, $P \in \mathscr{G}^{C1}$, from `S1` to `S2` as a result of activation or de-activation event , a related to a node, n in TFPG sub-model such that $\Psi_{act}(n) = e$ or $\Psi_{ina}(n) = a$.
  2. Activation and de-activation of fault propagation edge, e as a result of system mode change caused by the location change in a protection device automaton, $P \in \mathscr{G}^{C1}$ of the same component, such that the new system mode, m belongs to the set $EM(e)$.
  3. Location transition in one protection system automaton, $P1 \in \mathscr{G}^{C1}$ causing change in location of other automata, $P2 \in \mathscr{G}^{C1}$ in the same component model.
  4. Fault effect propagation between TFPG nodes of the same component model.

- **Inter-component Interactions**: These interactions transcends a TCD component boundary and can affect TFPG and TTA sub-models of a different TCD component. These interactions can denote:

  1. Fault propagation across component boundaries, i.e. activation of TFPG node, $n1$ in component C1, causes the activation of node $n2$ in component C2 as shown in Figure 6.1.
  2. Location transition in protection system automaton $P1 \in \mathscr{G}^{C1}$ causing change of location in an automata of a different component, $P1 \in \mathscr{G}^{C2}$.
  3. Activation and de-activation of fault propagation edges as a result of state change in protection devices automaton of a different component.

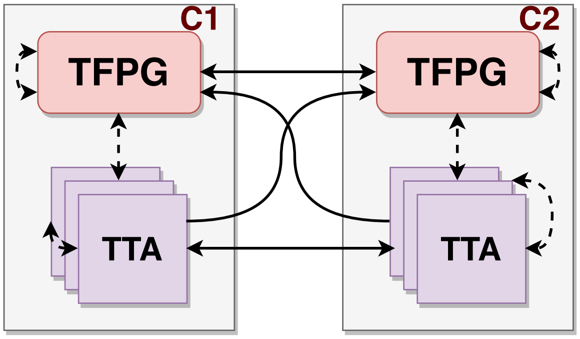


Figure 6.1: Component based TCD fault model
Bold and dash arrows represents inter and intra component interactions respectively

In this study, we are limiting our scope to physical faults in transmission lines and considering only branch overloads as abnormal conditions. However, the approach can be generalized easily to include other power system equipment such as transformers, buses and generators. A TCD component fault model for a transmission line consists of TFPG sub-models that captures the effect of grounding fault and time triggered automata of protection assemblies (primary) connected to its ends. We describe these sub-models in detail in following sub-sections.

6.3.1 TTA model of Protection Assemblies

A protection assembly is a logical containment of the instrument transformers, numerical relays and breakers that collectively detect and mitigate fault effects. A numerical relay is a collection of number of protection functions that safeguards a physical component against a variety of fault effects. A SEL-421 [16] relay has eight protection functions ranging from out-of-step synchronism to overcurrent protection. These protection functions are realized by one or more protection elements. For instance, SEL-421 relay provides 5 independent mho phase distance elements for distance protection against phase-to-phase, phase-to-phase-to-ground, and three-phase faults. Each protection element has a triggering condition, defined over physical variables (current and voltage samples) and protection action to isolate the fault effects such as breaker commands. Some protection elements are instantaneous in nature as the protection action is initiated as soon as the trigger condition is satisfied, for example, Zone 1 mho element in distance protection or instantaneous overcurrent elements. While other protection elements such as time definite time overcurrent relays asserts a protection action only if the trigger conditions holds for pre-defined period of time. The proposed

TCD component model consists of TTA templates of *Distance* and *Overcurrent* relays which are primarily used to safeguard system against transmission line faults and abnormal conditions pertaining to branch overloads.[1] Their TTA models are described in following sub-sections.

6.3.1.1 Distance Protection

Distance relays detect phase to ground and phase to phase faults by monitoring the apparent impedance. Any grounding fault in a transmission line leads to a sudden increase in current flowing through the conductor and decrease in bus voltages. The reduction in impedance (V/I) below a predefined value acts as a trigger for the distance relay to instruct the breakers to open and isolate the fault. Typically, distance relays are provided with multiple zones of protection to meet the stringent selectivity and sensitivity requirements, where zone is referred to as a segment of primary and back-up transmission lines. Modern relays such as SEL-421 consists of at least three independent distance elements that together provide distance protection. These elements, referred as Zone 1, Zone 2 and Zone 3 elements, are configured with specific impedance thresholds to detect faults in zones.

Some distance protection implements communication assisted trip schemes, where distance elements in neighboring numerical relays exchange trip signals to reduce the fault clearing time. A trip signal is sent by the relay (closest to the fault) to the adjacent relay connected at the other end of the transmission line. There are primarily four types of transfer trip protocols, 1) Direct Transfer Trip, 2) Direct Under-reaching Transfer Trip, 3) Permissive Under-reaching Transfer Trip (PUTT) and 4) Permissive Over-reaching Transfer Trip. In direct type protocols, only one of the relays have to detect fault conditions before sending the trip signal whereas in permissive type protocols, relays at both ends of the line have to detect the fault. In under reaching schemes, zone 1 element sends the trip signal whereas in overreaching schemes zone 2 element transmits the trip signal. In this study, we are considering the PUTT scheme only.

A zone 1 element is meant for the protection of the primary line only. Typically, it is set to cover 80% of the primary line length and provides fastest response due to the absence of any intentional time delay associated with the protection action. Operating time of the Zone 1 element is of the order of 1 cycle (16 ms) [16]. Figure 6.2 shows the TTA model of a zone 1 element. The

[1] We use the words relay and protection interchangeably in context of distance and overload relays as they are implemented as protection functions in a numerical relay.

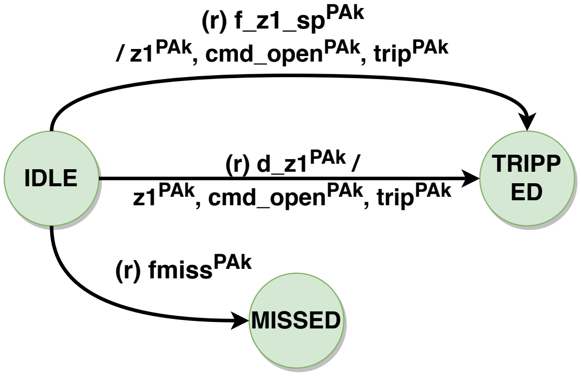

(r) f_z1_sp^PAk
/ z1^PAk, cmd_open^PAk, trip^PAk
IDLE
(r) d_z1^PAk /
z1^PAk, cmd_open^PAk, trip^PAk
TRIPP
ED
(r) fmiss^PAk
MISSED

Figure 6.2: TTA model of Zone 1 Element in k^{th} Protection Assembly

automaton has three locations with IDLE being the initial location. In IDLE location, the automaton periodically checks for the presence of three events, $f\_z1\_sp^{PAk}$, $fmiss^{PAk}$ and d_z1^{PAk}, where *PAk* is the protection assembly identifier. The event, $f\_z1\_sp^{PAk}$ denotes the presence of spurious detection fault in the element while $fmiss^{PAk}$ signifies the missed detection fault in the relay.[2] The event, d_z1^{PAk} is generated when the measured impedance falls below the zone 1 threshold as a result of a physical fault. If the measured impedance is less than threshold or the spurious detection fault is present, then automaton jumps to TRIPPED location while generating three synchronization events, $z1^{PAk}$, cmd_open^{PAk} and $trip^{PAk}$ signifying change in relay bits **M1P**, **TRIP**, **KEY** respectively. These events are considered observable as numerical relay can be configured to track these bits and transmit event summaries if change is detected. On the other hand, if missed detection fault is present, signaled by the event f_miss^{PAk}, the automaton transitions to MISSED location and does not respond to any physical fault conditions. Figure 6.3 shows the translated UPPAAL TA based on the algorithm 1. The translated TA is a template with input parameters as *r*, *disc_id*,*f_sp_id*, *fmiss_id*, *z1_id*, *cmd_id* and *trip_id* where *r* is element frequency, and rest are event identifiers related to d_z1^{PAk}, f_z1_sp, $fmiss^{PAk}$, $z1^{PAk}$, cmd_open^{PAk} and $trip^{PAk}$ respectively.

Zone 2 element covers 100% of the primary line and also serves as backup protection for some part of the adjacent line. Typically, zone 2 element is set to reach 50% of the shortest adjacent line provided that $Z_p + 0.5Z_b \geq 1.2Z_p$, where Z_p and Z_b are the impedance of the primary and shortest adjacent line. In case, if the adjacent line is smaller in comparison to the primary line, such that $Z_p + 0.5Z_b < 1.2Z_p$, then the zone 2 reach is set to $1.2Z_p$. The Zone 2 element is a time delayed

[2] Individual missed detection faults for every element are not considered.

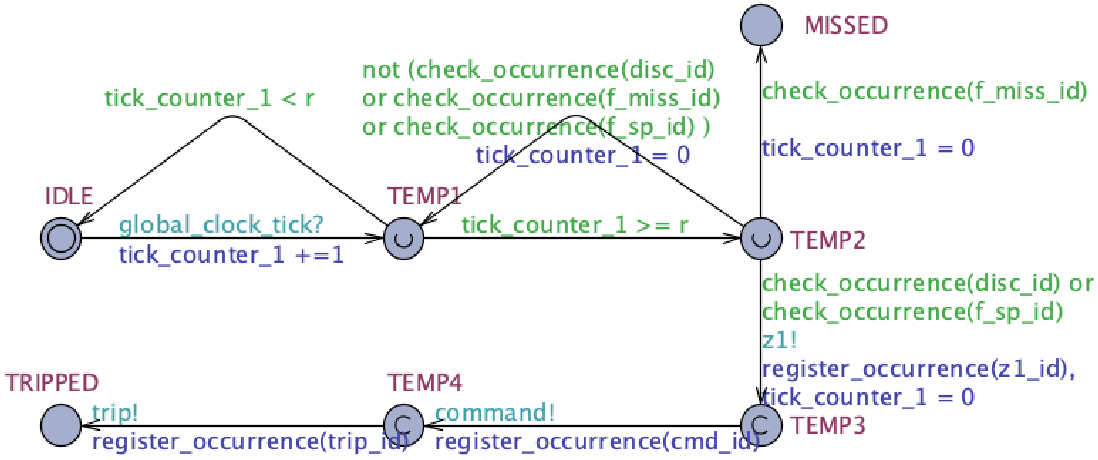

MISSED
tick_counter_1 < r
not (check_occurrence(disc_id)
or check_occurrence(f_miss_id)
or check_occurrence(f_sp_id))
tick_counter_1 = 0
check_occurrence(f_miss_id)
tick_counter_1 = 0
IDLE
TEMP1
global_clock_tick?
tick_counter_1 +=1
tick_counter_1 >= r
TEMP2
check_occurrence(disc_id) or
check_occurrence(f_sp_id)
z1!
register_occurrence(z1_id),
tick_counter_1 = 0
TRIPPED
TEMP4
trip!
register_occurrence(trip_id)
command!
register_occurrence(cmd_id)
TEMP3

Figure 6.3: UPPAAL TA template of Zone 1 Element

element i.e. it asserts a protection command only if the trigger condition is maintained for a predefined amount of time. This wait time allows a zone 1 element associated with adjacent numerical relay to engage first and clear the fault. Typically, a zone 2 element wait time is of the order of 15-30 cycles or 0.250-0.5 secs.

Figure 6.4 shows the TTA model of a zone 2 element in a protection assembly, *PAk*. The automaton consists of five locations with IDLE being the initial location. Similar to zone 1 automaton, this automaton also checks for the activation events related to missed detection fault, $fmiss^{PAk}$, zone 2 spurious detection fault, $f\_z2\_sp^{PAk}$ and reduction of impedance, d_z1^{PAk} and d_z2^{PAk}. The event d_z1^{PAk} is generated if the measured impedance is less than zone 1 threshold, and d_z2^{PAk} event is generated when the measured impedance is greater than zone 1 threshold but less than zone 2 threshold. After detecting the reduction in impedance, the automaton jumps to WAIT location. The automaton stays in the WAIT location for a maximum of $z2wait^{PAk}$ secs. The automaton jumps to TRIPPED location if the wait time expires or it detects a $trip^{PAk}$ event or the zone 1 element has issued a command, cmd_open^{PAk} to the breaker. In case the transition to TRIPPED occurs as a result of timer expiration, then the automaton generates an event symbolizing the command for the breaker to open. The automaton can also transition back to IDLE if the fault condition no longer exists by detecting the inactivation events d'_z1^{PAk} or d'_z2^{PAk}. In case if the automaton detects a spurious fault in IDLE location then it jumps to FAULT WAIT location and stays there for $z2wait^{PAk}$ secs before transitioning to TRIPPED while generating cmd_open^{PAk} event. Similarly,

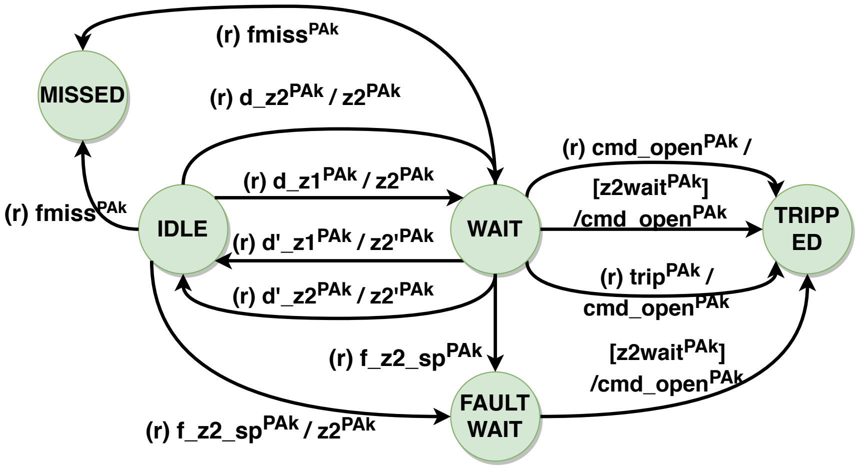


Figure 6.4: TTA model of Zone 2 Element in k^{th} Protection Assembly

the automaton transitions to `MISSED` from `IDLE` if missed detection fault is detected. The corresponding UPPAAL TA template of a zone 2 element is shown in Figure 6.5. The input parameters of the template include the relay frequency r, zone 2 wait time $delay$ and event identifiers f_miss_id, f_sp_id, $disc\_act\_id\_1$, $disc\_act\_id\_2$, $disc\_ina\_id\_1$, $disc\_ina\_id\_2$, cmd_id, $trip\_id$ that are related to the events $fmiss^{PAk}$, $f\_z2\_sp^{PAk}$, d_z1^{PAk}, d_z2^{PAk}, d'_z1^{PAk}, d'_z2^{PAk}, cmd_open^{PAk} and $trip^{PAk}$ respectively.

Zone 3 element is a remote back up that covers the primary line and the longest adjacent line. The zone 3 wait time is in the range 60-125 cycles i.e. 1-2 secs. The TTA model of a zone 3 element is similar to zone 2 element except the PUTT trip signal cannot be used to truncate the zone 3 wait time as shown in Figure 6.6 and the corresponding translated UPPAAL TA is highlighted in Figure 6.7.

6.3.1.2 Overload Protection

Overload protection in transmission lines is achieved through overcurrent elements based on the thermal model of line currents. If a line is allowed to be overloaded for prolonged amount of time, then the temperature of the conductor will rise, resulting in degradation of the insulation around the conductor which leads to two conductors coming in contact with each or external vegetation causing short circuit. The protection action in overcurrent elements is triggered when the current magnitude overshoots a pre-defined threshold, referred to as pick up current. In this work we are

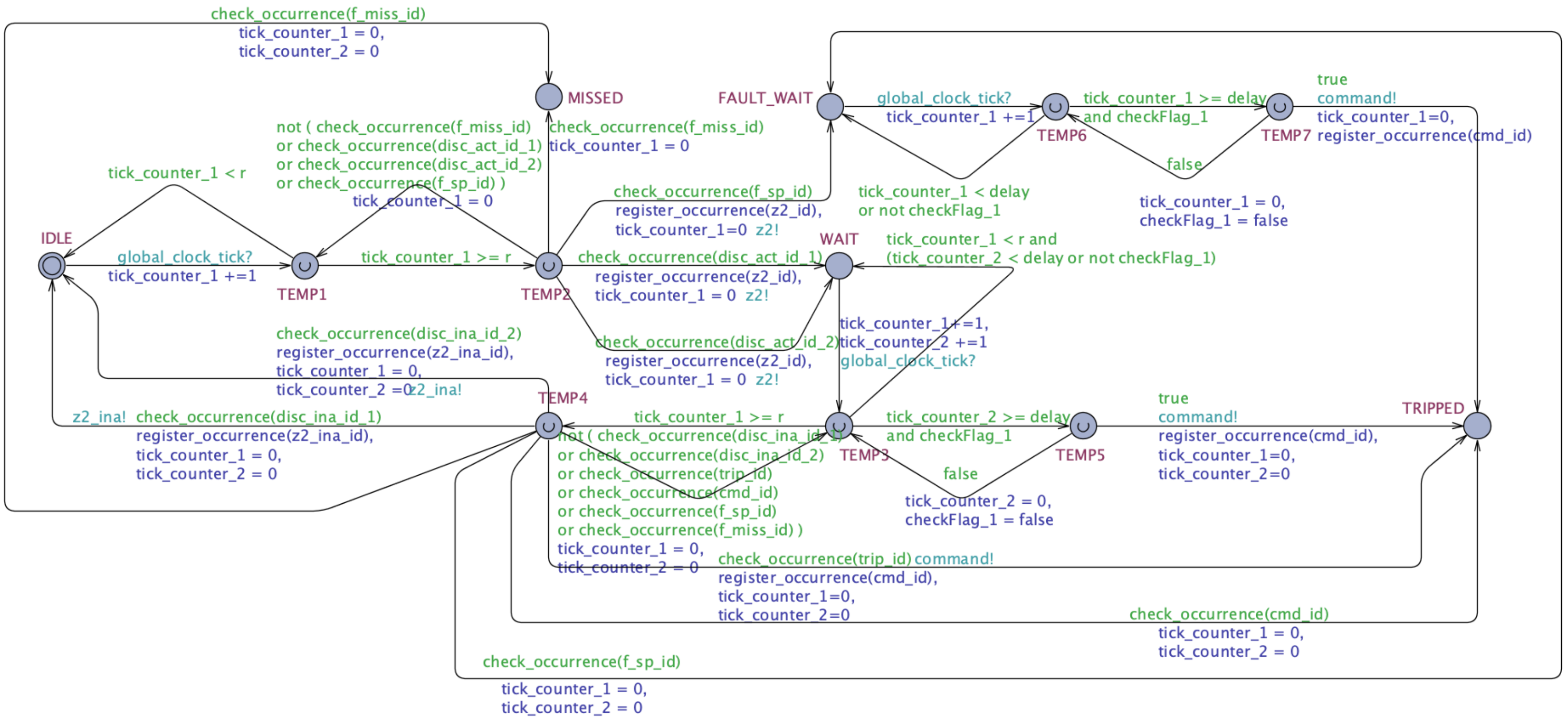


Figure 6.5: UPPAAL TA template of zone 2 Element

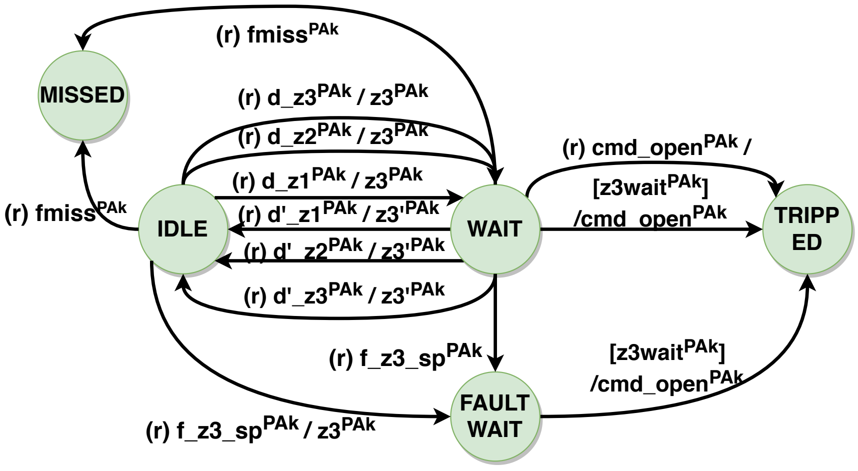

(r) fmiss^{PAk}
MISSED
(r) d_z3^{PAk} / z3^{PAk}
(r) d_z2^{PAk} / z3^{PAk}
(r) cmd_open^{PAk} /
(r) d_z1^{PAk} / z3^{PAk}
[z3wait^{PAk}]
(r) fmiss^{PAk}
(r) d'_z1^{PAk} / z3'^{PAk}
/cmd_open^{PAk}
IDLE
WAIT
TRIPP ED
(r) d'_z2^{PAk} / z3'^{PAk}
(r) d'_z3^{PAk} / z3'^{PAk}
(r) f_z3_sp^{PAk}
[z3wait^{PAk}]
/cmd_open^{PAk}
FAULT WAIT
(r) f_z3_sp^{PAk} / z3^{PAk}

Figure 6.6: TTA model of Zone 3 Element in k^{th} Protection Assembly

using a *Time-definite* overcurrent element that triggers a breaker to open only if the line current exceeds the pick up current for a specified amount of time. Figure 6.8 shows a TTA model of time definite overcurrent element (referred to as overload element in rest of the document) with five location with IDLE being the initial location. While in IDLE, the automaton checks for three events, $fmiss^{PAk}$, $f\_o\_sp^{PAk}$ and d_o^{PAk} periodically which denote the activation of missed detection fault, spurious detection fault and increase in line current above the pick up threshold. If $fmiss^{PAk}$ is observed then the automaton jumps to MISSED whereas detection of $f\_o\_sp^{PAk}$ forces the automaton to transition to FAUL-WAIT location while emanating an observable event, o^{PAk}, signifying the change in status of internal relay flag. The automaton stays in FAULT WAIT for $owait^{PAk}$ clock ticks before moving to TRIPPED location and producing breaker open command, denoted by the event cmd_open^{PAk}. In case, the event d_o^{PAk} is observed in IDLE location then the automaton jumps to WAIT and produces o^{PAk}. While in WAIT, the automaton checks for absence of overloading conditions, identified by d'_o^{PAk} or cmd_open^{PAk} generated by other elements in the assembly. If none of the events are observed in $owait^{PAk}$ secs then it jumps to TRIPPED while emanating cmd_open^{PAk} as shown in Figure 6.8. The corresponding UPPAAL TA template of the overload element is shown in Figure 6.9. The input parameters of the template include the relay frequency r, wait time $delay$ and event identifiers f_miss_id, f_sp_id, $disc\_act\_id\_1$, $disc\_ina\_id\_1$, cmd_id that are related to the events $fmiss^{PAk}$, $f\_o\_sp^{PAk}$, d_o^{PAk}, d'_o^{PAk}, cmd_open^{PAk} respectively. The parameter $owait$ in TTA or $delay$ in UPPAAL TA depends upon the thermal rating of the conductor

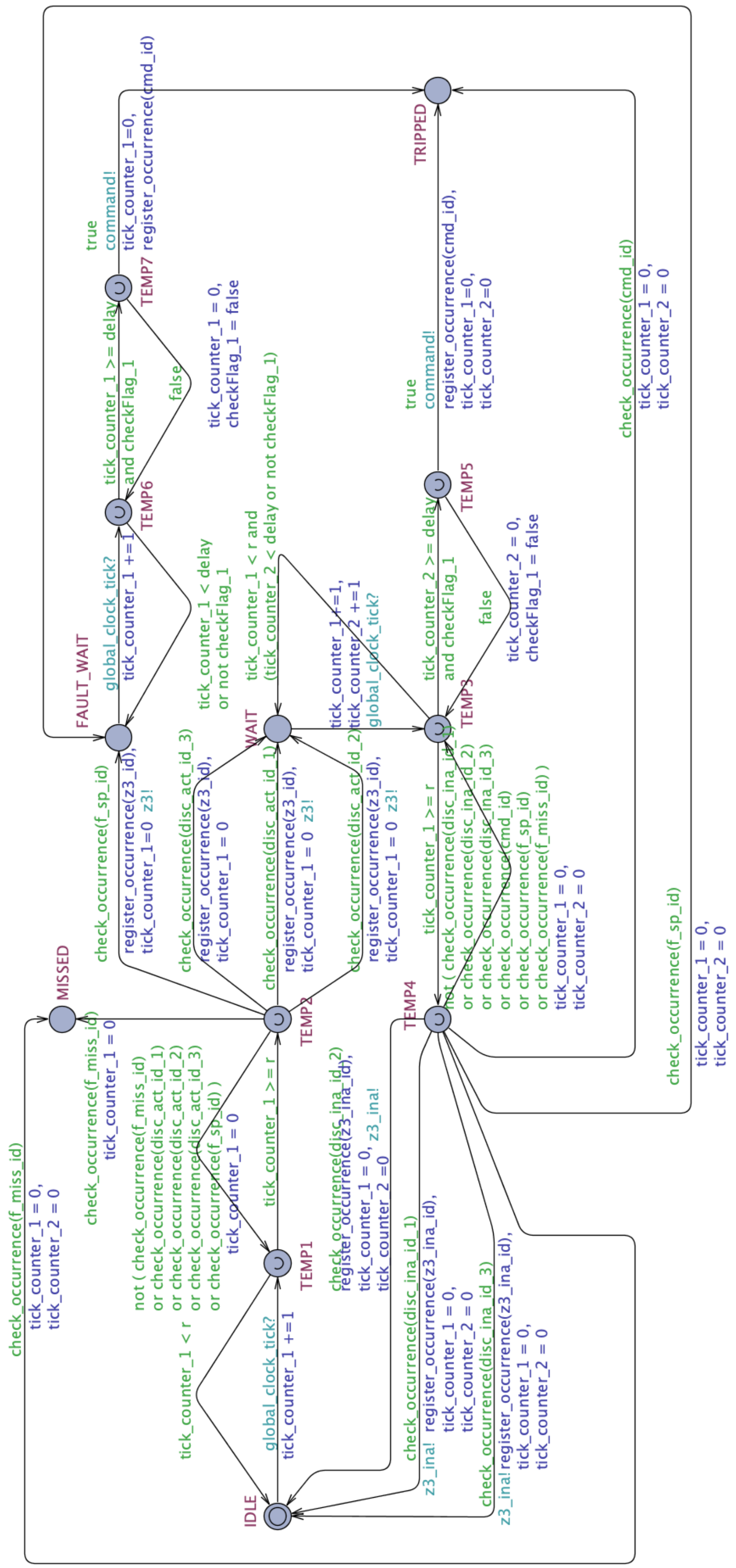

IDLE
TEMP1
TEMP2
MISSED
WAIT
FAULT_WAIT
TEMP3
TEMP4
TEMP5
TEMP6
TEMP7
TRIPPED
global_clock_tick?
tick_counter_1 +=1
tick_counter_1 < r
tick_counter_1 >= r
not (check_occurrence(f_miss_id)
or check_occurrence(disc_act_id_1)
or check_occurrence(disc_act_id_2)
or check_occurrence(disc_act_id_3)
or check_occurrence(f_sp_id))
tick_counter_1 = 0
check_occurrence(f_miss_id)
tick_counter_1 = 0
check_occurrence(f_miss_id)
tick_counter_1 = 0,
tick_counter_2 = 0
check_occurrence(f_sp_id)
register_occurrence(z3_id),
tick_counter_1=0 z3!
check_occurrence(disc_act_id_3)
register_occurrence(z3_id),
tick_counter_1 = 0
check_occurrence(disc_act_id_1)
register_occurrence(z3_id),
tick_counter_1 = 0 z3!
check_occurrence(disc_act_id_2)
register_occurrence(z3_id),
tick_counter_1 = 0 z3!
global_clock_tick?
tick_counter_1 +=1
tick_counter_1 >= delay
and checkFlag_1
false
true
command!
tick_counter_1=0,
register_occurrence(cmd_id)
tick_counter_1 = 0,
checkFlag_1 = false
tick_counter_1 < delay
or not checkFlag_1
tick_counter_1 < r and
(tick_counter_2 < delay or not checkFlag_1)
tick_counter_1 +=1,
tick_counter_2 +=1
global_clock_tick?
tick_counter_2 >= delay
and checkFlag_1
false
tick_counter_2 = 0,
checkFlag_1 = false
true
command!
register_occurrence(cmd_id),
tick_counter_1=0,
tick_counter_2=0
check_occurrence(cmd_id)
tick_counter_1 = 0,
tick_counter_2 = 0
tick_counter_1 >= r
not (check_occurrence(disc_ina_id_1)
or check_occurrence(disc_ina_id_2)
or check_occurrence(disc_ina_id_3)
or check_occurrence(cmd_id)
or check_occurrence(f_sp_id)
or check_occurrence(f_miss_id))
tick_counter_1 = 0,
tick_counter_2 = 0
check_occurrence(f_sp_id)
tick_counter_1 = 0,
tick_counter_2 = 0
check_occurrence(disc_ina_id_2)
register_occurrence(z3_ina_id),
tick_counter_1 = 0, z3_ina!
tick_counter_2 =0
check_occurrence(disc_ina_id_1)
z3_ina! register_occurrence(z3_ina_id),
tick_counter_1 = 0,
tick_counter_2 = 0
check_occurrence(disc_ina_id_3)
z3_ina! register_occurrence(z3_ina_id),
tick_counter_1 = 0,
tick_counter_2 = 0

Figure 6.7: UPPAAL TA template of zone 3 Element

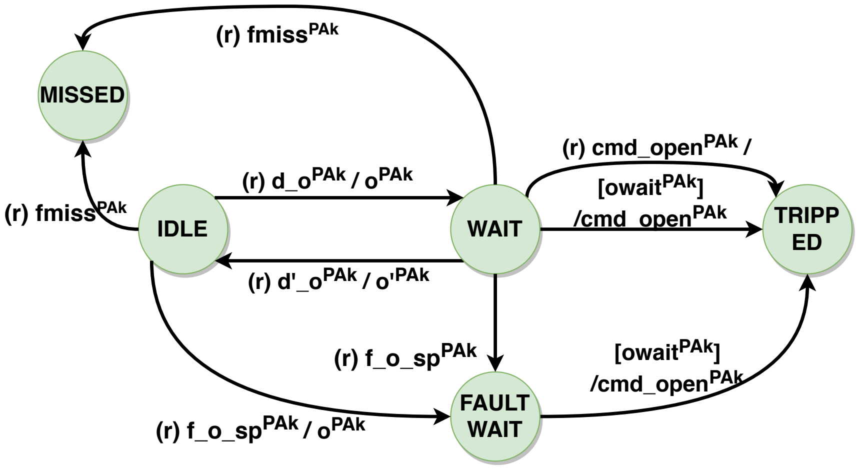


Figure 6.8: TTA model of overload element in k^{th} protection assembly

and in this work we have assumed this parameter to have a value of 10 minutes.

6.3.1.3 Breaker

Breakers are mechanical switches that disconnect the flow of current through a conductor. Figure 6.10 shows a time triggered model of the breaker. The automaton has six locations with `CLOSE` being the initial location. The automaton has three parameters, tto, ttc and r, where the first two denote the time to open and close the breaker while the third signifies the duration after which the breaker controller checks the status of instructions from the relays. While in `CLOSE` location, the automaton periodically checks for two events cmd_open^{PAk} and f_sc^{PAk}. The event, cmd_open^{PAk} represents the open command from the relay PAk and f_sc^{PAk} denotes the activation event related to stuck fault that forces the breaker to ignore further instructions. If cmd_open^{PAk} is observed, then the automaton transitions to `OPENING` location. The automaton stays in the location for a maximum of ttc secs and then transitions to `CLOSE` location while generating a synchronization event (observable) $act\_sc\_open^{PAk}$ implying the change of breaker status. However, while waiting in location, `OPENING`, if stuck fault becomes active, then the breaker transitions to `STUCK-CLOSE` location. A similar path is followed by the breaker automaton after receiving cmd_close^{PAk} event from the relay while in `OPEN` location. The breaker transitions to `CLOSING` and after spending `ttc` secs, it jumps to `CLOSE` location along with the generation observable $act\_sc\_close^{PAk}$ event. The corresponding UPPAAL TA template is shown in Figure 6.11. The input parameters of the

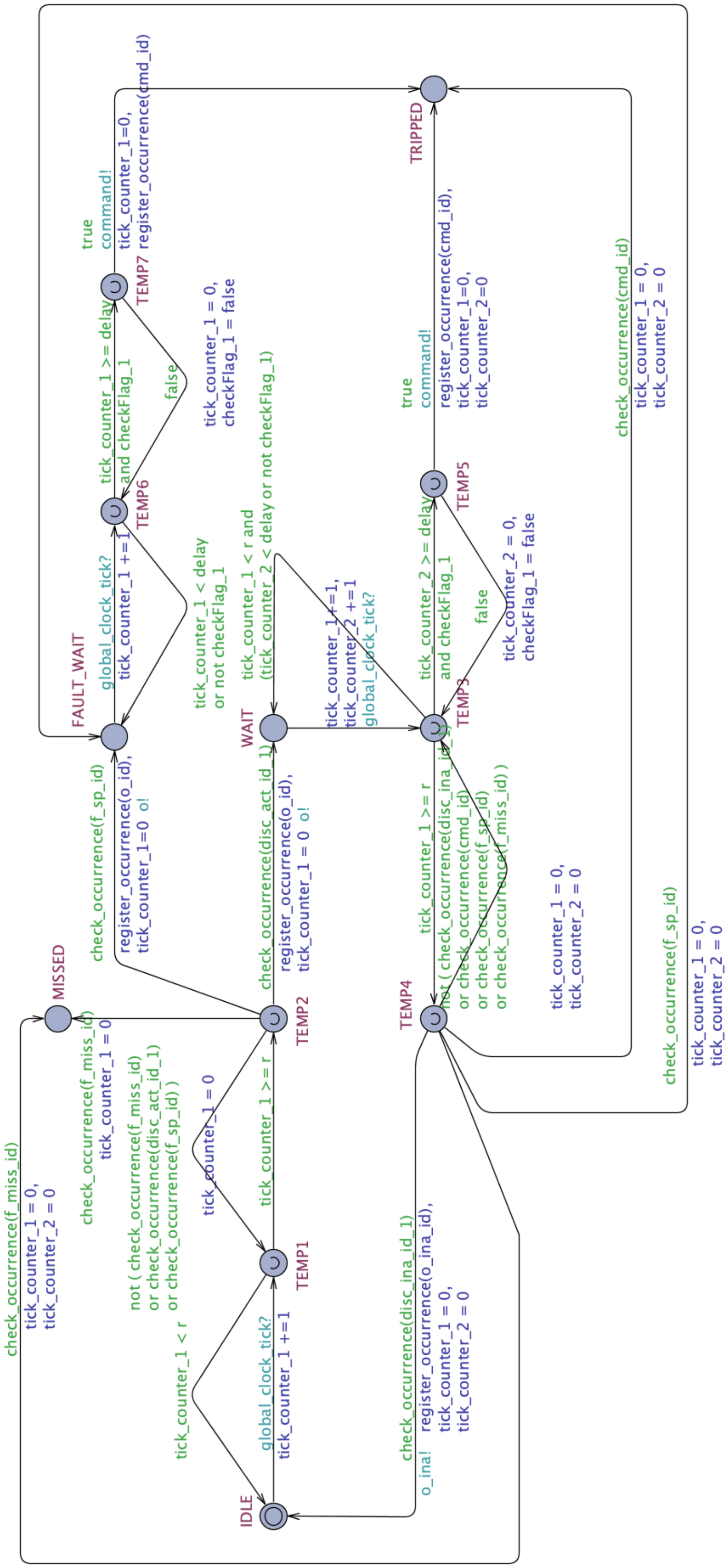


Figure 6.9: UPPAAL TA template of overload element

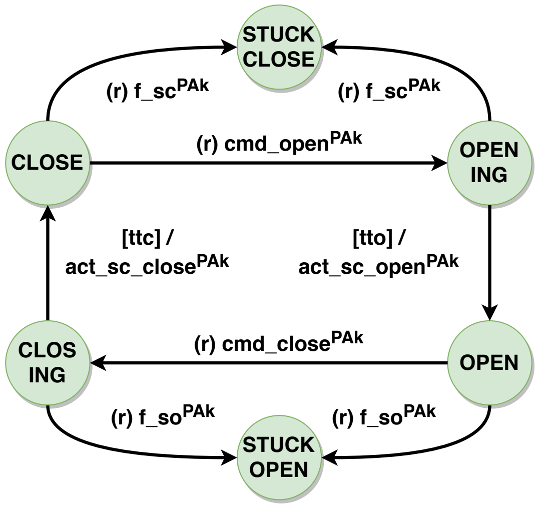


Figure 6.10: TTA Model of a breaker in k^{th} Protection Assembly

template include breaker controller frequency r, time to open tto, time to close ttc and identifiers cmd_id, cmd_id_2, f_stuck_id, act_id and act_id_2 that relates to event cmd_open^{PAk}, cmd_close^{PAk}, f_sc^{PAk}, $act\_sc\_open^{PAk}$, $act\_sc\_close^{PAk}$.

6.3.2 TFPG Model

TFPG sub-model in the TCD model of a transmission line includes two sets of nodes and edges. The first set represents the effect of grounding and wiring faults and the second set captures the impact of protection actions on branch overloads leading to abnormal operating conditions. The TFPG sub-model is described in more detail in the following sections.

6.3.2.1 Modeling physical faults and their effects

When a transmission line is subjected to a physical fault (phase to ground or phase to phase) a reduction of impedance is detected by the distance elements in the nerby protection assemblies. The reduction in impedance results in changing the state of the protection relays, indicated by observable event summaries. Thus, a causal model can be created between a physical fault of a transmission line and its observed effects i.e. reduction in impedance measured by different protection assemblies. Figure 6.12 shows the part of the TFPG sub-model associated with the TCD component model of a transmission line, TL_p. It consists of three fault nodes ($F_1^{TL\_p}$, $F_2^{TL\_p}$, $F_3^{TL\_p}$) and six

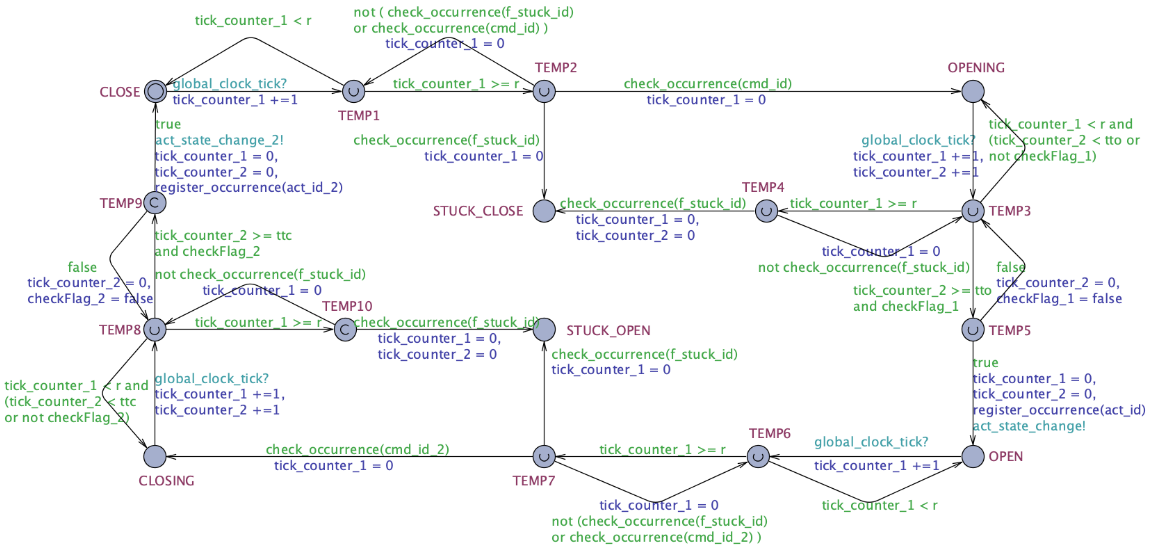


Figure 6.11: UPPAAL TTA template of a breaker

discrepancy nodes ($D1^{PAk}$, $D2^{PAk}$, $D3^{PAk}$) , ($D1^{PAj}$, $D2^{PAj}$, $D3^{PAj}$), where PAk and PAj are the protection assemblies connected to each end of the line. A fault node represents a physical fault along the segment of the line such that the response of all zone elements in PAk and PAj remains same. The lengths of these segments depends on the zone 1 reach of the protection assemblies PAk and PAj. For 80% reach of zone 1 elements in PAk and PAj, the nodes F_1 and F_3 cover 20% of line from both end whereas F_2 denotes the rest of the line, i.e., 20 - 80% as indicated in Figure 6.15. The two sets of discrepancies ($D1^{PAk}$, $D2^{PAk}$, $D3^{PAk}$), ($D1^{PAj}$, $D2^{PAj}$, $D3^{PAj}$) denote the reduction in impedance measured by the protection assemblies PAk, PAj respectively. The discrepancy node $D1^{PAk}$ signifies the measured impedance by PAk to be less than $z1thresh^{PAk}$ while $D2^{PAk}$ and $D3^{PAk}$ signify the measured impedance to be in the ranges ($z1thresh^{PAk}$, $z2thresh^{PAk}$) and ($z2thresh^{PAk}$, $z3thresh^{PAk}$) respectively, where $z1thresh^{PAk}$, $z2thresh^{PAk}$, $z3thresh^{PAk}$ are zone 1, 2, 3 thresholds of distance elements in PAk. The activation and de-activation of discrepancy nodes produces events, based on eq. (6.1) - (6.6), which act as stimulus for TTA models of zone elements described in previous section.

$$\Psi_{act}(D1^{PAk}) = d\_z1^{PAk} \tag{6.1}$$

$$\Psi_{ina}(D1^{PAk}) = d'\_z1^{PAk} \tag{6.2}$$

$$\Psi_{act}(D2^{PAk}) = d\_z2^{PAk} \tag{6.3}$$

$$\Psi_{ina}(D2^{PAk}) = d'\_z2^{PAk} \tag{6.4}$$

$$\Psi_{act}(D3^{PAk}) = d\_z3^{PAk} \tag{6.5}$$

$$\Psi_{ina}(D3^{PAk}) = d'\_z3^{PAk} \tag{6.6}$$

The TFPG node $F_1^{TL\_p}$ represents faults in 0-20% of the given line, therefore, any fault in this segment will result in measured impedance to fall in the range (0, $z1thesh^{PAj}$) for PAj and ($z1thresh^{PAk}$, $z2thresh^{PAk}$) for PAk. Thus, there should exist a fault propagation edge between node $F_1^{TL\_p}$ and discrepancy nodes $D1^{PAj}$, $D2^{PAk}$ as shown in Figure 6.12. Similarly, there are two fault edges between $F_3^{TL\_p}$ and discrepancy nodes, $D2^{PAj}$, $D1^{PAk}$ capturing the effect of any fault in segment, 80-100% of the line. Lastly, the fault edges starting from $F_2^{TL\_p}$ and ending at nodes $D1^{PAj}$, $D1^{PAk}$ models the fault effect propagation when a physical fault is injected 20-80% length of TL_p.

The fault effects transcend component boundaries and the zone 2 or zone 3 elements of a neighboring protection assembly can detect reduction in impedance. To capture inter-component fault effect propagation, we create two uncertain fault edges between the local fault node and discrepancies nodes, $D2^{PAi}$ and $D3^{PAi}$ associated with a remote protection assembly PAi connected to an arbitrary adjacent line TL_i. The fault effect can propagate only if the zone 3 element of PAi can detect fault conditions i.e., $z3thresh^{PAi} \geq z^{TL\_i} + z^{TL\_p}$, where z^{TL_i} and z^{TL_p} are the impedance of the lines. Similarly, there are incoming edges from the fault nodes associated with the TFPG sub models of neighboring transmission lines that reach the local discrepancy nodes, $D2^{PAk}$, $D2^{PAj}$, $D3^{PAk}$ and $D3^{PAj}$ as indicated in Figure 6.12. These inter component fault propagation edges are marked uncertain because the segment represented by the fault node is calculated based on the zone 1 threshold of primary protection assemblies. These line segments can be further divided based on secondary or tertiary protection assemblies. However, such refinement depends not only on the impedance thresholds of zone 2 and 3 elements but also on the actual power flow. We consider

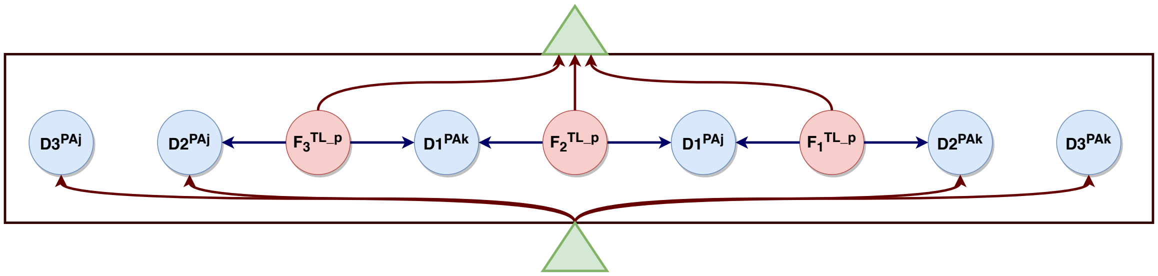


Figure 6.12: Partial TFPG in TCD component model of transmission line
Blue and red colored edges denote certain and uncertain edges respectively

the refinement of the fault nodes as the task to be handled as future work. It is important to note that inter component fault edges ($F_x^{TL\_y}$, $D2^{PAz}$) and ($F_x^{TL\_y}$, $D3^{PAz}$) are mutually exclusive i.e. fault effect from $F_x^{TL\_y}$ can only take one path and reach either $D2^{PAz}$ or $D3^{PAz}$.

The operating conditions that influences the fault propagation depends upon the flow of power from a generator (source) to the fault location (sink). Power can flow through a path if all the breakers along the path are in closed position. Quantifying different system modes is analogous to identifying different paths or breaker configurations for power to flow from source to sink. For small networks calculating these distinct paths between sources and fault location is feasible but for larger networks enumerating all system modes is infeasible as the number of path between two nodes can increase exponentially with the size of the network. The activation condition associated with an edge between nodes $F_x^{TL\_y}$ and Dz^{PAk} can also be viewed as presence of at-least one active[3] simple path starting from any of the sources and terminating at faulted equipment, TL_y such that the protection assembly PAk is part of it. Thus any path can be divided into two segments, the first segment is between the faulted equipment and the protection assembly, whereas the second segment starts at protection assembly and ends at the power source. We label the collection of protection assemblies in the first segment as a local mode, $m \in EM(F_x^{TL\_y}, Dz^{PAk})$ and the edge activation condition based on m is defined in eq. (6.7)

$$\Omega(m) = \bigwedge_{i \in m} \text{CLOSE}^i \wedge \Gamma(PAk, g, TL\_y) \tag{6.7}$$

where CLOSE^i is the location of breaker associated with the i^{th} protection assembly, g is an arbitrary generator (power source), $\Gamma(p,q,r)$ is boolean function that evaluates to true if there exists

[3] An active path imply all breakers along that path are in close location.

an active path between p and q that does not contain r. By defining the attribute EM of a fault edge based on local modes aids in avoiding huge overhead of enumerating all system modes as the number of local modes i.e distinct paths between protection assembly and fault location are very less as compared to complete distinct paths between power source and faulted location due to the proximity of the protection assembly to the fault location. The last attribute associated with an edge is the maximum and minimum time taken for the fault effect to propagate over the edge. This duration depends upon the time taken by protection relays to detect reduction in impedance. For modern protection relays, this time period ranges between [0, 0.032] secs [16].

6.3.2.2 Modeling Abnormal Conditions

Abnormal conditions pertain to operating conditions such as branch overloads that can gradually weaken the system resulting into uncontrolled tripping of power system equipment. In this study, we are interested in capturing the effect of protection actions (distance and overload elements) in creating a cascade of branch overloads while considering detection and stuck faults in relays and breakers. We extend the TFPG sub-model in the component TCD of a transmission line with two discrepancy nodes, O^{PAk}, O^{PAj} as shown in Figure 6.13. These discrepancies denote the detection of increase in current, i.e., overload conditions in the transmission line as measured by overload elements in protection assemblies PAk and PAj respectively.

A remote transmission line, TL_i, can experience the overload condition as result of opening of a breaker in any of the two protection assemblies PAk and PAj. The open command can be instructed by distance or overload elements in PAk and PAj. To model this causal relationship, an uncertain inter component fault propagation edge is created between every local discrepancy node (Dz^{PAk}, Dz^{PAj}, O^{PAk}, O^{PAj} $\forall z \in \{1,2,3\}$) and the overload discrepancy node of another component, O^{PAi}. The mode condition for such an edge requires that the breaker in the protection assembly associated with the source node is in OPEN location and the power should be flowing through the remote equipment. Enumerating all complete paths between power sources and loads which contain the remote equipment is challenging and we again specify these requirement based on local mode, m and Γ as described for an edge emanating from Dz^{PAk} in eq. (6.8)

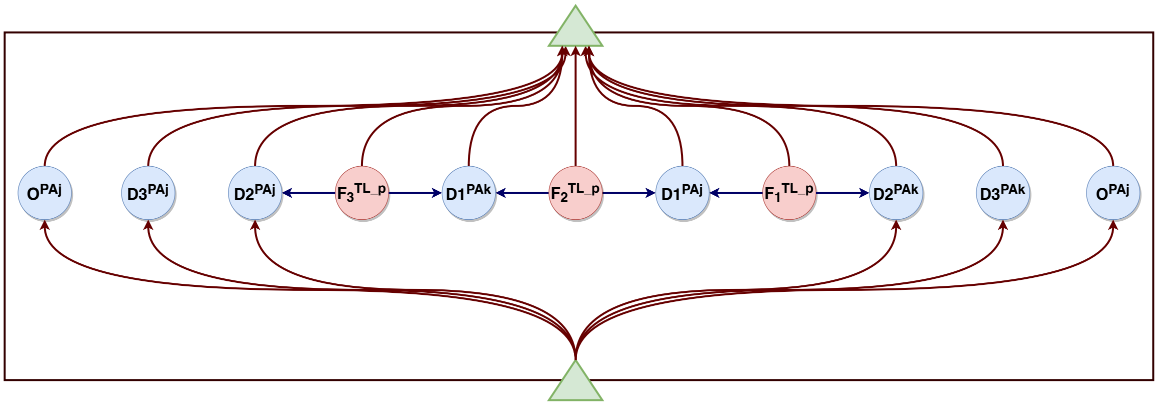


Figure 6.13: Complete TFPG in TCD component model of transmission line

$$\Omega(m) = \text{OPEN}^{PAk} \wedge \big((\Gamma(g, PAi, TL\_q) \wedge \Gamma(l, PAj, TL\_q)) \vee (\Gamma(g, PAi, TL\_q) \wedge \Gamma(l, PAj, TL\_q))\big) \tag{6.8}$$

where g is power source, l is a load and (PAi, PAj) are the protection assemblies connected to the two ends of remote transmission line, TL_q. Similarly, there are incoming edges that terminate at O^{PAk} or O^{PAj} implying the action of remote protection assemblies can overload TL_p as indicated in Figure 6.13. Its important to point out that the actual overloading of the equipment depends upon the quantitative power flow and topology analysis, which a qualitative TFPG model cannot completely capture. Thus to overcome this limitation, we over-approximate the causal relationship between opening of a breaker in one component and overloading of other by connecting the discrepancy nodes of a TCD component model of a line to every overload discrepancy node in the TCD component model of the remaining lines in the rest of the network along with marking these edges as uncertain. Lastly, the time attribute, ET of these edges depends upon the time it takes for the overcurrent element to detect these rise in current and is of the order of 0-2 cycles, i.e., 0-0.032 secs [17].

6.3.3 System Fault Model Synthesis

The system TCD fault model is synthesized by aggregating the individual TCD component models as described in the eq. (6.9), where $\mathscr{M}^{sys}$ is the system fault model and $\mathscr{M}^{c_i}$ is the i^{th} com-

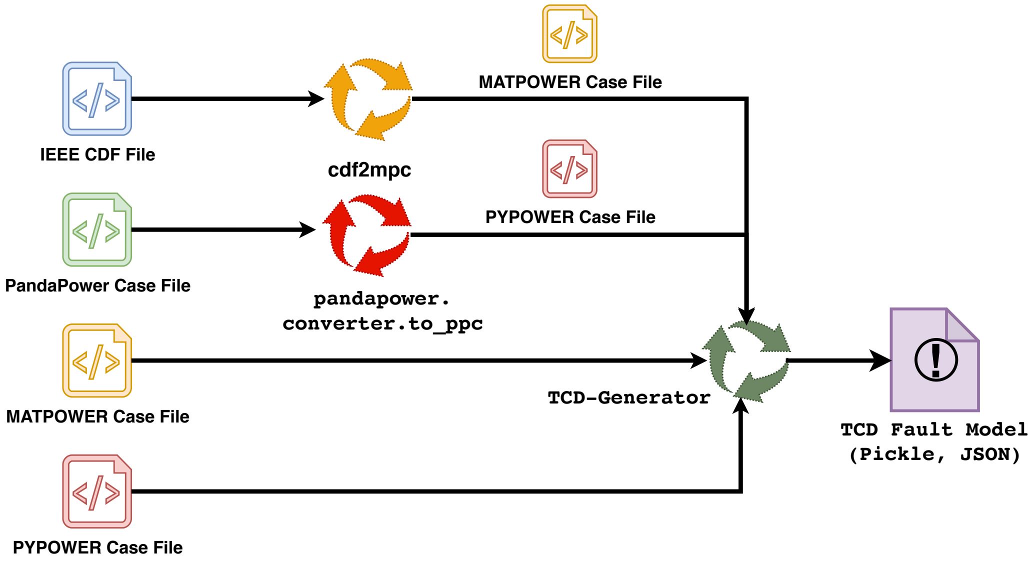


Figure 6.14: Fault Model Synthesis Block Diagram

ponent model. The semantics of the complete model can be derived by the parallel asynchronous composition of the individual TCD models.

$$\mathscr{M}^{sys} = \mathscr{M}^{c_1} \cup \mathscr{M}^{c_2} \cup \ldots \cup \mathscr{M}^{c_n} \tag{6.9}$$

Creating TCD fault model for cyber physical energy systems manually is tedious and error prone task due to the large number of equipment involved. We have created a model transformation, $\mathscr{A}$, that generates TCD fault model from a power network as highlighted in Figure 6.14. The input types supported in the current version includes, case formats of widely used load flow solvers[4] such as MatPower [143], PyPower [144] and PandaPower [145] along with IEEE Common Data Format (CDF) [146]. According to Figure 6.14, the block labeled as *TCD-Generator* performs the transformation by translating PyPower or MatPower case files, whereas, the remaining supported input formats are first translated to MatPower or Pypower files using standard utility functions such as *cdf2mpc* and *pandapower.coverter.to_ppc*. The generated TCD fault model can be serialized in two formats *Pickle* or *JSON* and the corresponding translation algorithm is described in listing 5.

The algorithm accepts a network graph, N outlining different types of equipment along with their physical connections and produces a TCD fault model, $\mathscr{M}$. The translation procedure starts

[4] In power engineering, a load flow solver performs numerical analysis of the flow of power in the interconnected system.

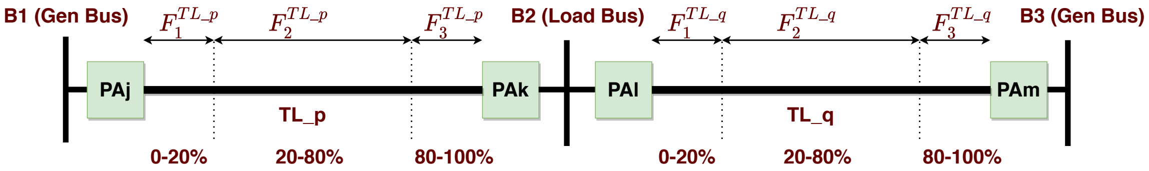


Figure 6.15: Two transmission line network

by creating two maps, to_bus and $from\_bus$ to store the buses connected to every transmission line [lines 2-4]. After that two hash map based adjacency lists are created, where key is transmission line label and values are the neighboring lines. Using the adjacency lists, then algorithm determines primary and secondary protection assemblies for every transmission line. This information is stored in terms of three maps, p_asm, s_asm_to, s_asm_from as indicated in the algorithm [lines 5-9]. Based on these maps [lines 3-9], the algorithm synthesizes a TCD component model of each transmission line, $\mathscr{M}^t$ [line 40] by creating fault nodes (physical and cyber)[lines 11-12] , discrepancies related to impedance reduction and overload [lines 13-14], intra-component certain fault edges between physical fault nodes and local discrepancies (type $D1$ and $D2$) followed by identifying all local modes for these edges [lines 15-18]. As stated in previous section, a local mode is an enumeration of a path between fault source and discrepancy associated with a given protection assembly. Since the hop count between the faulted equipment and secondary protection device is bounded due to zone reaches, enumerating all paths has polynomial complexity. After creating local edges, the algorithm creates incoming uncertain inter-component edges between fault nodes in TCD components of other lines and local discrepancy nodes (type $D2$ and $D3$) signifying impedance reduction followed by edges terminating at local overload discrepancy nodes from discrepancy nodes in the TCD component of remaining lines [lines 22-35]. In the end, TFPG node activation and de-activation events are created followed by instantiation of TTA templates [lines 37-39]. Table 6.1 shows the generated system TCD model (aggregation of two component models) for a two transmission line system as highlighted in Figure 6.15. The network consists of two transmission lines TL_p and TL_q that transfer power from two generators indicated by generator buses $B1$, $B3$ to a common load bus $B2$.

Algorithm 5: Fault Model Synthesis

```
Input: $N$
Output: $\mathscr{M}$
begin
    $G, B, T, T^{XR}, L \leftarrow$ identifyEquipment ($N$)
    $to\_bus \leftarrow$ createToBusMap ($N$)
    $from\_bus \leftarrow$ createFromBusMap ($N$)
    $a\_list\_to, a\_list\_from \leftarrow$ createAdjacencyLists ($N$)
    $p\_asm \leftarrow$ identifyPrimProtection ($N, to\_bus, from\_bus$)
    $line\_reach \leftarrow$ identifyZoneReach ($N, p\_asm, a\_list\_to, a\_list\_from$)
    $s\_asm\_to \leftarrow$ identifySecProtection ($N, line\_reach, a\_list\_to, p\_asm$)
    $s\_asm\_from \leftarrow$ identifySecProtection ($N, line\_reach, a\_list\_from, p\_asm$)
    foreach $t \in T$ do
        $F_{phy} \leftarrow$ createPhyFaults ($t$)
        $F_{cyb} \leftarrow$ createCybFaults ($p\_asm, t$)
        $D_{dist} \leftarrow$ createDistanceDiscrepancies ($p\_asm, t$)
        $D_{ovr} \leftarrow$ createOverloadDiscrepancies ($p\_asm, t$)
        $E_{intra} \leftarrow$ createIntraFaultEdges ($F_{phy}, D_{dist}$)
        $ND \leftarrow$ markEdgesCertain ($E_{intra}, ND$)
        $ET \leftarrow$ markPropagationTime ($E_{intra}, ET$)
        $EM, M_{intra}, \Omega \leftarrow$ identifyLocalModes ($E_{intra}, EM, Omega$)
        $E \leftarrow E_{intra}$
        $M \leftarrow M_{intra}$
        foreach $t_1 \in T - t$ do
            $temp \leftarrow$ createDistanceDiscrepancies ($p\_asm, t_1$)
            $temp \leftarrow temp \cup$ createOverloadDiscrepancies ($p\_asm, t_1$)
            $E^{ovr}_{inter} \leftarrow$ createInterFaultEdges ($temp, D_{ovr}$)
            $PA \leftarrow (s\_asm\_to[t_1] \cup s\_asm\_from[t_1]) \cap p\_asm[t]$
            $E^{dist}_{inter} \leftarrow \varnothing$
            $temp \leftarrow$ createPhyFaults ($t_1$)
            foreach $p \in PA$ do
                $E^{dist}_{inter} \leftarrow \cup$ createInterFaultEdges ($temp, D_{dist}$)
            end
            $ND \leftarrow$ markEdgesUnCertain ($E^{ovr}_{inter} \cup E^{dist}_{inter}, ND$)
            $ET \leftarrow$ markPropagationTime ($E^{ovr}_{inter} \cup E^{dist}_{inter}, ET$)
            $EM, M_{inter}, \Omega \leftarrow$ identifyLocalModes ($E^{ovr}_{inter} \cup E^{dist}_{inter}, EM, \Omega$)
            $E \leftarrow E \cup E^{ovr}_{inter} \cup E^{dist}_{inter}$
            $\mathscr{M} \leftarrow \mathscr{M} \cup M_{inter}$
        end
        $\Sigma_{act}, \Psi_{act} \leftarrow$ createActivationEvents ($D_{dist} \cup D_{ovr} \cup F_{phy} \cup F_{cyb}$ )
        $\Sigma_{ina}, \Psi_{ina} \leftarrow$ createInActivationEvents ($D_{dist} \cup D_{ovr}$ )
        $Q, Q_0, Tr, \Phi, \Sigma \leftarrow$ createAssemblyTTA ($p\_asm, t, \Sigma_{act}, \Sigma_{ina}$)
        $M_t \leftarrow \{F_{phy}, F_{cyb}, D_{dist}, D_{ovr}, E, M, ET, EM, ND, \Omega, \Psi_{act}, \Psi_{ina}, Q, Q_0, Tr, \Phi, \Sigma\}$
        $\mathscr{M} \leftarrow \mathscr{M} \cup M_t$
    end
end
```

Table 6.1: TCD model: Two line transmission system

Element	**TCD Component model:** TL_p	**TCD Component model:** TL_q
Faults (F)	$F_1^{TL\_p}, F_2^{TL\_p}, F_3^{TL\_p}, Fmiss^{PAj}, Fmiss^{PAk}, F\_z1\_sp^{PAj}, F\_z2\_sp^{PAj}, F\_z3\_sp^{PAj}, F\_z1\_sp^{PAk}, F\_z2\_sp^{PAk}, F\_z3\_sp^{PAk}, F\_o\_sp^{PAj}, F\_o\_sp^{PAk}, F\_sc^{PAk}, F\_so^{PAk}, F\_sc^{PAj}, F\_so^{PAj}$	$F_1^{TL\_q}, F_2^{TL\_q}, F_3^{TL\_q}, Fmiss^{PAm}, Fmiss^{PAl}, F\_z1\_sp^{PAm}, F\_z2\_sp^{PAm}, F\_z3\_sp^{PAm}, F\_z1\_sp^{PAl}, F\_z2\_sp^{PAl}, F\_z3\_sp^{PAl}, F\_o\_sp^{PAm}, F\_o\_sp^{PAl}, F\_sc^{PAl}, F\_so^{PAl}, F\_sc^{PAm}, F\_so^{PAm}$
Discrepancies (D)	$D1^{PAk}, D2^{PAk}, D3^{PAk}, O^{PAk}, D1^{PAj}, D2^{PAj}, D3^{PAj}, O^{PAj}$	$D1^{PAl}, D2^{PAl}, D3^{PAl}, O^{PAl}, D1^{PAm}, D2^{PAm}, D3^{PAm}, O^{PAm}$
Edges (E)	Illustrated in Figure 6.16	Illustrated in Figure 6.16
Duration (ET)	$[0, 0.032]\ \forall e \in E$	$[0, 0.032]\ \forall e \in E$
Edge Uncertainty (ND)	True for all dotted and False for solid edges shown in Figure 6.16	
Events (Σ)	$f_1^{TL\_p}, f_2^{TL\_p}, f_3^{TL\_p}, fmiss^{PAj}, fmiss^{PAk}, f\_z1\_sp^{PAj}, f\_z2\_sp^{PAj}, f\_z3\_sp^{PAj}, f\_z1\_sp^{PAk}, f\_z2\_sp^{PAk}, f\_z3\_sp^{PAk}, f\_d\_o\_sp^{PAj}, f\_d\_o\_sp^{PAk}, f\_sc^{PAk}, f\_sd\_o^{PAk}, f\_sc^{PAj}, f\_sd\_o^{PAj}, d\_z1^{PAk}, d\_z2^{PAk}, d\_z3^{PAk}, d\_o^{PAk}, d\_z1^{PAj}, d\_z2^{PAj}, d\_z3^{PAj}, d\_o^{PAj}, cmd\_close^{PAk}, cmd\_close^{PAj}, cmd\_open^{PAk}, cmd\_open^{PAj}, act\_sc\_close^{PAk}, act\_sc\_open^{PAk}, act\_sc\_close^{PAj}, act\_sc\_open^{PAj}, z1^{PAk}, z2^{PAk}, z3^{PAk}, trip^{PAk}, z1^{PAj}, z2^{PAj}, z3^{PAj}, trip^{PAj}, d'\_z1^{PAk}, d'\_z2^{PAk}, d'\_z3^{PAk}, d'\_o^{PAk}, d'\_z1^{PAj}, d'\_z2^{PAj}, d'\_z3^{PAj}, d'\_o^{PAj}$	$f_1^{TL\_q}, f_2^{TL\_q}, f_3^{TL\_q}, fmiss^{PAm}, fmiss^{PAl}, f\_z1\_sp^{PAm}, f\_z2\_sp^{PAm}, f\_z3\_sp^{PAm}, f\_z1\_sp^{PAl}, f\_z2\_sp^{PAl}, f\_z3\_sp^{PAl}, f\_d\_o\_sp^{PAm}, f\_d\_o\_sp^{PAl}, f\_sc^{PAl}, f\_sd\_o^{PAl}, f\_sc^{PAm}, f\_sd\_o^{PAm}, d\_z1^{PAl}, d\_z2^{PAl}, d\_z3^{PAl}, d\_o^{PAl}, d\_z1^{PAm}, d\_z2^{PAm}, d\_z3^{PAm}, d\_o^{PAm}, cmd\_close^{PAl}, cmd\_close^{PAm}, cmd\_open^{PAl}, cmd\_open^{PAm}, act\_sc\_close^{PAl}, act\_sc\_open^{PAl}, act\_sc\_close^{PAm}, act\_sc\_open^{PAm}, z1^{PAl}, z2^{PAl}, z3^{PAl}, trip^{PAl}, z1^{PAm}, z2^{PAm}, z3^{PAm}, trip^{PAm}, d'\_z1^{PAl}, d'\_z2^{PAl}, d'\_z3^{PAl}, d'\_o^{PAl}, d'\_z1^{PAm}, d'\_z2^{PAm}, d'\_z3^{PAm}, d'\_o^{PAm}$
Locations (Q)	$zone1\_IDLE^{PAk}, zone2\_IDLE^{PAk}, zone3\_IDLE^{PAk}, ovr\_IDLE^{PAk}, zone1\_MISSED^{PAk}, zone2\_MISSED^{PAk}, zone3\_MISSED^{PAk}, ovr\_MISSED^{PAk}, zone1\_TRIPPED^{PAk}, zone2\_TRIPPED^{PAk}, zone3\_TRIPPED^{PAk}, ovr\_TRIPPED^{PAk}, zone2\_WAIT^{PAk}, zone3\_WAIT^{PAk}, ovr\_WAIT^{PAk}, zone2\_FAULT-WAIT^{PAk}, zone3\_FAULT-WAIT^{PAk}, ovr\_FAULT-WAIT^{PAk}, OPEN^{PAk}, CLOSE^{PAk}, OPENING^{PAk}, CLOSING^{PAk}, STUCK\_CLOSE^{PAk}, zone1\_IDLE^{PAj}, zone2\_IDLE^{PAj}, zone3\_IDLE^{PAj}, ovr\_IDLE^{PAj}, zone1\_MISSED^{PAj}, zone2\_MISSED^{PAj}, zone3\_MISSED^{PAj}, ovr\_MISSED^{PAj}, zone1\_TRIPPED^{PAj}, zone2\_TRIPPED^{PAj}, zone3\_TRIPPED^{PAj}, ovr\_TRIPPED^{PAj}, zone2\_WAIT^{PAj}, zone3\_WAIT^{PAj}, ovr\_WAIT^{PAj}, zone2\_FAULT-WAIT^{PAj}, zone3\_FAULT-WAIT^{PAj}, ovr\_FAULT-WAIT^{PAj}, OPEN^{PAj}, CLOSE^{PAj}, OPENING^{PAj}, CLOSING^{PAj}, STUCK\_CLOSE^{PAj}$	$zone1\_IDLE^{PAl}, zone2\_IDLE^{PAl}, zone3\_IDLE^{PAl}, ovr\_IDLE^{PAl}, zone1\_MISSED^{PAl}, zone2\_MISSED^{PAl}, zone3\_MISSED^{PAl}, ovr\_MISSED^{PAl}, zone1\_TRIPPED^{PAl}, zone2\_TRIPPED^{PAl}, zone3\_TRIPPED^{PAl}, ovr\_TRIPPED^{PAl}, zone2\_WAIT^{PAl}, zone3\_WAIT^{PAl}, ovr\_WAIT^{PAl}, zone2\_FAULT-WAIT^{PAl}, zone3\_FAULT-WAIT^{PAl}, ovr\_FAULT-WAIT^{PAl}, OPEN^{PAl}, CLOSE^{PAl}, OPENING^{PAl}, CLOSING^{PAl}, STUCK\_CLOSE^{PAl}, zone1\_IDLE^{PAm}, zone2\_IDLE^{PAm}, zone3\_IDLE^{PAm}, ovr\_IDLE^{PAm}, zone1\_MISSED^{PAm}, zone2\_MISSED^{PAm}, zone3\_MISSED^{PAm}, ovr\_MISSED^{PAm}, zone1\_TRIPPED^{PAm}, zone2\_TRIPPED^{PAm}, zone3\_TRIPPED^{PAm}, ovr\_TRIPPED^{PAm}, zone2\_WAIT^{PAm}, zone3\_WAIT^{PAm}, ovr\_WAIT^{PAm}, zone2\_FAULT-WAIT^{PAm}, zone3\_FAULT-WAIT^{PAm}, ovr\_FAULT-WAIT^{PAm}, OPEN^{PAm}, CLOSE^{PAm}, OPENING^{PAm}, CLOSING^{PAm}, STUCK\_CLOSE^{PAm}$
Initial Locations (Q_0)	$zone1\_IDLE^{PAk}, zone2\_IDLE^{PAk}, zone3\_IDLE^{PAk}, ovr\_IDLE^{PAk}, zone1\_IDLE^{PAj}, zone2\_IDLE^{PAj}, zone3\_IDLE^{PAj}, ovr\_IDLE^{PAj}, CLOSE^{PAk}, CLOSE^{PAj}$	$zone1\_IDLE^{PAl}, zone2\_IDLE^{PAl}, zone3\_IDLE^{PAl}, ovr\_IDLE^{PAl}, zone1\_IDLE^{PAm}, zone2\_IDLE^{PAm}, zone3\_IDLE^{PAm}, ovr\_IDLE^{PAm}, CLOSE^{PAl}, CLOSE^{PAm}$
Activation event map (Ψ_{act})	$\Psi_{act}(D1^{PAk}) = d\_z1^{PAk}, \Psi_{act}(D2^{PAk}) = d\_z2^{PAk}, \Psi_{act}(D3^{PAk}) = d\_z3^{PAk}, \Psi_{act}(O^{PAk}) = d\_o^{PAk}, \Psi_{act}(D1^{PAj}) = d\_z1^{PAj}, \Psi_{act}(D2^{PAj}) = d\_z2^{PAj}, \Psi_{act}(D3^{PAj}) = d\_z3^{PAj}, \Psi_{act}(O^{PAj}) = d\_o^{PAj}$	$\Psi_{act}(D1^{PAl}) = d\_z1^{PAl}, \Psi_{act}(D2^{PAl}) = d\_z2^{PAl}, \Psi_{act}(D3^{PAl}) = d\_z3^{PAl}, \Psi_{act}(O^{PAl}) = d\_o^{PAl}, \Psi_{act}(D1^{PAm}) = d\_z1^{PAm}, \Psi_{act}(D2^{PAm}) = d\_z2^{PAm}, \Psi_{act}(D3^{PAm}) = d\_z3^{PAm}, \Psi_{act}(O^{PAm}) = d\_o^{PAm}$
De-activation event map (Ψ_{ina})	$\Psi_{ina}(D1^{PAk}) = d'\_z1^{PAk}, \Psi_{ina}(D2^{PAk}) = d'\_z2^{PAk}, \Psi_{ina}(D3^{PAk}) = d'\_z3^{PAk}, \Psi_{ina}(O^{PAk}) = d'\_o^{PAk}, \Psi_{ina}(D1^{PAj}) = d'\_z1^{PAj}, \Psi_{ina}(D2^{PAj}) = d'\_z2^{PAj}, \Psi_{ina}(D3^{PAj}) = d'\_z3^{PAj}, \Psi_{ina}(O^{PAj}) = d'\_o^{PAj}$	$\Psi_{ina}(D1^{PAl}) = d'\_z1^{PAl}, \Psi_{ina}(D2^{PAl}) = d'\_z2^{PAl}, \Psi_{ina}(D3^{PAl}) = d'\_z3^{PAl}, \Psi_{ina}(O^{PAl}) = d'\_o^{PAl}, \Psi_{ina}(D1^{PAm}) = d'\_z1^{PAm}, \Psi_{ina}(D2^{PAm}) = d'\_z2^{PAm}, \Psi_{ina}(D3^{PAm}) = d'\_z3^{PAm}, \Psi_{ina}(O^{PAm}) = d'\_o^{PAm}$
Timing constraints (Φ)	$(r^{PAk}), [z2wt^{PAk}], [z3wt^{PAk}], [tto^{PAk}], [ttc^{PAk}], (r^{PAj}), [z2wt^{PAj}], [z3wt^{PAj}], [tto^{PAj}], [ttc^{PAj}]$	$(r^{PAl}), [z2wt^{PAl}], [z3wt^{PAl}], [tto^{PAl}], [ttc^{PAl}], (r^{PAm}), [z2wt^{PAm}], [z3wt^{PAm}], [tto^{PAm}], [ttc^{PAm}]$
Transitions (T)	Illustrated in Figures 6.2, 6.4, 6.6, 6.8 and 6.10	

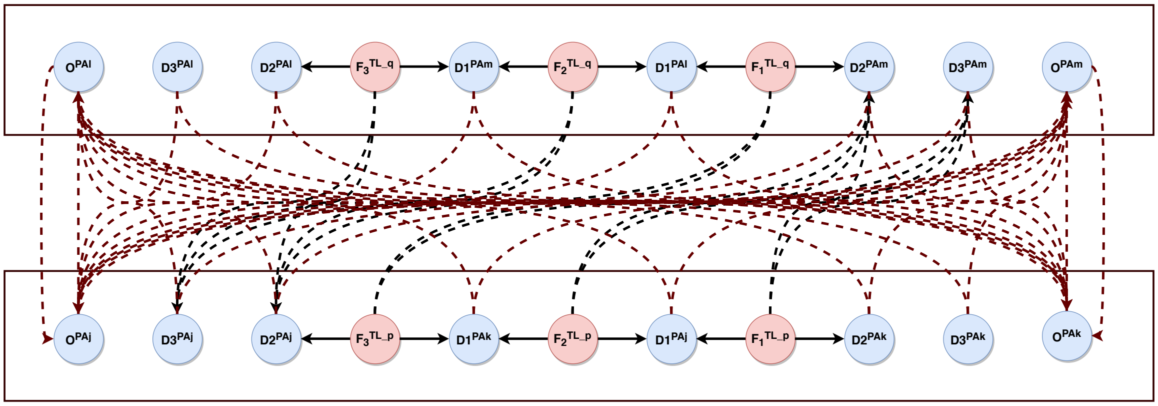


Figure 6.16: TFPG sub-model for two transmission network

6.4 Evaluation

In this section, we show the soundness of the proposed TCD fault model synthesis approach for a power transmission system (CPES). We also present the scalability results of the generation algorithm with the help of number of standard IEEE power networks. The system fault model is composed of individual fault models where each component model consists of a set of templates of the same behavioral models (TTA) and collection of fault edges representing effect of physical faults and abnormal conditions. We used a hybrid evaluation strategy where we verify the TTA models w.r.t to safety and liveness requirements encoded as TCTL properties and validate the signature of physical faults, i.e., edges between fault nodes and discrepancy nodes with extensive simulation based testing using a Python based dynamic simulator [147].

6.4.1 TTA Verification

We begin the TTA verification by identifying the different safety and liveness requirements based on the behavior of protection assembly components. These requirements are enumerated as follows

(**R1**) In the absence of cyber faults, a zone 1 fault in a transmission line should be detected by all zone elements of the associated distance protection within 32 milliseconds.

(**R2**) In the absence of cyber faults, a zone 2 fault in a transmission line should be detected by zone 2 and 3 elements of the associated distance protection within 32 milliseconds.

(R3) In the absence of cyber faults, a zone 3 fault in a transmission line should be detected by only zone 3 element of the neighboring distance protection within 32 milliseconds.

(R4) In the absence of cyber faults, overload protection should detect the abnormal conditions within 32 seconds.

(R5) In the presence of missed detection fault, distance and overload protection elements should fail to detect fault and abnormal conditions.

(R6) In the presence of zone 1, zone 2 zone 3 and overload spurious detection faults, the respective elements should trigger breaker commands irrespective of the actual faults or abnormal conditions.

(R7) In the absence of stuck close fault, a breaker should transition from close to open location after receiving open command.

(R8) In the presence of stuck close fault, a breaker should ignore the commands from protection elements.

The above mentioned requirements are evaluated in context of an arbitrary component TCD model with three physical fault nodes (F1, F2, F3) and ten cyber fault nodes, where F1, F2 and F3 models the injection of zone 1, zone 2 and zone 3 faults respectively. The fault model also consists of four discrepancy nodes and six fault propagation edges along with six TTA models labeled as *Z1element*, *Z2element*, *Z3element*, *Ovrelement*, *Breaker1* and *Breaker2*. The first three elements are associated with distance protection while the fourth is an overload protection element and the last two models the behavior of a breaker. The complete system also includes four fault injector automata, *phy*, *cyber1*, *cyber2*, *cyber3*, a mode calculator, a discrepancy de-activator and a global clock ticker. The first fault injector in the list introduces physical fault in the system while the rest injects cyber faults i.e. detection faults in distance protection, overload relay and stuck faults in breaker at a given time governed by the template parameter, *inject_time*. The TCTL properties based on the concrete example and the requirements satisfied by them are listed as follows.

(P1) `phy.S1 and cyber1.S5 --> Z1element.TRIPPED and Z2element.WAIT and Z3element.WAIT and abs_time > phy.inject_time + F1_D1.t_max`

(P2) `phy.S2 and cyber1.S5 --> Z1element.IDLE and Z2element.WAIT and Z3element.WAIT and abs_time > phy.inject_time + F2_D2.t_max`

(P3) `phy.S2 and cyber1.S5 --> Z1element.IDLE and Z2element.WAIT and Z3element.WAIT and abs_time > phy.inject_time + F3_D3.t_max`

(P4) `F1_D1.S9 and cyber1.S5 and cyber2.S3 and cyber3.S4 --> Ovrelement.WAIT and abs_time > phy.inject_time + F1_D1.t_min + breaker1.tto + D1_D4.t_max`

(P5) `F2_D2.S9 and cyber1.S5 and cyber2.S3 and cyber3.S4 --> Ovrelement.WAIT and abs_time > phy.inject_time + F1_D1.t_min + Z2elment.delay + breaker1.tto + D2_D4.t_max`

(P6) `F3_D3.S9 and cyber1.S5 and cyber2.S3 and cyber3.S4 --> Ovrelement.WAIT and abs_time > phy.inject_time + F1_D1.t_min + Z3elment.delay + breaker1.tto + D2_D4.t_max`

(P7) `cyber1.S1 --> Z1.element.MISSED and Z2.element.MISSED and Z3.element.MISSED and abs_time > cyber1.inject_time`

(P8) `cyber2.S1 --> Ovrelement.MISSED and abs_time > cyber2.inject_time`

(P9) `(cyber3.S2 or cyber3.S4) and (Z1element.TRIPPED or Z2.element.TRIPPED and Z3.element.TRIPPED) --> breaker1.OPEN`

(P10) `(cyber3.S1 or cyber3.S4) and (Ovrelement.TRIPPED) --> breaker2.OPEN`

(P11) `cyber3.S1 or cyber3.S3 --> breaker1.CLOSE and abs_time > cyber3.inject_time`

(P12) `cyber3.S2 or cyber3.S3 --> breaker2.CLOSE and abs_time > cyber3.inject_time`

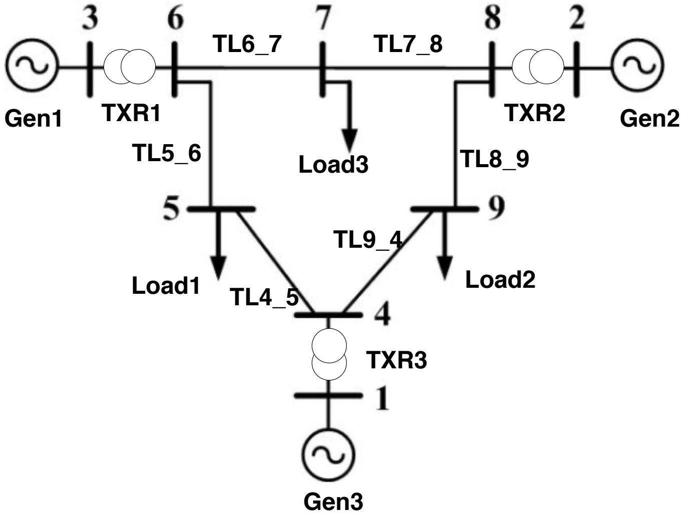


Figure 6.17: WSCC 9 Bus System

6.4.2 TFPG Validation

We validate the proposed TFPG model by proving that all feasible fault propagation paths (see definition 6.4.1) can be produced by the TFPG model. In the context of CPES, the first element in the fault propagation path represents a physical fault in a transmission line and the following element consists of a set of impedance reeduction discrepancies that constitute fault signature and the rest of the elements of the sequence are collection of multiple overload discrepancies. With the help of a simulation model [147], we first show the physical fault signature of all transmission line faults is complete and based on this result, we present a theoretical proof to demonstrate the appropriate coverage of the TFPG model.

Definition 6.4.1 (Fault propagation path)**.** *A fault propagation path is a sequence of sets of activating TFPG nodes where the initiating node is a physical fault node and the subsequent elements are collections of one or more discrepancy nodes.*

To validate the fault signature of transmission line faults, we used a standard Western System Coordinating Council network with 3 generators, 9 buses, 3 transformers, 3 loads and 6 transmission lines as highlighted in Figure 6.17. Each transmission line is divided into 10 equal segments and a fault is injected one at a time in each segment at a random location using uniform distribution. We observed that the activation of discrepancies with intra-component *certain* edge, (ND = False)

Line	$\lvert actual\rvert / \lvert expected\rvert$		$actual - expected$	
	Certain	Un-certain	Certain	Un-Certain
TL4_5	1, 0	0.47, 0.08	0, 0	0, 0
TL5_6	1, 0	0.45, 0.1	0, 0	0, 0
TL6_7	1, 0	0.47, 0.08	0, 0	0, 0
TL7_8	1, 0	0.44, 0.11	0, 0	0, 0
TL8_9	1, 0	0.44, 0.11	0, 0	0, 0
TL9_4	1, 0	0.44, 0.11	0, 0	0, 0

Table 6.2: Transmission line fault signature simulation results (mean, standard deviation)

are correctly identified by the model. However, TFPG over-approximates the inter-component *un-certain* (ND=True) discrepancies as anticipated. Table 6.2 summarises the results for each transmission line by highlighting the ratio and difference between expected and actual (simulated) activated discrepancies related to certain and un-certain edges.

Proposition 1. *The set of all feasible fault propagation paths is a subset of the set of all fault propagation paths produced by TCD model, $\mathscr{M}$.*

Proof. Let's assume there exists a feasible fault propagation path $\pi = F_a \rightarrow D_1 \rightarrow D_2 \rightarrow D_3$ which cannot be generated by the TCD model, $\mathscr{M}$, i.e., $\pi \notin \Pi_{\mathscr{M}}$, where $\Pi_{\mathscr{M}}$ is the set of all fault propagation paths. Let's assume the set of impedance reduction discrepancies generated by the TCD model is $D_1^{\mathscr{M}}$ and based on the simulation results we can safely say that $D_1 \subset D_1^{\mathscr{M}}$ which implies that there does not exist any fault propagation edge between F_a and discrepancies in the set D_1 that is excluded by set of edges between F_a and the set of discrepancies $D_1^{\mathscr{M}}$.

Similarly, the generation algorithm 5 over-approximates the overload discrepancies by creating multiple outgoing fault propagation edges to the overload discrepancies of all the remaining component models such that every successive discrepancy set consists of all overload discrepancies. Thus, all the fault propagation edges between D_1, D_2 and D_2, D_3 are covered in the set of edges between $D_1^{\mathscr{M}}$, $D_2^{\mathscr{M}}$ and $D_3^{\mathscr{M}}$. Hence our initial assumption is wrong and if π is feasible then it is member of $\Pi_{\mathscr{M}}$. □

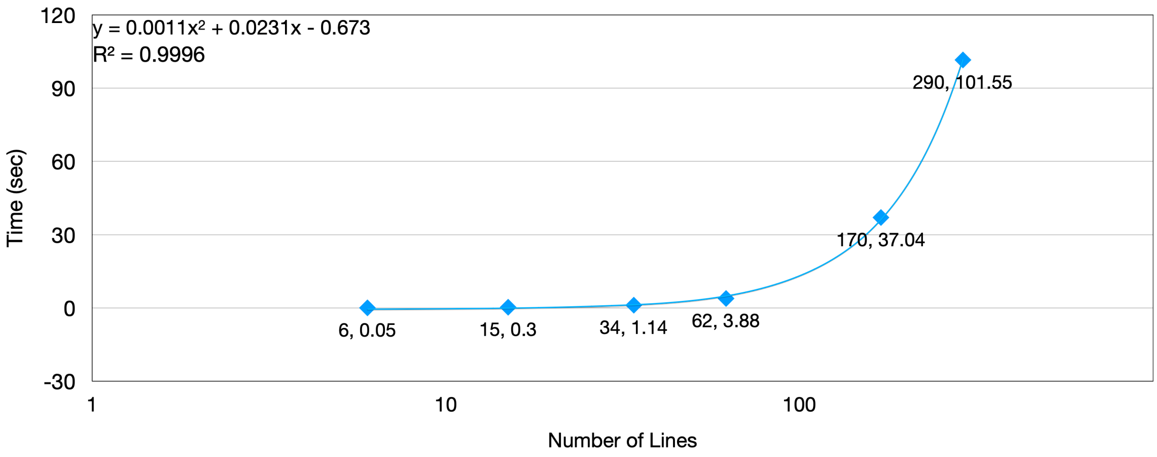


Figure 6.18: Timing analysis of the TCD generation algorithm
The annotation on the graph includes the number of lines and time to generate the TCD model

6.4.3 Scalability and Timing Analysis

The identification of inter-component fault edges and local modes is the most computational expensive task in generating a TCD model. The identification of inter-component fault edges involve identifying secondary protection assemblies for a given component whereas local mode depends upon the enumerating all paths between protection assemblies and fault location. Due to constraints imposed by the zone 2 and zone 3 thresholds, the hop count between faulty equipment and secondary protection assemblies is limited, which restricts the number of feasible paths between the two, leading to a polynomial time complexity of the generation algorithm as indicated by Figure 6.18. The y-axis denotes teh time to generate the TCD model in secs whereas x-axis has a logarithmic scale in the number of lines obtained from six standard IEEE power transmission networks, 9, 14, 39, 57, 118, 300 Bus System [5]. The scale of the generated TCD model is summarised in Table 6.3, which shows the linear rate of increase of all elements of TCD model with size of network (number of lines) except fault edges and local modes.

6.5 Summary

In this chapter, we described the component TCD model that represents 1) the effect of physical fault in transmission lines, 2) overload conditions caused due to fault isolation and 3) the response

[5] https://icseg.iti.illinois.edu/power-cases/

Components	**9 Bus**	**14 Bus**	**39 Bus**	**57 Bus**	**118 Bus**	**300 Bus**
Generators	3	5	10	7	54	69
Loads	3	11	21	42	99	196
Branches	9	20	46	78	179	409
Lines	6	15	34	62	170	290
Transformers	3	5	12	16	9	119
Fault Nodes	78	195	442	806	2210	3770
Discrepancies	48	120	272	496	1360	2320
Fault Edges	564	3798	18452	62944	495524	1356036
Locations	288	720	1632	2976	8160	13920
Transitions	504	1260	2856	5208	14280	24360
Events	300	750	1700	3100	8500	14500

Table 6.3: Scale of generated TCD models for different IEEE networks

of distance and overload protection elements while modeling the detection faults. We also presented a graph theoretic approach for generating a system TCD model from individual TCD component models without the help of any load flow or power system simulation. In the end, we validated the TCD component model using a 9 Bus System.

6.6 Contributions

We introduced TCD component model, generation algorithm and associated tool chain targeting only the physical faults in the workshop paper titled, *A component-based approach for modeling failure propagations in power systems* [148] followed by a conference paper, titled, *Towards diagnosing cascading outages in cyber physical energy systems using temporal causal models* [149] which extended the fault model to include abnormal operating conditions. Python based utility to generate TCD model from the system topology and fault models of standard IEEE networks along with UPPAAL TA templates for protection system components can be downloaded from the repository [140].

Chapter 7

Fault Diagnosis using TCD Fault Models

7.1 Problem Statement

Develop a transformation, $\mathscr{A}_2 : \left(H \times A \times \mathscr{M}^{fault} \times t\right) \rightarrow (H \times t)$, that creates a new or updates the existing hypotheses to explain the current state of the CPES on the basis of alarms produced by the protection system components and their observed state. H, A, $\mathscr{M}^{fault}$, t represent hypothesis, alarms, fault model and time respectively.

7.2 Challenges

The foremost challenge is to create a diagnosis system that produces an integrated hypothesis set by extending the classical TFPG reasoning algorithm to include the diagnosis of latent faults in protection system such as detection and stuck faults. The second challenge is to enable reasoning algorithm to handle delayed or out of order processing of received alarms. This challenge arises due to the geographically distributed nature of power system. This dispersion can lead to significant delay in the reception of an alarm due to unreliable communication channels, clock differences etc.

Protection equipment such as zone 1 element of a numerical relay can isolate faulty equipment in less than a second. These fast acting protection equipment impose constraints on the performance of the reasoning algorithm. Thus, a reasoning algorithm should be able to process alarms and produce correct hypotheses in a reasonably swift fashion so that system operator has sufficient time to perform corrective actions.

7.3 Solution Approach

We present a hierarchical approach for diagnosing faults in CPES. Figure 7.1 shows the layered architecture of the proposed diagnosis system indicating the type of components in each layer and their respective data exchange. The bottom layer consists of the power system apparatus or physical

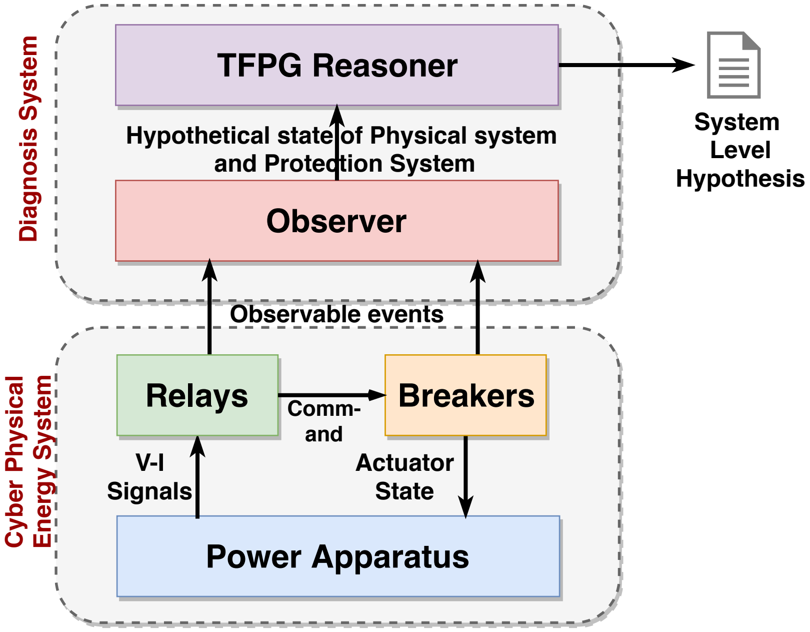


Figure 7.1: Data-flow diagram of fault diagnosis in CPES using TCD fault models

components such as generators, transmission lines, transformers, etc and the next layer consists of protection equipment such as numerical relays and breakers that are responsible for safeguarding the system. The relays in the protection layer controls the state of the breakers based on the voltage or current samples and the actuator state governs the power flow in the physical components. The amalgamation of these layers forms a CPES such as transmission system or distribution system.

The next level in the hierarchy is referred to as an Observer layer. This layer hosts a number of local diagnosers, called *Observers*, which track the behavior of protection assemblies by observing the event summaries generated by the numerical relays and hypothesize about the state of physical system using TTA models. Observers generate multiple local hypothesis in the form of alarms and forward it to the final layer. The last layer consists of a global TFPG reasoner that combines local hypothesis from all observers to create an integrated hypothesis set. Each hypothesis in the set provides reasoning about faults in both physical and cyber components based on the alarms received from observers, TFPG model and system topology. The top two layers collectively represent the TCD based diagnosis system, described in more detail in the following sections

7.3.1 Observer Layer

The responsibility of the observer layer is to parse the event summaries produced by numerical relays in different protection assemblies and estimate the state of the tracked component. For every protection assembly, the layer consists of three observers, i.e., dedicated observer routines for distance protection, overload relays and breaker. The input event summary includes time stamped events related to breaker trigger command, actuator physical state change, detection of change in impedance and line current. The function of distance and overload protection observer is to infer the activation events, d_z1^{PAk}, d_z2^{PAk}, d_z3^{PAk} and d_o^{PAk} while the breaker observer signals the changing mode conditions as well as presence of stuck faults. We refer the output of an observer as a derived alarm that is consumed by the global TFPG reasoner. A derived alarm is represented as a tuple, (e_t, e_l ,t), where e_t denotes the type of an event (activation and de-activation of TFPG nodes or breaker state change), e_l is an identifier the signifies the label associated with the tracked element in the protection assembly and t is the hypothetical time of occurrence of the respective event in the physical system. Each observer is modeled as UPPAAL TA and the following subsections describe them in detail.

7.3.1.1 Distance Protection Observer

Distance protection observer signals the presence of physical fault conditions and determine the location of fault relative to the protection assembly i.e zone 1, 2 or 3 by inferring the presence of events, d_z1^{PAk}, d_z2^{PAk} or d_z3^{PAk}. Figure 7.2 shows the UPPAAL TA template of the distance protection observer that tracks three zone elements based on the TTA models described in the previous chapter. The automaton consists of twenty locations while `S1` being the initial location. The observer responds to the input events, $z1$, $z2$, $z3$, $z1\_ina$, $z2\_ina$, $z3\_ina$, cmd_open serialized in the event summaries and generates six local hypotheses represented by boolean variables, hyp_Z1_act, hyp_Z1_ina, hyp_Z2_act, hyp_Z2_ina, hyp_Z3_act and hyp_Z3_ina, where the evaluations, $hyp\_Z_j\_act$ = `True` and $hyp\_Z_j\_ina$ = `True` indicate the presence of activation event $d\_z_j^{PAk}$ and de-activation event $d'\_z_j^{PAk}$ respectively.

In the absence of cyber faults, the activation event, d_z1^{PAk} forces all the zone element automata to leave `IDLE` location and generate observable events $z1$, $z2$, $z3$ related to detection of fault conditions. Based on this expected behavior, observer transitions out of `S1` to finally reach `S8` while setting the variable hyp_z1_act. The automaton can take a total of six paths depending upon the order the events are observed. For instance, if the order of observance is d_z2, d_z3, d_z1 then the observer will follow the path, $\texttt{S1} \rightarrow \texttt{S2} \rightarrow \texttt{S4} \rightarrow \texttt{S8}$. At each intermediate location, the automaton waits for at most *lat* secs which is equal to the sum of maximum detection time of fault conditions (t_{max}) and propagation delay (δ). In the current implementation, *lat* is chosen to be 20 milliseconds where δ is assumed to be 4 milliseconds and the value of t_{max} is 16 milliseconds as per relay datasheet [16]. For d_z2^{PAk}, only zone 2 and 3 element automata should transition out of `IDLE` location leading to generation of $z2$ and $z3$ events. These events force the observer automaton to transition to `S4` via `S2` or `S3`, depending upon the order of observance. While in `S4`, the observer waits for *lat* seconds before concluding the presence of d_z2^{PAk} by setting hyp_z2_act variable if no other event is detected. Similarly, for d_z3^{PAk} only zone 3 element should generate $z3$ event forcing the observer to jump to `S2` and deducing the presence of d_z3^{PAk} by setting the variable hyp_z3_act, if no other event is detected in *lat* seconds.

The distance protection observer cannot diagnose cyber or detection faults due to the absence of the indicator events that distinguish between nominal and faulty behavior as described in the paper [150]. The missed detection fault forces all zone elements and the corresponding observer to ignore the fault conditions. While the spurious detection fault causes the zone elements to generate events under no phyiscal fault conditions compelling the observer to incorrectly deduce the presence of physical faults. Table 7.1 summaries the response of the observer to various combinations of spurious detection faults based on Figure 7.2. These detection faults are diagnosed by the TFPG reasoner based on derived alarms received from multiple observers. The other limitation of the observer model is that it assumes a minimum temporal separation between fault injection events in the physical. This time separation is required by the observer to correctly distinguish the two separately occurring fault conditions. In the current implementation, the time period is equal to the zone 3 wait time i.e. 1.5 secs.

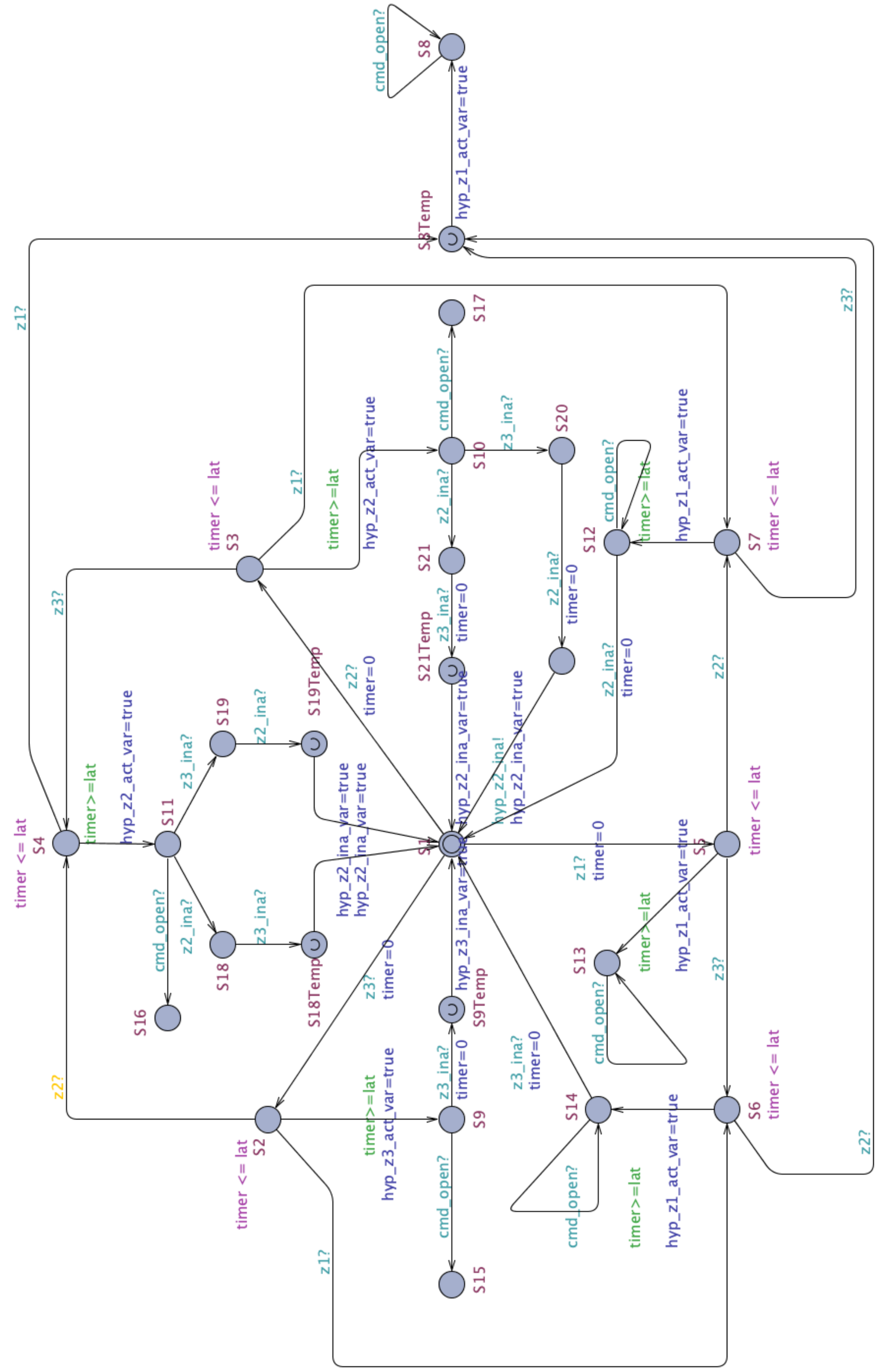


Figure 7.2: Distance protection observer UPPAAL TA

Table 7.1: Observer response to zone element spurious detection faults

Zone 1	Zone 2	Zone 3	Output
Yes	X	X	hyp_z1_act ← True
No	Yes	X	hyp_z2_act ← True
No	No	Yes	hyp_z3_act ← True

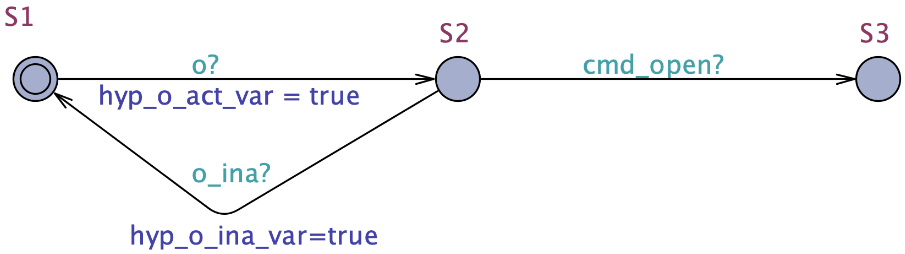


Figure 7.3: Overload protection observer UPPAAL TA

7.3.1.2 Overload Protection Observer

Overload protection observer is responsible for detecting the presence and absence of branch overload in the physical system based on the event summaries received from numerical relays. Figure 7.3 shows the UPPAAL TA template of the observer that tracks an overcurrent element. The automaton has three locations with `S1` being the initial location. The observer reacts to events o^{PAk}, o_ina^{PAk} and cmd_open where o and o_ina are related to assertion and negation of trigger condition of the overcurrent element. The observer generates two hypothesis represented by the events, *hyp_o_act* and *hyp_o_ina* to signal the presence and absence of overload conditions. While in initial location `S1`, the observer waits for the overload relay to signal overload conditions i.e. observance of o. After detecting the overload conditions, the observer signals generates the event *hyp_o_act* and transitions to `S2` locations. While in `S2`, the observer either waits for o' or *cmd_open* events. If the *o_ina* is detected then observer declares the absence of overload conditions by emanating *hyp_o_ina* and transitioning back to `S1`. On the other hand if the overload conditions persists and observer detects a breaker command, it transitions to `S3`. Similar to distance protection observer, this observer is not able to diagnose detection faults and will depend upon TFPG reasoner to revise its hypothesis.

7.3.1.3 Breaker Observer

Figure 7.4 illustrates the UPPAAL TA template of an observer that tracks operation of a breaker. The breaker observer indicates mode change events i.e. when the physical state of the breaker changes from open to close or close to open and the presence of stuck faults. The automaton consists of four locations with `S1` being the initial location. The observer tracks the breaker by extracting events such as cmd_open, cmd_close, act_sc_open and act_sc_close from the event summaries. The events cmd_open, cmd_close relates to the bit change when a numerical relays instructs a breaker to open and close and st_open, st_close denotes the change in the state of breaker to open and close.

The observer transitions from `S1` to `S2` after observing cmd_open event and waits for for $tto+lat$ secs to detect act_sc_open. The expression, $tto+lat$ is the sum of the actual time to open the breaker (50 milliseconds) and the delay associated with transmission and parsing of event summaries (4 milliseconds). If the automaton observes the event before the deadline, it moves to location `S3` and emits mode change event, denoted by setting the variable $hyp_act\_sc\_open\_var$ to true. On the other hand, if the event is not observed, the automaton moves back to location `S1` and concludes the presence of stuck close fault by setting the variable $hyp\_f\_stuck\_close\_var$ to true . In a similar fashion, the automaton jumps from location `S3` to `S4` after observing cmd_close event and while in `S4`, the observer waits for $ttc+lat$ secs to detect act_sc_close event before concluding stuck close fault. If the event is detected, the automaton jumps to `S1` while setting the variable $hyp_act\_sc\_close\_var$ to true, otherwise stuck fault is signaled by setting $hyp\_f\_stuck\_open\_var$ to true while transitioning back to `S3`.

7.3.2 TFPG Reasoner Layer

Observer layer cannot diagnose missed and spurious detection faults as distance and overload protection observers generate derived alarms based on limited local information. A global diagnoser is required to produce an integrated hypothesis set based on the derived alarms from the observer layer that includes reasoning about the physical faults in power equipment, missed or spurious detection faults in protection relays and stuck faults in breakers. The reasoner uses TFPG sub-model of system TCD model that link alarms from different observers to fault nodes as described in last chapter. The underlying reasoning algorithm is graph theoretic in nature and exploits consistency

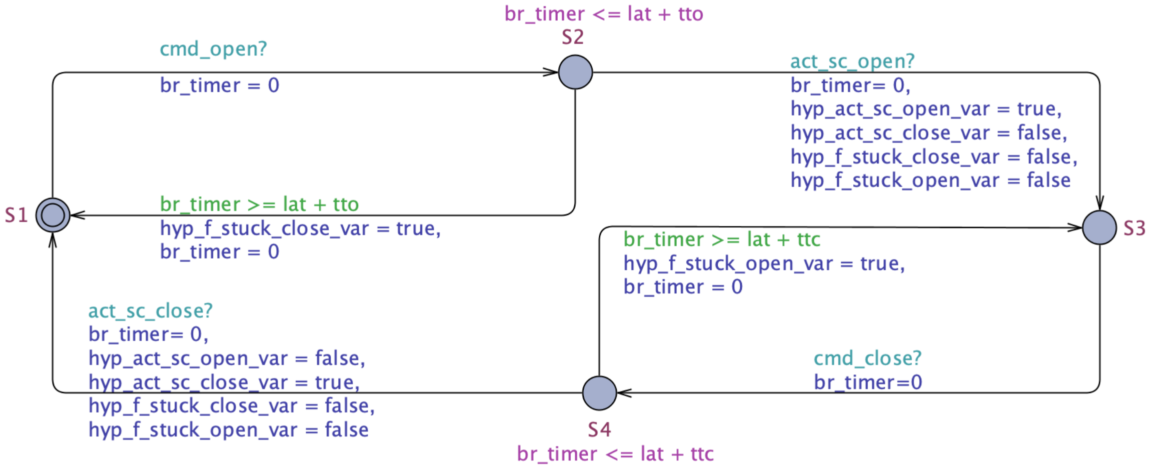

br_timer <= lat + tto
S2
cmd_open?
br_timer = 0
act_sc_open?
br_timer= 0,
hyp_act_sc_open_var = true,
hyp_act_sc_close_var = false,
hyp_f_stuck_close_var = false,
hyp_f_stuck_open_var = false
S1
br_timer >= lat + tto
hyp_f_stuck_close_var = true,
br_timer = 0
S3
br_timer >= lat + ttc
hyp_f_stuck_open_var = true,
br_timer = 0
act_sc_close?
br_timer= 0,
hyp_act_sc_open_var = false,
hyp_act_sc_close_var = true,
hyp_f_stuck_close_var = false,
hyp_f_stuck_open_var = false
cmd_close?
br_timer=0
S4
br_timer <= lat + ttc

Figure 7.4: Breaker observer behavior model

relationship between TFPG nodes defined using three state mappings: *physical*, *observed*, and *hypothetical*.

A physical state corresponds to the actual state of all nodes in the TFPG model. At any time t, the physical state is given by a map $PS_t : V \rightarrow \{ON, OFF\} \times \mathbb{R}_+$, where V is the set of nodes in the TFPG model. An ON state for a node indicates that the fault effect has reached the node (activated), otherwise, it is set to OFF. We use attributes $PS_t(v)$.state and $PS_t(v)$.time to refer the state of the node v and the last time it is changed. A similar map, EPS_t defines the state of edges based on the current mode of the system. The state of an edge, $e \in E$ is ON if $m \in EM(e)$, where m is the current system mode and EM is the edge mode map defined in TCD model.

An observed state at time t is defined by two maps $OS_t : D \rightarrow \{ON, OFF\} \times \mathbb{R}_+$, $EOS_t : E \rightarrow \{ON, OFF\} \times \mathbb{R}_+$ where D is the set of discrepancy nodes and E is set of fault propagation edges. The observed state of the system (node + edges) may not be consistent with the physical state due to potential alarm failures. Consistency is a a binary relation on the set of observed states for adjacent nodes at a given time and is defined in terms of the causality and propagation timing information expressed in the TFPG model. Formally, for any two TFPG nodes d and d', we say d is timing consistent with d' at time t if there exists an edge, i.e., $(d, d') \in E$ and one of the equations (7.1) - (7.4) are satisfied.

$$OS_t(d).\text{state} == \texttt{OFF} \,\wedge\, OS_t(d').\text{state} == \texttt{OFF} \tag{7.1}$$

$$OS_t(d).\text{state} == \texttt{ON} \,\wedge\, OS_t(d').\text{state} == \texttt{OFF} \,\wedge\, EOS_t(d,d').\text{state} \;==\; \texttt{OFF} \tag{7.2}$$

$$\begin{gathered} OS_t(d).\text{state} == \texttt{ON} \,\wedge\, OS_t(d').\text{state} == \texttt{OFF} \,\wedge\, EOS_t(d,d').\text{state} \;==\; \texttt{ON} \wedge \\ t < \max(OS_t(d).\text{time}\,, EOS_t(d,d').\text{time}) + ET(d,d').\text{tmin} \end{gathered} \tag{7.3}$$

$$\begin{gathered} OS_t(d).\text{state} == \texttt{ON} \,\wedge\, OS_t(d').\text{state} == \texttt{ON} \wedge \\ ET(d,d').\text{tmax} \geq OS_t(d').\text{time} - \max(OS_t(d).\text{time}\,, EOS_t(d,d').\text{time}) \geq ET(d,d').\text{tmin} \end{gathered} \tag{7.4}$$

The aim of the reasoning process is to find a consistent and plausible explanation of the physical state based on the observed state. Such an explanation is given in the form of a valid hypothetical state. Similar to physical states, hypothetical states are defined for both discrepancies and fault nodes. A hypothetical state defines state of all TFPG nodes and the interval in which each node changes its state. Formally, it is given by a map, $HS_t : V \to \{ON, OFF\} \times \mathbb{R}_+ \times \mathbb{R}_+$. We will write $HS(v)$.terl, $HS(v)$.tlat to indicate the estimated earliest and latest time of state change of node v. A hypothetical state is an estimation of the physical state of all nodes in the system which must be consistent with the modal and temporal constraints of the underlying TFPG. Formally, we say the hypothetical state, HS_t of a node d, is consistent if

- $HS_t(d) == \texttt{OFF}$ and one of the equations (7.5) or (7.6)) holds true for all any edge $(v,d) \in E$.
- $HS_t(d) == \texttt{ON}$ and both of the equations (7.7) and (7.8) are true for any edge $(v,d) \in E$.

$$HS_t(v) == \texttt{OFF} \tag{7.5}$$

$$\begin{gathered} HS_t(v) == \texttt{ON} \,\wedge\, EOS_t(v,d).\text{state} == \texttt{ON} \wedge \\ t < \max(HS_t(v).\text{tlat},\, EPS_t(v,d).\text{time}) + ET(v,d).\text{tmin} \end{gathered} \tag{7.6}$$

$$HS_t(d).\text{terl} \geq \min_{v \in \mathscr{U}_d} \left(HS_t(v).\text{terl} + ET(v,d).\text{tmin} \right) \tag{7.7}$$

$$HS_t(d).\text{tlat} \leq \min_{v \in \mathscr{U}_d} \left(HS_t(v).\text{tlat} + ET(v,d).\text{tmax} \right) \tag{7.8}$$

where E, V is the set of all fault propagation edges and nodes in system TCD model. $\mathcal{U}_d$ is a set of parent nodes of d such that hypothetical state of each element in $\mathcal{U}_d$ is ON i.e. $\mathcal{U}_d = \{v \in V | (v,d) \in E \textit{ and } HS_t(v).\text{state} = \texttt{ON}\}$.

7.3.2.1 Hypothesis Structure and Ranking

A consistent hypothetical state of the system, i.e., valuation of HS_t, constitutes a hypothesis. Theoretically, the number of plausible hypotheses can grow exponentially w.r.t observed alarms over time. The mean time between failure rates for numerical relays with PUTT scheme is considered 75-100 years [16], which implies the event with large number of numerical relays failing in a same year to have very low probability. Based upon this fact, we filter out those hypotheses that lists more than 5 detection faults in order to limit the size of hypothesis set from increasing exponentially. Furthermore, we use law of parsimony to rank hypotheses in the set, which suggests if a hypothesis can explain consistently all of the observed events, then it should be considered more plausible than the one, which additionally requires the assumption of alarm failure (protection system faults). Thus generated hypotheses are ranked based on the ability of a hypothesis to explain all observed alarms with minimum number of physical and protection system faults. For better readability, a hypothesis can also be viewed as a collection of projected fault nodes with their supporting and inconsistent observed alarms. Formally, we define hypothesis as a tuple $h = < F_h, C_h, I_h, M_h, E_h >$, where

- $F_h \subset F$ is a set of faults projected by the hypothesis, h. The set is further divided into two disjoint sets, F_h^{phy} and F_h^{cyber} based on the nature of faults i.e. physical or protection system.
- $C_h \subset D$ is a set of discrepancies that support the hypothesis h.
- $I_h \subset D$ is a set of discrepancies that are inconsistent with the hypothesis h. An observed discrepancy can be inconsistent w.r.t to a hypothesis if it violates consistency constraints defined in previous section.
- $M_h \subset D$ is a set of discrepancies that were expected to be observed but did not signal.
- $E_h \subset D$ is a set of discrepancies that are expected to signal in the future according to the hypothesis.

7.3.2.2 TFPG Reasoner

The objective of the reasoning process is to provide a set of consistent state estimation (hypotheses) based on the observations that closely matches the actual state of the system. The TFPG reasoner creates new hypotheses or updates the existing set in response to types of events. We classify these triggering events into two categories, 1) *External* and 2) *Internal*.

- An external event correspond to the derived alarms received from the observers. These alarms can be further divided into two groups, *Node State Change* and *Mode Change*. A node state change event occurs when a distance or overload protection observer sends a derived alarm related to activation or de-activation of a TFPG node such as $d\_z_j^{PAk}$, $d'\_z_j^{PAk}$ etc. Whereas a mode change event occurs when the derived alarm is received from a breaker and is related to change in physical status such as *mode_change_close*.

- An internal event is generated for a given hypothesis, h on the premise that a specific expected alarm, d will not be observed by a future deadline, t_{co}^d. These events are also referred to as time-out events and represented by a tuple (h, d, t_{co}^d).

The reasoner executes the sequence of steps highlighted in algorithm 6 after receiving an event. The algorithm handles different types of events i.e. node state change, timeout or mode change, by calling functions `handleNodeStateChange` (algorithm 8), `handleTimeOut` (algorithm 13) and `hanleModeChange` (algorithm 12) respectively. The input to the master algorithm includes, event to be processed (e_k), hypothesis set after processing k-1 event ($\mathscr{H}_{k-1}$), the equipment status map ($\mathscr{U}_{k-1}$), TCD model ($\mathscr{M}$), network topology as collection of adjacency lists ($\mathscr{G}$) observed node state map (OS_{k-1}), observed edge state map (EOS_{k-1}), timestamp of last processed event (t_{k-1}) and time indexed set or registry of reasoner response to previous events ($\mathscr{I}_{k-1}$). The output of the algorithm includes the updated hypothesis set $\mathscr{H}_k$, equipment status map $\mathscr{U}_k$, node state map OS_k, edge state map EOS_k and the set of timeout events T to be executed in future. The algorithm starts by inspecting the timestamp of the incoming event to detect the out of order events. If the time-stamp of the incoming event is older (less) than the previous processed event, $\mathscr{T}(e_k) < t_{k-1}$, then e_k is considered as out-of-order (Line 2). After detecting an out-of-order event, the reasoning algorithm roles back the $\mathscr{I}_{k-1}$ to event e_j such that the associated timestamp $\mathscr{T}(e_j)$ is less than $\mathscr{T}(e_k)$ (Line

4) using function `rollBack`. The output of the function includes the updated reasoner response registry $\mathscr{I}_j$, hypothesis set $\mathscr{H}_j$, equipment status $\mathscr{U}_j$, node state map OS_j, edge state map EOS_j and time-sorted collection of events, $E^{t_{k-1}}_{\mathscr{T}(e_k)}$ between the time-stamps $\mathscr{T}(e_k)$ and t_{k-1}. The reasoner processes every event in $E^{t_{k-1}}_{\mathscr{T}(e_k)}$ by calling function `handleEvent` (Line 7). After processing each event, the hypothesis set, equipment status, node state map, edge state map and registry of reasoner response is updated and used for processing subsequent event. The generated timeout events are added back to the set $E^{t_{k-1}}_{\mathscr{T}(e_k)}$ if the timestamp of an event lies between $\mathscr{T}(e_k)$ and t_{k-1}, or otherwise added to the set T (Lines 8-11).

The routine `handleEvent` (algorithm 7) identifies the type of the incoming event through a function `getType` (Line 3) and dispatches `handleNodeStatChange`, `handleModeChange` and `handleTimeout` if the type of the event is `NodeStateChange` (Lines 4-6), `ModeChange` (Line 7-9) and `Timeout` (Lines 10-12) respectively. The function is also responsible for storing the reasoner response in $\mathscr{I}_k$ based on the output of above mentioned function calls (Line 13). The function `handleNodeStateChange` (algorithm 8) begins by identifying the nature of the state change i.e. activation or de-activation event (Line 3). An activation event must be a member of domain of the function Ψ_{act} and the corresponding label is given by the inverse map $\Psi^{-1}_{act}(\mathscr{L}(e_k))$ where $\mathscr{L}$ is the labeling function (Line 4). The function further calls two procedures depending upon the kind of the node, i.e. Fault or Discrepancy (Lines 5-8). On the other hand if the event, e_k is related to node de-activation, then corresponding discrepancy label is given by the Ψ^{-1}_{ina} (Line 11) and node state map is updated (Lines 12-13).[1]

The algorithm 9 illustrates the sequence of steps followed by reasoner after encountering fault node activation event. This derived alarm can only be related to stuck fault associated with breakers in protection assemblies as rest of the detection faults are not diagnosable by the observer layer. The reasoner updates the hypothesis set by iterating over each hypothesis and adding the stuck fault to the fault set by using function `addFaultNode` (Lines 3-5) followed by updating the remaining maps (Lines 6-10).

The function `handleDiscrepancyActivation` (algorithm 10) shows the procedure executed by the reasoner after receiving an event e_k related to activation of the discrepancy, *label*. The algorithm begins by updating the node map OS_k and creating a copy of the hypothesis set $\mathscr{H}_{k-1}$ as

[1] Due to the assumption of persistent faults, fault node de-activation event is not possible.

H_{copy}. The reasoner iterates over each hypothesis $h \in H$ and check for the timing consistency for the discrepancy *label* using function `checkTimingConsistency` (Line 7), which implements the constraints eq. (7.5) - eq. (7.8) as described in previous section. If *label* is deemed consistent then the flag `isExplained` is set to `True` and its moved from the expected set to consistent set (Line 8-9). The expected set of the hypothesis h is also updated with the child nodes that are not part of consistent set and corresponding timeout events are added to T (Lines 11-16). On the other hand, if the discrepancy *label* is inconsistent then its added to inconsistent set and corresponding spurious detection faults are added to the hypothesis (Lines 17-20). If none of the existing hypothesis is able to explain the observed alarm, denoted by `False` value of the flag `isExplained` then reasoner tries to create new hypotheses using `createNewHypothesis` (Line 24). If the current event is the only event i.e. the size of hypothesis set is 0 then the reasoner adds $|H|+1$ hypothesis to $\mathscr{H}_k$ (Line 25-17), where the extra hypothesis is also referred as *null hypothesis set* which marks every received discrepancy activation alarm as spurious. On the other hand, if the if the size of existing hypothesis set $\mathscr{H}_{k-1}$ is not 0, then the reasoner creates $\mathscr{H}_k$ by merging the newly created hypothesis set H with $\mathscr{H}_{k-1}$ (Line 29-31) followed by updating the inconsistent discrepancy set of every hypothesis $h \in H$ to include all active alarms except e_k (Line 32-37) and add to $\mathscr{H}_k$ (Line 38).

The reasoner executes the sequence of steps highlighted in algorithm 12 to process a mode change event. A mode change event is generated by the observer when a breaker changes its location from close to open. The algorithm begins by updating the equipment status map $\mathscr{U}_k$ (Lines 3-6) followed by edge state map, EOS_k (Lines 7-11). The reasoner iterates over every hypothesis in the set to re-evaluate the expected discrepancies by clearing the existing set and then creating new timeout events for every child node of each consistent discrepancy thats not part of missed set (Lines 15-23). The function `handleTimeout` is executed after encountering a timeout event. A timeout event contains extra information such as 1) hypothesis identifier, 2) discrepancy node that should have been signaled and 3) the parent node. The reasoner updates every hypothesis in the set by examining the expected set and if the target discrepancy is the member of the expected set, then its moved from the expected to missed set by calling the function `moveFromExpectedToMissed`. Please not that the function examines the uncertainty flag and takes into account the possibility of other timeout events that list the same discrepancy as target before removing it from the expected set.

Algorithm 6: TFPG Reasoning Algorithm

```
Input: $e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $t_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$
Output: $\mathcal{H}_k$, $\mathcal{U}_k$, $OS_k$, $EOS_k$, $\mathcal{I}_k$, $T$
begin
    if $\mathcal{T}(e_k) < t_{k-1}$ then
        /* Out of order event */
        $E^{t_{k-1}}_{\mathcal{T}(e_k)}$, $\mathcal{H}_j$, $\mathcal{U}_j$, $\mathcal{I}_j$, $OS_j$, $EOS_j \leftarrow$ rollBack($\mathcal{I}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{T}(e_k)$)
        $H \leftarrow \mathcal{H}_j$ , $U \leftarrow \mathcal{U}_j$ , $I \leftarrow \mathcal{I}_j$ , $OS \leftarrow OS_j$ , $EOS \leftarrow EOS_j$
        foreach $e \in E^{t_{k-1}}_{\mathcal{T}(e_k)}$ do
            $H,U,I,OS,EOS,A \leftarrow$ handleEvent($e,H,U,I,OS,EOS,\mathcal{M},\mathcal{G}$)
            foreach $a \in A$ do
                if $\mathcal{T}(a) > t_{k-1}$ then $T \leftarrow T \cup a$
                else $E^{t_{k-1}}_{\mathcal{T}(e_k)} \leftarrow E^{t_{k-1}}_{\mathcal{T}(e_k)} \cup a$
            end
        end
    else
        /* In order event */
        $\mathcal{H}_k,\mathcal{U}_k,OS_k,EOS_k,\mathcal{I}_k,T \leftarrow$
         handleEvent($e_k,\mathcal{H}_{k-1},\mathcal{U}_{k-1},OS_{k-1},EOS_{k-1},\mathcal{I}_{k-1},\mathcal{M},\mathcal{G}$)
    end
end
```

Algorithm 7: Function:handleEvent

```
Function handleEvent
    Input: $e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$
    Output: $\mathcal{H}_k$, $\mathcal{U}_k$, $OS_k$, $EOS_k$, $\mathcal{I}_k$, $T$
    begin
        switch getType($e_k$) do
            case NodeStateChange do
                $\mathcal{H}_k,\mathcal{U}_k,OS_k,EOS_k,T \leftarrow$ handleNodeStateChange($e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$,
                 $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$)
            end
            case ModeChange do
                $\mathcal{H}_k,\mathcal{U}_k,OS_k,EOS_k,T \leftarrow$ handleModeChange($e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$,
                 $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$)
            end
            case Timeout do
                $\mathcal{H}_k,\mathcal{U}_k,OS_k,EOS_k,T \leftarrow$ handleTimeout($e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$,
                 $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$)
            end
            $\mathcal{I}_k \leftarrow (e_k,\mathcal{H}_k,\mathcal{U}_k,OS_k,EOS_k)$
        end
    end
end
```

Algorithm 8: Function: handleNodeStateChange

```
Function handleNodeStateChange
    Input: $e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{M}$, $\mathcal{G}$
    Output: $\mathcal{H}_k$, $\mathcal{U}_k$, $OS_k$, $EOS_k$, $T$
    begin
        if $\mathcal{L}(e_k) \in \texttt{Domain}(\Psi_{act}^{-1})$ then
            $label \leftarrow \Psi_{act}^{-1}(\mathcal{L}(e_k))$
            if $label \in F$ then
                $\mathcal{H}_k, \mathcal{U}_k, OS_k, EOS_k, T \leftarrow$ handleFaultActivation($e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$,
                  $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$, $label$)
            else
                $\mathcal{H}_k, \mathcal{U}_k, OS_k, EOS_k, T \leftarrow$ handleDiscrepancyActivation($e_k$, $\mathcal{H}_{k-1}$,
                  $\mathcal{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{I}_{k-1}$, $\mathcal{M}$, $\mathcal{G}$, $label$)
            end
        else
            $label \leftarrow \Psi_{ina}^{-1}(\mathcal{L}(e_k))$
            $OS_k \leftarrow OS_{k-1}$
            $OS_k(label) \leftarrow (\texttt{OFF}, \mathcal{T}_{e_k})$
            $\mathcal{H}_k \leftarrow \mathcal{H}_{k-1}$
            $\mathcal{U}_k \leftarrow \mathcal{U}_{k-1}$
            $EOS_k \leftarrow EOS_{k-1}$
            $T \leftarrow \varnothing$
        end
    end
end
```

Algorithm 9: Function: handleFaultActivation

```
Function handleFaultActivation
    Input: $e_k$, $\mathcal{H}_{k-1}$, $\mathcal{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathcal{M}$, $\mathcal{G}$, $label$
    Output: $\mathcal{H}_k, \mathcal{U}_k, OS_k, EOS_k, T$
    begin
        foreach $h \in H_{k-1}$ do
            $h \leftarrow$ addFaultNode($h$, $label$, $\mathcal{T}(e_k)$)
        end
        $OS_k \leftarrow OS_{k-1}$
        $\mathcal{H}_k \leftarrow \mathcal{H}_{k-1}$
        $\mathcal{U}_k \leftarrow \mathcal{U}_{k-1}$
        $EOS_k \leftarrow EOS_{k-1}$
        $T \leftarrow \varnothing$
    end
end
```

Algorithm 10: Function: handleDiscActivation

```
Function handleDiscActivation
    Input: $e_k$, $\mathscr{H}_{k-1}$, $\mathscr{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathscr{M}$, $\mathscr{G}$, $label$
    Output: $\mathscr{H}_k, \mathscr{U}_k, OS_k, EOS_k, T$
    Initialization: $isExaplained \leftarrow False$
    begin
        $OS_{k-1}(label) \leftarrow$ (ON, $\mathscr{T}(e_k)$)
        $OS_k \leftarrow OS_{k-1}$ , $EOS_k \leftarrow EOS_{k-1}$ , $\mathscr{U}_k \leftarrow \mathscr{U}_{k-1}$ , $H_{copy} \leftarrow \mathscr{H}_{k-1}$
        foreach $h \in H_{copy}$ do
            if checkTimingConsistency($h, label, OS_k, EOS_k$) then
                isExplained ← True
                $h \leftarrow$ moveFromExpectedToConsistent($h, label$)
                $children \leftarrow$ getReachableChildren($label, \mathscr{U}_k, EOS_k$)
                foreach $child \in children$ do
                    if $child \notin$ getConsistent($h$) then
                        $h \leftarrow$ addToExpected($h, child, ND(label, child)$)
                        $T \leftarrow T \cup$ Timeout($h, label, child, ET(label, child)$.tmax)
                    end
                end
            else
                $h \leftarrow$ addToInconsistent($h, label, \mathscr{T}(e_k)$)
                $h \leftarrow$ addFaultNode($h$, getSpuriousFault($label$))
            end
        end
        $\mathscr{H}_k \leftarrow H$
        if $\neg isExplained$ then
            $(H, A) \leftarrow$ createHypothesis($e_k$, $\mathscr{U}_k$, $OS_k$, $EOS_k$, $\mathscr{M}$, $\mathscr{G}$, $label$)
            if $|\mathscr{H}_{k-1}| < 1$ then
                $\mathscr{H}_k \leftarrow H \cup$ Hypothesis(getSpuriousFault($label$))
            else
                foreach $h \in H$ do
                    foreach $hyp \in \mathscr{H}_{k-1}$ do
                        $\mathscr{H}_k \leftarrow \mathscr{H}_k \cup$ merge($h, hyp$)
                    end
                    foreach $n \in OS_k$ do
                        if $OS_k(n) \wedge n \neq label$ then
                            $h \leftarrow$ addToInconsistent($h, n$)
                            $h \leftarrow$ addFaultNode($h$, getSpuriousFault($n$))
                        end
                    end
                    $\mathscr{H}_k \leftarrow \mathscr{H}_k \cup h$
                end
            end
            $T \leftarrow T \cup A$
        end
    end
end
```

Algorithm 11: Function: createHypothesis

```
Function createHypothesis
    Input: $e_k$, $\mathscr{U}_k$, $OS_k$, $EOS_k$, $\mathscr{M}$, $\mathscr{G}$, $label$
    Output: $H$, $T$
    begin
        $P \leftarrow$ getReachableParentFaultNodes($label$, $\mathscr{U}_k$, $EOS_k$)
        foreach $p \in P$ do
            $h \leftarrow$ Hyothesis($p$)
            $h \leftarrow$ addToConsistent($h, label$)
            $tmax_{occ} \leftarrow \mathscr{T}(e_k) + ET(p, label)$.tmin
            if $\mathscr{T}(e_k) - ET(p, label)$.tmax $\leq EOS_k(p, label)$.time $\leq$
              $\mathscr{T}(e_k) - ET(p, label)$.tmin then $tmax_{occ} \leftarrow ET(p, label)$.tmax

            $children \leftarrow$ getReachableChildren($p, \mathscr{U}_k, EOS_k$)
            foreach $child \in children$ do
                if $child \neq label \wedge \neg OS_k(child)$.state then
                    $tmax \leftarrow \max(tmax_{occ}, EOS_k(p, child)$.time$) + ET(p, child)$.tmax
                    if $tmax < \mathscr{T}(e_k)$ then
                        $h \leftarrow$ addToMissing($child, ND(p, child)$)
                    else
                        $h \leftarrow$ addToExpected($h, child, ND(p, child)$)
                        $T \leftarrow T \cup$ Timeout($h, p, child, tmax$)
                    end
                end
            end
            $H \leftarrow H \cup h$
        end
    end
end
```

Algorithm 12: Function: handleModeChange

```
Function handleModeChange
    Input: $e_k$, $\mathscr{H}_{k-1}$, $\mathscr{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathscr{M}$, $\mathscr{G}$, $label$
    Output: $\mathscr{H}_k, \mathscr{U}_k, OS_k, EOS_k, T$
    begin
        $OS_k \leftarrow OS_{k-1}$
        $equipment \leftarrow$ getEquipment($label$)
        $\mathscr{U}_k \leftarrow \mathscr{U}_{k-1}$
        $\mathscr{U}_k(equipment) \leftarrow$ False
        foreach $edge \in E$ do
            $s \leftarrow$ evaluateConstraint($\Omega(EM(edge))$)
            if $s \neq EOS_{k-1}$.state then $EOS_k(edge) \leftarrow (s, \mathscr{T}(e_k))$
            else $EOS_k(edge) \leftarrow EOS_{k-1}(edge)$
        end
        $\mathscr{H}_k \leftarrow \mathscr{H}_{k-1}$
        foreach $h \in \mathscr{H}_k$ do
            $h \leftarrow$ clearExpected($h$)
            foreach $c \in$ getConsistent($h$) do
                foreach $child \in$ getReachableChildren($c$) do
                    if $child \notin$ getConsistent($h$) $\cup$ getMissed($h$) then
                        $h \leftarrow$ addToExpected($child, ND(c, child)$)
                        $T \leftarrow T \cup$ Timeout($h, c, child, ET(c, child)$.tmax $+ \mathscr{T}(e_k)$)
                    end
                end
            end
        end
    end
end
```

Algorithm 13: Function: handleTimeout

```
Function handleTimeout
    Input: $e_k$, $\mathscr{H}_{k-1}$, $\mathscr{U}_{k-1}$, $OS_{k-1}$, $EOS_{k-1}$, $\mathscr{M}$, $\mathscr{G}$, $label$
    Output: $\mathscr{H}_k, \mathscr{U}_k, OS_k, EOS_k, T$
    begin
        $OS_k \leftarrow OS_{k-1}$,  $EOS_k \leftarrow EOS_{k-1}$,  $\mathscr{U}_k \leftarrow \mathscr{U}_{k-1}$,  $T \leftarrow \varnothing$,  $\mathscr{H}_k \leftarrow \mathscr{H}_{k-1}$
        $p \leftarrow$ getParentNode($label$)
        $d \leftarrow$ getExpectedDiscrepancyNode($label$)
        foreach $h \in \mathscr{H}_k$ do
            if $d \in$ getExpected($h$) then
                $h \leftarrow$ moveFromExpectedToMissed($h, d, ND(p, d)$)
            end
        end
    end
end
```

7.4 Evaluation

In this section we evaluate the proposed TCD based diagnosis system by 1) verifying the correctness of the different observer logics associated with breaker, distance and overload protection, 2) validating the response of TFPG reasoning algorithm with the help of multiple scenarios involving physical, detection and stuck faults.

7.4.1 Verification of Observer Logic

We begin the observer logic verification by identifying the different safety and liveness requirements based on the expected behavior of observers related to protection assembly components. These requirements are enumerated as follows

(**R1**) The change in the breaker state from close to open is always detected by the breaker observer.

(**R2**) A breaker stuck operation is always detected by the breaker observer.

(**R3**) In the absence of a missed detection fault, a distance relay observer should correctly infer the fault.

(**R4**) In the absence of a missed detection fault, an overload relay observer should correctly infers the overload conditions.

(**R5**) The distance relay observer fails to infer the physical fault when the missed detection fault is present.

(**R6**) The presence of spurious detection faults is incorrectly inferred by the distance relay observer as physical fault.

(**R7**) The overload relay observer fails to infer the physical fault when the missed detection fault is present.

(**R8**) The presence of spurious detection faults is incorrectly inferred by the overload relay observer as physical fault.

The above mentioned requirements are evaluated in context of an arbitrary TCD model, similar to the one described in the section 6.4.1, but with three additional timed automata templates related to breaker, distance and overload protection observers labeled as `brObs`, `drObs` and `orObs` respectively. The verified TCTL properties that satisfy these requirements are listed as follows.

(P1) `cyber3.S2 and cyber3.inject_time > 0 and cyber3.inject_time < f_physical_time --> hyp_stuck_close_var`

(P2) `sc_close_to_open_var --> hyp_sc_close_to_open_var`

(P3) `((f_physical_vars[0] and not cyber2.injected) or (f_physical_vars[0] and cyber2.injected and cyber2.inject_time - f_physical_time - f_z1_d1.t_max> 0)) --> hyp_zone_act_vars[0]`

(P4) `((f_physical_vars[1] and not cyber2.injected) or (f_physical_vars[1] and cyber2.injected and cyber2.inject_time - f_physical_time - f_z2_d2.t_max> 0)) --> hyp_zone_act_vars[1]`

(P5) `((f_physical_vars[2] and not cyber2.injected) or (f_physical_vars[2] and cyber2.injected and cyber2.inject_time - f_physical_time - f_z3_d3.t_max> 0)) --> hyp_zone_act_vars[2]`

(P6) `((f_physical_vars[3] and not cyber2.injected) or (f_physical_vars[3] and cyber2.injected and cyber2.inject_time - f_physical_time - f_o1_d4.t_max> 0)) --> hyp_ovr_act_var`

(P7) `(f_physical_vars[0] and cyber2.injected and cyber2.inject_time > 0 and cyber2.inject_time < f_physical_time + f_z1_d1.t_min ) --> not hyp_zone_act_vars[0]`

(P8) `(f_physical_vars[1] and cyber2.injected and cyber2.inject_time > 0 and cyber2.inject_time < f_physical_time + f_z2_d2.t_min ) --> not hyp_zone_act_vars[1]`

(P9)
```
(f_physical_vars[2] and cyber2.injected and cyber2.inject_time
> 0 and cyber2.inject_time < f_physical_time + f_z3_d3.t_min )
--> not hyp_zone_act_vars[2]
```

(P10)
```
(f_physical_vars[3] and cyber2.injected and cyber2.inject_time
> 0 and cyber2.inject_time < f_physical_time + f_o1_d4.t_min )
--> not hyp_ovr_act_var
```

(P11)
```
(f_sp_vars[0] and f_sp_vars[1] and f_sp_vars[2] and
cyber1.inject_time > 0 and cyber1.inject_time < f_physical_time
+ f_z1_d1.t_min) --> hyp_zone_act_vars[0]
```

(P12)
```
(not f_sp_vars[0] and f_sp_vars[1] and f_sp_vars[2] and
cyber1.inject_time > 0 and cyber1.inject_time < f_physical_time
+ f_z2_d2.t_min ) --> hyp_zone_act_vars[1]
```

(P13)
```
(not f_sp_vars[0] and not f_sp_vars[1] and f_sp_vars[2] and
cyber1.inject_time > 0 and cyber1.inject_time < f_physical_time
+ f_z3_d3.t_min ) --> hyp_zone_act_vars[2]
```

7.4.2 Validation of TFPG Reasoner

We utilized multiple event traces of WSCC 9 Bus system (see Figure 6.17) generated from a Simulink Simscape [151] model to validate the TFPG reasoning algorithm. The simulations are performed using a fixed step discrete solver with a step size of 1 ms in phasor simulation mode. The events from the simulation model are serialized and then processed by TCD diagnosis system in an offline setting. We simulated four scenarios that are described are described as follows:

7.4.2.1 Physical Fault

Event Trace: In this scenario, a 3 phase to ground fault is injected in the line between buses 4 and 5, TL_4_5 at 10.00 secs. After 0.008 secs (relay frequency), various zone elements in the primary and secondary protection assemblies of TL_4_5 detect the reduction in impedance and produce event summaries. The zone 1, zone 2 and zone 3 elements of protection assembly $TL\_4\_5\_PA\_4$ (situated between line TL_4_5 and bus 4) emit $Z1\_TL\_4\_5\_PA\_4$, $Z2\_TL\_4\_5\_PA\_4$ and $Z3\_TL\_4\_5\_PA\_4$ events. The zone 2 and zone 3 elements of $TL\_4\_5\_PA\_5$ produce $Z2\_TL\_4\_5\_PA\_5$ and $Z3\_TL\_4\_5\_PA\_5$ events. The back-up protection assemblies, $TL\_9\_4\_PA\_9$ and $TL\_5\_6\_PA\_6$, produce the following sets of events, ($Z2\_TL\_9\_4\_PA\_9$, $Z3\_TL\_9\_4\_PA\_9$), ($Z3\_TL\_5\_6\_PA\_6$) respectively. Since the zone 1 elements of both primary protection assemblies have detected the fault, instruction to open the breaker is also sent, as denoted by a pair of events, $cmd\_open\_TL\_4\_5\_PA\_4$ and $cmd\_open\_TL\_4\_5\_PA\_5$. Breaker on both ends respond to the command and change their physical state at 10.058 sec while generating events, ($act\_sc\_open\_TL\_4\_5\_PA\_4$, $act\_sc\_open\_TL\_4\_5\_PA\_5$) to signal the change in actuator state to open. As a result of state change, the zone elements in the backup protection relays are no longer able to detect fault conditions and emit de-activation event, $Z2'\_TL\_9\_4\_PA\_9$, $Z3'\_TL\_9\_4\_PA\_9$, $Z3'\_TL\_5\_6\_PA\_6$ at 10.066 secs.

Diagnosis Results: The observer associated with the distance protection in $TL\_4\_5\_PA\_4$ processes the events $Z1\_TL\_4\_5\_PA\_4$, $Z2\_TL\_4\_5\_PA\_4$ and $Z3\_TL\_4\_5\_PA\_4$ in that order. It transitions from the state S5 to S8 via S7 and produces a derived alarm $hyp\_d\_Z1\_TL\_4\_5\_PA\_4$ indicating the activation of TFPG discrepancy node, $D\_Z1\_TL\_4\_5\_PA\_4$, at t=10.008 secs. Similarly the observer associated with the other primary protection assembly, $TL\_4\_5\_PA\_5$ absorbs the respective zone 1, zone 2 and zone 3 alarms and goes through the same state transition leading to final state S8 and generating $hyp\_d\_Z1\_TL\_4\_5\_PA\_5$. The observer related to the $TL\_9\_4\_PA\_9$ processes events, $Z2\_TL\_9\_4\_PA\_9$ and $Z3\_TL\_9\_4\_PA\_9$ by transitioning to S4 via S3. This observer waits for an alarm from zone element of $TL\_9\_4\_PA\_9$ for 4 milliseconds before generating a derived alarm. Similarly, the other back-up protection relay $TL\_5\_6\_PA\_6$, transition to S2 after receiving $Z3\_TL\_5\_6\_PA\_6$ and wait for either zone 1 or zone 2 alarm for the same amount of time.

At this instant the TFPG reasoner has two derived to alarms, $hyp\_d\_Z1\_TL\_4\_5\_PA\_4$, $hyp\_d\_Z1\_TL\_4\_5\_PA\_5$ related to discrepancy activation. As a result, the TFPG reasoner creates 3 hypotheses with ids: H4, H5 and H6. The hypothesis, H6 is a special *null* hypothesis, which marks every received alarm as spurious and lists spurious detection faults in all zone elements of $TL\_4\_5\_PA\_4$ and $TL\_4\_5\_PA\_5$, i.e., $F\_z1\_sp\_TL\_4\_5\_PA\_4$, $F\_z2\_sp\_TL\_4\_5\_PA\_4$, $F\_z3\_sp\_TL\_4\_5\_PA\_4$, $F\_z1\_sp\_TL\_4\_5\_PA\_5$, $F\_z2\_sp\_TL\_4\_5\_PA\_5$, $F\_z3\_sp\_TL\_4\_5\_PA\_5$. The next hypothesis, H5, correctly identifies the root cause of the system aberration and lists $F2\_TL\_4\_5$ as fault source. The consistent discrepancy set of this hypothesis includes $D\_Z1\_TL\_4\_5\_PA\_4$ and $D\_Z1\_TL\_4\_5\_PA\_5$, the hypothesis H5 requires the activation of one discrepancy from each pair ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$) and lists them in expected set. Lastly, the hypothesis H4, lists the adjacent line segment, represented by $F1\_TL\_4\_5$ as the root cause and requires the activation of $D\_Z2\_TL\_4\_5\_PA\_5$ from the other primary protection instead of $D\_Z1\_TL\_4\_5\_PA\_5$. Since this hypothesis is not able to explain $D\_Z1\_TL\_4\_5\_PA\_5$, it marks it as inconsistent and adds the following spurious detection faults, $F\_z1\_sp\_TL\_4\_5\_PA\_5$, $F\_z2\_sp\_TL\_4\_5\_PA\_5$, $F\_z3\_sp\_TL\_4\_5\_PA\_5$ to the fault set. It also lists activation of one discrepancy from each pair ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$) in the expected set.

At time t=10.012 the wait period of the observers associated with distance protection in $TL\_9\_4\_PA\_9$ and $TL\_5\_6\_PA\_6$ expires, resulting in the state transition of respective observers to S11 and S9 respectively with the generation of two derived alarms $hyp\_d\_Z2\_TL\_9\_4\_PA\_9$ and $hyp\_d\_Z3\_TL\_5\_6\_PA\_6$. The number of spurious detection faults in H6 increases to 9. Both hypotheses H5 and H6, remove the two discrepancies, $D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_5\_6\_PA\_6$ from the expected set and add to the consistent set. At time t=10.04, the wait time for the expected discrepancy, $D\_Z2\_TL\_4\_5\_PA\_5$ expires and is moved to the missed set. Since the edge between this discrepancy and the fault node is certain, a missed detection fault, $F\_miss\_TL\_4\_5\_PA\_5$ is added to the fault set of H4. The number of fault nodes in the hypothesis H4, H5 and H6 are 5, 1, and 9 respectively. According to law of parsimony, the hypothesis H5 is ranked first followed by H4 and H6.

7.4.2.2 Physical and Missed Detection Fault

Event Trace: In this scenario, addition to physical fault at 10.00 secs, a missed detection fault in protection assembly, $TL\_4\_5\_PA\_5$ is injected at t = 0 secs. As a result of the missed detection, none of the associated zone elements are able detect the fault conditions. Thus, the list of events at t=10.008 secs is limited to $Z1\_TL\_4\_5\_PA\_4$, $Z2\_TL\_4\_5\_PA\_4$, $Z3\_TL\_4\_5\_PA\_4$, $Z2\_TL\_9\_4\_PA\_9$, $Z3\_TL\_9\_4\_PA\_9$, $Z3\_TL\_5\_6\_PA\_6$, and $cmd\_open\_TL\_4\_5\_PA\_4$. Similar to previous scenario, the breaker in the assembly $TL\_4\_5\_PA\_4$ changes its state to open at 10.058 secs, forcing the back up protection assembly, $TL\_9\_4\_PA\_9$ to generate discrepancy de-activation events $Z2'\_TL\_9\_4\_PA\_9$, $Z3'\_TL\_9\_4\_PA\_9$ at 10.066. At t = 11.008 the wait time of the zone 3 element in $TL\_5\_6\_PA\_6$ expires and command to open the breaker, $cmd\_open\_TL\_5\_6\_PA\_6$ is produced. The corresponding breaker responds to this event by changing the state at 11.058 secs and generating $act\_sc\_open\_TL\_5\_6\_PA\_6$ event.

Diagnosis Results: The observer associated with the distance protection in $TL\_4\_5\_PA\_4$ processes the events $Z1\_TL\_4\_5\_PA\_4$, $Z2\_TL\_4\_5\_PA\_4$ and $Z3\_TL\_4\_5\_PA\_4$ and transitions from the state S5 to S8 while producing a derived alarm $hyp\_d\_Z1\_TL\_4\_5\_PA\_4$, indicating the activation of TFPG discrepancy node $D\_Z1\_TL\_4\_5\_PA\_4$ at t=10.008 secs. The observer related to the $TL\_9\_4\_PA\_9$ processes events, $Z2\_TL\_9\_4\_PA\_9$ and $Z3\_TL\_9\_4\_PA\_9$ by transitioning to S4 via S3. This observer waits for an alarm from zone element of $TL\_9\_4\_PA\_9$ for 4 milliseconds before generating a derived alarm. Similarly the other back up protection relay $TL\_5\_6\_PA\_6$, transitions to S2 after receiving $Z3\_TL\_5\_6\_PA\_6$ and waits for either zone 1 or zone 2 alarm for the same amount of time.

At this instant, the TFPG reasoner has a single derived alarm, $hyp\_d\_Z1\_TL\_4\_5\_PA\_4$ to process. As a result, the TFPG reasoner creates 3 hypotheses with ids: H4, H5 and H6. The hypothesis, H6 is the *null* hypothesis, that lists spurious detection faults in all zone elements of $TL\_4\_5\_PA\_4$ i.e., $F\_z1\_sp\_TL\_4\_5\_PA\_4$, $F\_z2\_sp\_TL\_4\_5\_PA\_4$, $F\_z3\_sp\_TL\_4\_5\_PA\_4$. The next hypothesis, H5, correctly identifies the root cause of the system abnormality and lists $F2\_TL\_4\_5$ as a fault source. The consistent discrepancy set of this hypothesis includes $D\_Z1\_TL\_4\_5\_PA\_4$ and the expected set requires the activation of $D\_Z1\_TL\_4\_5\_PA\_4$ and one discrepancy from each pair ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$). Lastly, the hypothesis H4, lists the adjacent line segment, represented by $F1\_TL\_4\_5$ as the root cause and requires the activation of $D\_Z2\_TL\_4\_5\_PA\_5$ from the other primary protection assembly. It also lists activation of one discrepancy from each pair ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$) in the expected set.

At time t=10.012 the wait period of the observers associated with distance protection in $TL\_9\_4\_PA\_9$, $TL\_5\_6\_PA\_6$ expires, resulting in the transition to S11 and S9 respectively with the generation of two derived alarms *hyp_d_Z2_TL_9_4_PA_9* and *hyp_d_Z3_TL_5_6_PA_6*. The number of spurious detection faults in H6 increases to 6. Both hypotheses, H5 and H6, remove the two discrepancies, *D_Z2_TL_9_4_PA_9*, *D_Z3_TL_5_6_PA_6* from their expected sets and add to the consistent set. At time t=10.04, the wait time for the discrepancies, (*D_Z2_TL_4_5_PA_5*, *D_Z1_TL_4_5_PA_5*) expires and are moved to the missed set in H4 and H5 respectively. Since the edge between these discrepancy and the fault nodes are certain, missed detection faults *F_miss_TL_4_5_PA_5*, *F_miss_TL_4_5_PA_4*, are also added to the fault set of H4 and H5 respectively. The number of faults listed in H4 and H5 are same i.e. 2 and are ranked first followed by H6.

7.4.2.3 Physical and Breaker Fault

Event Trace: In this scenario, a stuck close fault in the breaker of protection assembly, *TL_4_5_PA_5* is injected at t=0 and physical fault is injected in the line *TL_4_5* at 10.00 secs. The events produced in this scenario at t=10.008 is similar to first scenario. However at t = 10.058 only one breaker (associated wih *TL_4_5_PA_4*) changes its state to open and *act_sc_open_TL_4_5_PA_4* is generated. At 10.066 secs, the back up protection assembly, *TL_9_4_PA_9* generates discrepancy de-activation events, *Z2′_TL_9_4_PA_9* and *Z3′_TL_9_4_PA_9*. Similar to previous scenario, at t = 11.008 the wait time of the zone 3 element in *TL_5_6_PA_6* expires and command to open the breaker, *cmd_open_TL_5_6_PA_6* is produced which leads to change in breaker state at 11.058 secs and generation of *act_sc_open_TL_5_6_PA_6* event.

Diagnosis Results: The diagnosis results in this scenario are similar to the first scenario till t=10.058 secs. However, at t = 10.062, the wait time for the observer associated with breaker in *TL_4_5_PA_5* expires and it transitions back to S1 while generating a derived alarm related to fault node activation, *hyp_f_sc_TL_4_5_PA_5*. The reasoner updates all three hypotheses and add the corresponding fault, *F_sc_TL_4_5_PA_5* in the fault set. The final count of faults in H4, H5 and H7 is 6, 2 and 10 respectively.

7.4.2.4 Spurious Detection Fault

Event Trace: In this scenario, a spurious detection fault is injected in the zone 1 element of $TL\_4\_5\_PA\_5$ at t = 10.000 secs. As a result of the fault, the protection assembly generates $Z1\_TL\_4\_5\_PA\_5$ event. An open command is issued at t = 10.008 secs, resulting in the generation of event, $cmd\_open\_TL\_4\_5\_PA\_5$, which leads to the change in breaker state at t = 10.058 and production of event $act\_sc\_open\_TL\_4\_5\_PA\_5$.

Diagnosis Results: The observer associated with the protection assembly, $TL\_4\_5\_PA\_5$ processes the event $Z1\_TL\_4\_5\_PA\_5$ and jumps to `S5` at 10.008. It waits for an alarm from other zone elements for 4 milliseconds. At time t=10.012, the wait period of the observer expires, resulting in the state transition of the observer to `S13` with the generation of a derived alarms $hyp\_d\_Z1\_TL\_4\_5\_PA\_5$. At this instant, the TFPG reasoner has a single derived alarm, $hyp\_d\_Z1\_TL\_4\_5\_PA\_5$ to process. As a result, the TFPG reasoner creates 3 hypotheses with ids: `H4`, `H5` and `H6`. The hypothesis, `H6` is the *null* hypothesis, which marks the received alarm as spurious i.e., $F\_z1\_sp\_TL\_4\_5\_PA\_5$ is added to fault set. Please note that the reasoner has created spurious detection fault in only one zone element as none of the other elements had generated alarms. This additional information is relayed to the reasoner by the observers. The next hypothesis, `H5`, lists the root cause of the system aberration and records $F2\_TL\_4\_5$ as fault source. The consistent discrepancy set of this hypothesis includes $D\_Z1\_TL\_4\_5\_PA\_4$ and the expected set includes the activation of $D\_Z1\_TL\_4\_5\_PA\_4$ and one discrepancy from each pair ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$). Lastly, the hypothesis `H4`, lists the adjacent line segment, represented by $F3\_TL\_4\_5$ as the root cause and requires the activation of $D\_Z2\_TL\_4\_5\_PA\_4$. It also lists activation of one discrepancy from each pair, ($D\_Z2\_TL\_9\_4\_PA\_9$, $D\_Z3\_TL\_9\_4\_PA\_9$), ($D\_Z2\_TL\_5\_6\_PA\_6$, $D\_Z3\_TL\_5\_6\_PA\_6$) in the expected set. At t= 10.040 the wait time for the discrepancies in the expected set of `H4` and `H5` expires and all the discrepancies from expected set are moved to missed set. The missed detection faults related to primary protection relay $TL\_4\_5\_PA\_4$, i,e, $Fmiss\_TL\_4\_5\_PA\_4$ is added to fault sets of both hypothesis. However, the missed detection faults related to back up protection assemblies,$TL\_5\_6\_PA\_6$ and $TL\_9\_4\_PA\_9$ are not added as the propagation edges between their respective discrepancies and the current fault nodes are uncertain.

The final count of fault nodes in three hypotheses, H4, H5 and H6 is 2, 2 and 1 respectively.

The TCD based diagnosis system is able to correctly diagnose faults in the above mentioned scenarios. Apart from these four cases, we synthetically generated 150 similar scenarios by forward traversal of TCD fault model of IEEE 14, 39, 57 , 117 and 300 bus networks [152]. In all those cases, the TCD diagnoser is able to correctly diagnose the faults as well. The event traces and the respective diagnosis system logs can be downloaded from the github repository [140].

7.4.3 Timing and Scalability Analysis

The TFPG reasoner starts with a set of single fault hypotheses i.e. it creates a hypothesis for every fault that can explain the initial alarm. We selected this methodology based on the underlying system assumption that all faults are temporally separated by at least 1 second as described in chapter 6. For single fault hypothesis approach, the upper bound on the size of hypothesis set initially is equal to n + 1, where n is the number of fault nodes (physical) in the system TCD model. However, in our use case, the number of initial hypothesis can be inferred from the type of the discrepancy node, i.e. for $D1^{PAk}$ type discrepancies, the number of initial hypothesis generated is always 3 while for $D2^{PAk}$ and $D3^{PAk}$ type discrepancies, the number of hypotheses produced is $3(\alpha+1)$, where α is the number of transmission lines for which zone 2 and 3 elements of PAk provides secondary or backup protection. The size of the hypothesis set remains stable till an alarm is received which cannot be explained by any of thesis hypotheses in the existing hypothesis set. As a result, the new hypothesis set is composed of the cartesian product of the exiting hypotheses with the recently created hypotheses that explains the latest alarm leading to exponential growth of hypothesis set. Figure 7.5 shows the evolution of the size of hypothesis set when physical fault is injected in 5 different transmission lines in IEEE 57 bus system. To manage the exponential growth, we weed out those hypothesis from the set that list more than 5 cyber faults in protection devices as the probability of such an event is extremely low [153]. This results in a manageable hypothesis set with polynomial growth as shown in figure 7.5.

Figure 7.6 shows the average time required to process different kinds of events per hypothesis when the size of underlying power network is increased. The response time for all events except mode change remains constant. As highlighted in algorithm 12, the processing of mode change

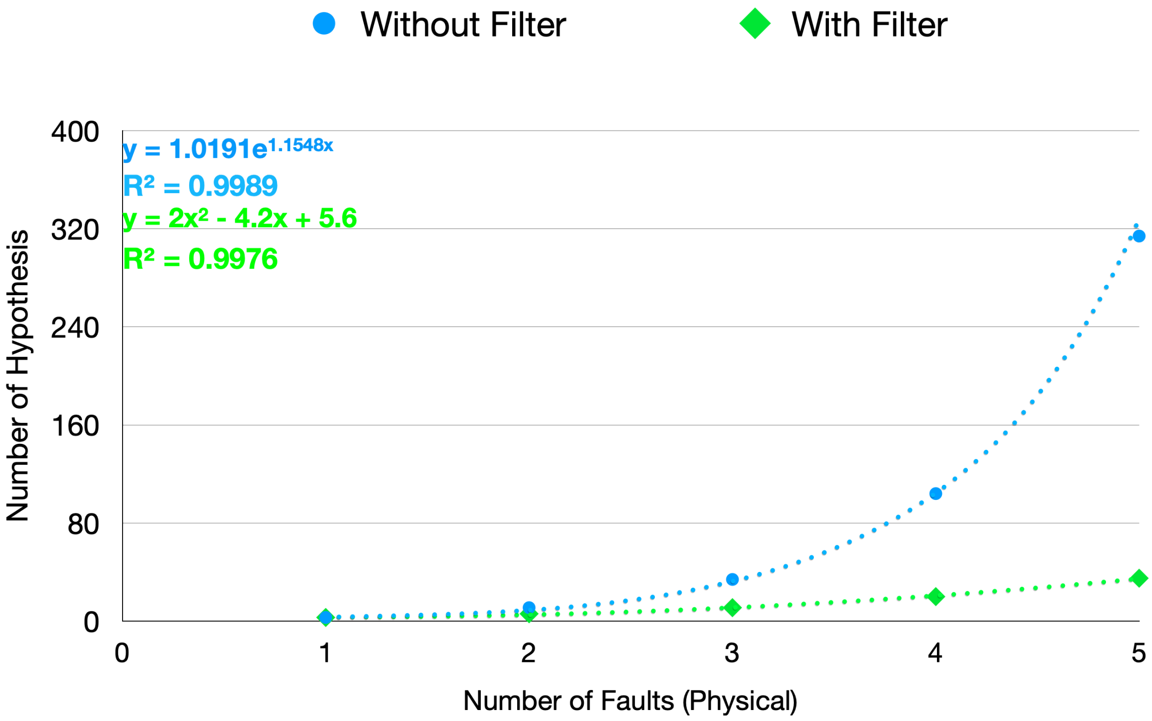


Figure 7.5: Scalability analysis: Size of hypothesis set w.r.t number of faults

event requires changing the state of all fault edges which increases quadratically with increase in network size, the resulting growth is also polynomial in nature.

7.5 Summary

In this chapter, we presented a hierarchical fault diagnosis system based on TCD fault model. The diagnosis system consists of multiple observers which track the behavior of protection system components and produce hypothesis (derived alarms) based on local information. The diagnosis system also consist of a centralized TFPG reasoner that creates a system level integrated hypothesis based on derived alarms received from various observers. We described and verified the observer logic of breaker, distance and overload relays using UPPAAL model checker. We also discussed in detail the TFPG reasoning algorithm which is able to handle out of order events. In the end, we validated the reasoning algorithm with the help of several scenarios involving multiple faults.

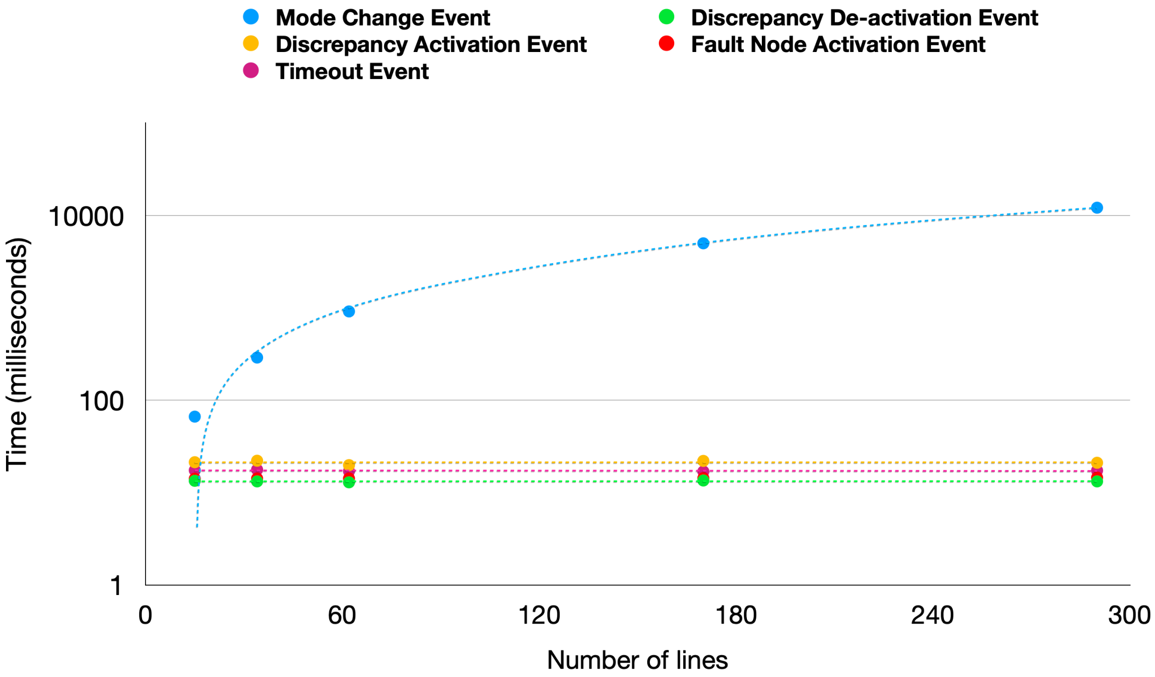


Figure 7.6: Timing analysis: Event processing time w.r.t size of the network

7.6 Contributions

The hierarchical TCD based diagnosis system was introduced in the publication titled, *Hierarchical Reasoning about Faults in Cyber-Physical Energy Systems using Temporal Causal Diagrams* [154]. A follow-up paper titled, *Distributed diagnosis of faults in safety critical systems using qualitative models* [To Be Submitted] extends the previous work to include out of order event processing and complexity analysis. All UPPAAL TA templates of observer logic and complete python based implementation of the diagnosis engine along with a trace simulator are available for download from the Github repository [140].

Chapter 8

Applications: Fault Prognostics and Cascade Mitigation

8.1 Introduction

Power system equipment is constantly exposed to dynamic environments caused due to changing loading conditions, physical degrading of the components and external faults such as earthing and winding faults. The safety of the system is ensured by a large infrastructure of *protection system assemblies*. A protection system assembly is composed of instrument transformers, intelligent software enabled protection relays and high-voltage circuit breakers. Relays sample the scaled down voltage and current signals from instrument transformers and based on embedded relay logic ascertain the presence of a fault. On detecting the presence of faulty conditions, the relay sends a tripping signal to the breaker which isolates the faulty component from the system. However, due to a lack of system wide perspective and hidden faults (incorrect settings), the actions of protection devices have been known to cause cascading outages. A cascading outage is defined as an uncontrolled loss of any system facilities or load as a result of fault isolation. Such cascading outages in power grids successively weaken the system by increasing stress on other components and can lead to complete blackouts.

There are two stages associated with cascading outages. During the first stage, system operators evaluate system conditions against different grid stability criteria to identify state trajectories and take some control actions to improve the operating conditions and prevent the possible cascading outages. The control cost is minimal compared with the massive cost of cascading outages. The current industry practice involves performing on demand analysis of cascades using different simulation models. However, these simulations take a considerable amount of time to finish and data generated by these complex simulation models is difficult to analyze in a timely manner. This increases the line operators' response time to anticipate the future state of the system. Thus, an efficient surrogate model is required that can quickly classify the stability of the system and provides cascade progression in case the system is deemed unstable.

8.2 Related Research

8.2.1 Prognostics Methodologies

Current industry practice is to determine (offline) critical components of the power system such that their outages can successively weaken the system leading to blackouts. The process of finding these critical component outages is called *contingency analysis*. N-1 contingency analysis refers to a single component outage out of total N system components. The cascades in the past have been caused due to the interaction of more than 1 independent component outage. Hence, it is required to perform high-order (N-k) contingency analysis where k is the number of initial outages in a system with N components. It is well understood that calculating higher order contingencies are infeasible as the total number of combinations grows exponentially.

A number of methodologies exist in the literature that tries to identify contingencies in a power network. We categorize these techniques into the following two categories:

8.2.1.1 Topology Based

There is a substantial amount of research being done into understanding cascading phenomenon using *topological contagion* models [155]. These threshold based contagion models have been deemed useful in comprehending problems like disease spreading [156] and social influence spreading [157]. Similar approaches have been proposed for finding contingencies in power systems [158]. Contagion models are based on the assumption that a component outage affects only nearby components. However, the premise of cascade progression being a local phenomenon does not hold well in power systems.

8.2.1.2 Simulation Based

The second approach uses simulation models of power grids to understand the cascade propagation and identify critical component outages that can severely impact the power system. This approach is used by line operators in offline and online settings. Examples of such simulation models are DCSIMSEP [159], Oak Ridge-PSERC-Alaska (OPA) [160], Manchester model [161], TRELSS [162] and COSMIC [163]. However, these simulations take a considerable amount of time to finish. Moreover, the generated data from these simulations is complicated and thus difficult to understand and summarize quickly.

8.2.2 Mitigation Strategies

Once the cascade conditions have been identified, pre-defined actions such as load shedding can be used to suppress the cascading effects of overloads, voltage and frequency instabilities. An alternative approach is to curtail a percentage of the load instead. Curtailment provides an effective means of handling the cascade effects without disconnecting the complete load. For example, the cascading failures during the blackout of Aug 2003 in the USA could have been avoided by removing a relatively small amount of load in the Cleveland area [164, 165]. However, the most effective load curtailment is not always obvious. Line operators rely on optimal power flow algorithm to identify the suitable generator or load re-dispatch actions. General practice is to use simple linear programming to find minimalistic load shedding actions that can prevent the progression of a cascade. But the linear approximation of the underlying system can be misleading and can result in incorrect load management. A number of approaches based on model predictive control [164, 166, 167] have been proposed that tackles problems of voltage collapse and successive branch outages due to overloads. However, above mentioned model predictive control strategies are not always guaranteed to provide an optimal solution because of the limitation of the underlying approximation of the mathematical model and limited number of control actions as per the control horizon.

In this chapter we present a systematic approach of finding load curtailment actions in an offline setting. However, this methodology covers 1) identifying critical state of the system 2) encoding the blackout causing states as a transition relation using binary decision diagrams and 3) calculating mitigating actions at run-time.

8.3 Solution Approach

Our approach utilizes reduced order binary decision diagrams [168] (BDDs) to encode different blackout causing outages (contingencies). The advantage of using BDDs is their ability to encode complex behaviors that can be used in reasoning about cascade progression efficiently while in-

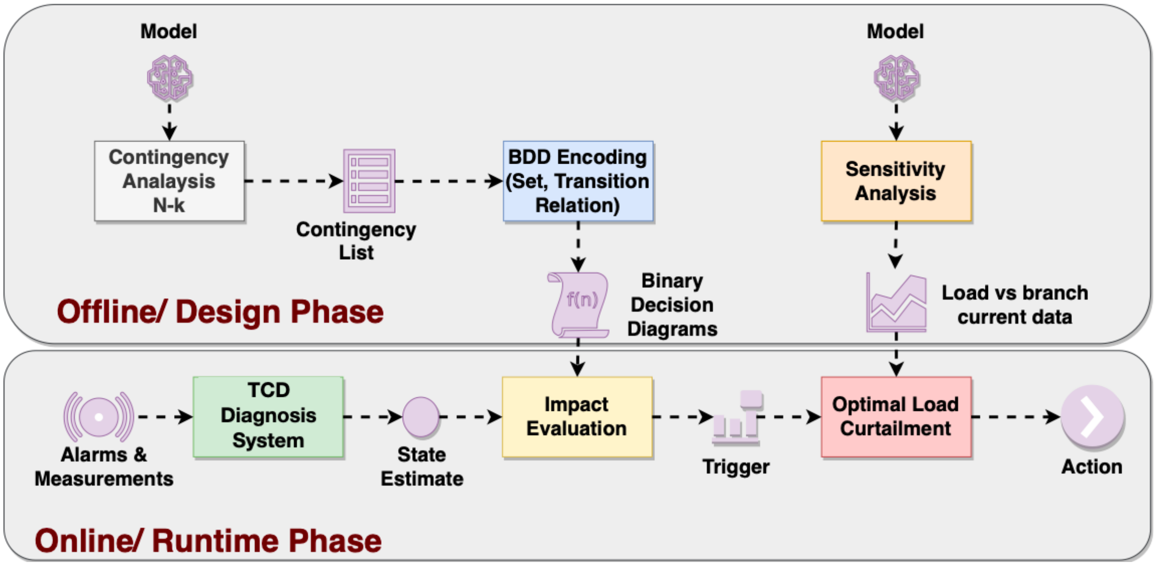


Figure 8.1: Fault Prognostics and Cascade Mitigation Workflow

curring small memory footprint and at the same time allowing fast access time. The proposed cascade prognosis methodology consists of two phases 1) *Offline* and 2) *Online* as described in Figure 8.1. Initially, in the offline phase, critical outages or contingencies are identified followed by storing these contingencies and their respective progression in BDDs. Whereas in online phase, actual prognosis is done using BDDs created in the previous stage and load curtailment actions are calculated based on the current state of the system.

Cascading outages in power system are primarily caused by production and demand imbalance. The initiating event can be generator (source), transmission line or transformer (branch) outage caused by fault isolation or planned maintenance. The initial outages can increase stress in the rest of the system causing secondary effects in terms of branch overloads, bus voltage fluctuations and frequency instability. These secondary effects can lead to more outages by the action of protection devices which can further destabilize the system leading to blackouts. If the secondary effects can be removed by curtailing a part of the load, then cascading outages can be prevented. We formulate the load curtailment as a non-linear optimization problem and utilize OpenMDAO [169] that uses external steady state simulator, OpenDSS [170] to find optimal control actions. The optimization framework, OpenMDAO acts as an orchestrator for finding voltage and current gradients by triggering OpenDSS to solve the power flow equations at different values of load demands and generator power injections. Our approach is different from the existing approaches as it does not assume linear power flow model. The online and offline phases are discussed in more detail in the following subsections.

8.3.1 Identification of Contingencies

We developed a simple cascade simulation model (based on steady state calculations) that successively solves the power flow (using OpenDSS) by removing the overloaded branches from the system after the initial component outages. The simulation keeps on tripping the overloaded branches till a blackout situation is reached or there are no more secondary effects (overloads) in the system. This cascade simulation model caters to slowly progressing cascades that eventually lead to blackouts involving overloads. We have adopted a conservative approach where all the secondary effects of initial outages are mitigated through the existing (pre-defined) protection schemes that isolate the overloaded components from the system.

Algorithm 14: Algorithm for finding critical N-k contingencies

```
Input: Model, k, Branch
Output: T, TR
A ← choose(Branch, k)                         ▷ Generating contingency list
j ← 1
for $j \leq \binom{|Branch|}{k}$ do
    Prev ← A[j], Next ← ∅, Temp ← ∅, Start ← A[j]
    Model.apply_contingency(Prev)             ▷ Applying $j^{th}$ contingency
    while True do
        if Model.check_blackout() then
            T ← T ∪ Start                     ▷ Save contingency
            TR ← TR ∪ Temp                    ▷ Save the sequence of branch outages
            break
        else
            Next ← Model.get_overloads()      ▷ Identify overloaded branches
            if Next ≠ ∅ then
                Temp ← Temp ∪ (Next ∪ Prev, Prev)
                Prev ← Temp
                Model.trip_branches(Next)     ▷ Tripping overloaded branches
            else
                break
            end if
        end if
    end while
    j ← j + 1                                 ▷ Iterate to next contingency
end for
```

Listing 14 shows the underlying algorithm to find N-k contingencies. The input parameter of the algorithm includes a OpenDSS model (*Model*), an integer representing the order of contingencies (k) and a set of all branch labels (*Branch*). The output of the algorithm consists of two sets T, TR that represents a collection of initiating events and their respective progressions. The set, $T = \{s_1, s_2, ..., s_n\}$ is a collection of all contingencies that can cause blackout, where s_i is some combination of branch outages. The set, $TR_{s_1} = \{(s_1, s_2), (s_2, s_3), ...(s_i, s_j)\}$ represents the progression of cascade caused due to s_1, where s_i represents the initial branch outages and s_j implies the branch outages as a consequence s_i. The algorithm starts with tripping k lines at random and solving the power flow to update the branch currents and bus voltages. The second step is to check for the blackout criteria. The blackout criteria is configurable in terms of the percentage of the original load (demand) that is not operational. For a blackout criteria of 40%, if more than 40% of the net system load demand cannot be satisfied in a given state, then the system is considered to have reached blackout. If the system is not in a blackout state, then secondary effects of the branch outages are investigated by checking the overloads in rest of the system. If no overloads are found then, the system is considered to have reached a safe state from where it cannot reach blackout. On the other hand, if some secondary overloads are present, the transition relation, represented by $Temp$ is updated followed by tripping all those branches. After branch tripping, the blackout criteria is checked again and the process repeats until a blackout state is reached or the system reaches a stable state (no overloads).

8.3.2 Efficient Storage Mechanisms

We employ compact and efficient data structure, ordered Binary Decision Diagrams (BDDs) to store the progressions of cascading outages. A binary decision diagram is a data structure that is used to represent boolean functions. A BDD is a directed acyclic graph that consists of two types of nodes, A) *Decision Nodes*: Each decision node represents a boolean variable, V_i, and has two child nodes, *high* and *low*. The edge from node, V_i to a low (or high) child represents an assignment of V_i to 0 (1). B) *Terminal Nodes*: There are two types of terminal nodes called 0-terminal and 1-terminal. A path from the root node to the 1-terminal (0-terminal) represents a variable assignment for which the represented Boolean function is true (false). A reduced ordered

BDD has a fixed ordering i.e different variables exists in the same order along different paths and has an important *canonicity* property i.e. for a fixed variable ordering, each boolean function has a unique representation. On an abstract level, these BDDs are used as a compressed representation of sets and transition relations that has relatively small memory footprint and allow fast retrieval of the encoded information as operations are performed directly on the compressed form. The progression of a cascade is represented by state transitions, where state defined by the component outages. Based on the initiating outages identified using algorithm 14 two BDDs are created with the following functionalities :

- First BDD labeled as, B_T, stores the set of initiating events that will cause cascading outages in the rest of the system leading to a complete system blackout.
- The Second BDD, labeled as, B_{T_R} stores the progression of all the initiating events captured by B_T as a translation relation.

8.3.2.1 BDD encoding of collection of initiating events

Let T be the set of branch outages identified by the offline N-k contingency analysis. Each element in T represents a collection of initiating events i.e. independent branch outages that can trigger a cascading phenomenon leading to a blackout. Since the main objective is to encode these sets of line outages, every element of T can be represented by a unique boolean vector $(v_1, v_2, v_3, ..., v_n)$, each $v_i \in 0, 1$, of length equal to the number of branches in the system. Then $T \subseteq S$ can be represented by a characteristic function $f_T : \{0,1\}^n \rightarrow \{0,1\}$ which maps a particular evaluation of $(v_1, v_2, v_3, ..., v_n)$ to either 0 or 1, where S is the power set. For each $s \in S$, if the value mapped by f_T is 1 then $s \in T$ otherwise s in not the member of T.

We define a labeling function for S, $L(S) : S \rightarrow P(Branch)$, where $Branch$ is a set of branch outages, say $(tl_1, tl_2, tl_3, ..., tl_n)$ associated to an initiating event combination, $s \in S$. Hence s can be represented by a boolean vector $(v_1, v_2, v_3, ..., v_n)$ where v_i is 1 if $tl_i \notin L(s)$. Here $v_i = 1$ means the power is flowing through branch tl_i i.e. all the breakers associated to the branch are closed whereas the value 0 implies no power flow. As a BDD, the initiating event combination, $s \in S$ is represented by the boolean function, $l_1 \cdot l_2 \cdot l_3 ... \cdot l_n$ where l_i is tl_i if $tl_i \notin L(s)$ otherwise $\overline{tl_i}$. The set T can be represented by the boolean function f_T,

$(l_{11} \cdot l_{12} \cdot l_{13} ... \cdot l_{1n}) + (l_{21} \cdot l_{22} \cdot l_{23} ... \cdot l_{2n}) + ... + (l_{j1} \cdot l_{j2} \cdot l_{j3} ... \cdot l_{jn})$
where $(l_{k1} \cdot l_{k2} \cdot l_{k3} ... \cdot l_{kn})$ represents the initiating event set s_k.

8.3.2.2 BDD encoding of progression of cascading outage

The progression of cascade can be modeled by transition relation. A translation relation is a boolean function, $f_{T_R} : S \times S \to \{0,1\}$, which outputs 1 if there exists a transition between two given states otherwise 0. Similar to T, set of valid transitions, T_R can be viewed as subset of all possible transitions, i.e. $T_R \subseteq S \times S$. An element $t \in T_R$ implies a transition from state s to s' and is represented by a pair of boolean vectors $((v_1, v_2, v_3, ..., v_n), (v'_1, v'_2, v'_3, ..., v'_n))$ where v_i is 1 if $tl_i \notin L(s)$ and 0 otherwise; and similarly, v'_i is 1 if $tl_i \notin L(s')$. A single transition link can be represented by a boolean function $(l_1 \cdot l_2 \cdot l_3 ... \cdot l_n) \cdot (l'_1 \cdot l'_2 \cdot l'_3 ... \cdot l'_n)$ and the complete set T_R can be represented as disjunction of such formulas as shown in the case of initiating events.

8.3.3 Identifying Cascade Progression

After creating these BDDs, the next task is to evaluate the current system state s_t, represented by a boolean vector $(v_1, v_2, ..., v_n)$, where n is the number of branches, is a member of the set T. If $s_t \in T$, then the progression of s_t can be calculated by finding the set of reachable states from s_t under a given transition relation, f_{T_R}. The operation of finding the set of states reachable from a given state is called image computation and the process of calculating image iteratively till a fixed point is reached is a fundamental step in many state exploration algorithms. The algorithm 15 shows the algorithm in determining the cascade progression for a given state of the system.

Algorithm 15: Algorithm for determining the evolution of current state, S_0

```
Input: $S_0 = (v_1, v_2, ..., v_n)$ ; $B_T$ ; $B_{T_R}$
Output: $S_{reach}$
Initialize: $S_{reach} = \phi$
if Evaluate($B_T, S_0$) == True then
   i = 0
   while $S_i \neq \phi$ do
      $S_{reach} = S_{reach} \cup S_i$                ▷ Update reachable set
      $S_{i+1} = Image(B_{T_R}, S_i) \setminus S_{reach}$   ▷ Identify new reachable states
      i = i+1
   end while
end if
```

The algorithm 15, requires a set of line outages (S_0), the BDDs B_T, B_{T_R}. The output of the algorithm is a sequence of states reachable from S_0 if $S_0 \in T$. The Evaluate(B_T, S_0) function checks whether S_0 is a member of set T represented by B_T. IF yes, then recursively next states are found and added to the set S_{reach} until fixed point is reached when for a given state S_{i-1} no new next state is defined i.e S_i is empty.

8.3.4 Cascade Mitigation

Cascading outages can be mitigated by adjusting the load demand of the system such that secondary branch overloads disappear. However, the amount of load curtailment should be minimized as a large difference between the power supplied by the generators and load demand can increase instability and leading to system collapse. We formulate identification of load curtailment as an optimization problem, described in equations (1)-(6), where, $\mathbf{L}$ (ohms) is a vector of load demands of size M, $\Delta\mathbf{L}$ is a vector of decision variables, such that ΔL_i denotes the (ratio) curtailment of load $L_i \in \mathbf{L}$ by $\Delta L_i \cdot |L_i|$ ohms. $\mathbf{I}$ is the collection of branch currents in the system.

$$\min_{\Delta\mathbf{L}} \quad \sum_{i=0}^{M} w_i \cdot \Delta L_i \cdot |L_i| \tag{8.1}$$

$$0 \le |I_j| \le I_j^{Max}, \quad \forall I_j \in \mathbf{I} \tag{8.2}$$

$$0 \le \Delta L_i \le 1, \quad \forall L_i \in \mathbf{L} \tag{8.3}$$

$$\Phi_{L_i} = \Phi_{(1-\Delta L_i)\cdot L_i}, \quad \forall L_i \in \mathbf{L} \tag{8.4}$$

$$\sum_i (\Delta L_i) \cdot |L_i| \le L_{max}^{total}, \quad \forall L_i \in \mathbf{L} \tag{8.5}$$

$$\mathbf{I} = f(\Delta\mathbf{L}) \tag{8.6}$$

The objective function is the weighted sum of all load curtailments as shown in equation 8.1 where weights, w_i models the importance associated with a load. For instance, critical loads can be establishments of national or societal importance such as hospitals and government buildings having large w_i. The inequality constraint described in equation 8.2 ensures no branch overloads are present in the final solution, where I_j^{Max} defines an upper limit on the current that can flow through a branch. The inequality constraints 8.3 describes the extent to which individual loads can be changed. The equality constraint 8.4 ensures the power factor, Φ of all loads are maintained i.e real and reactive load is shed in equal proportions. The inequality constraint 8.5 describes the upper bound on the total load, L_{max}^{total} that can be shed from the system. In the current implementation, it is 20% of the total system load. The function, f in equation 8.6 models the load flow equations. It updates the branch currents ($\mathbf{I}$) according to the change in system loads.

The convergence of optimization algorithm depends upon the size of the input search space i.e. number of design variables and their initial estimates. In order to speed up the convergence our proposed optimization framework performs sensitivity analysis to find filter out the loads that do not affect a given branch overload and obtain initial estimates. The sensitivity analysis can be broken into 3 sequential steps described as follows:

8.3.4.1 Data point generation

In this step, the effect of a varying absolute value of load demand, L_i on all the branch currents, $\mathbf{I}$ is observed. The load (L_i) vs branch current (I_j) data points are stored for learning a regression model. We have used Full Factorial Design of Experiment (DoE) analysis that uniformly samples the input space i.e. range [0, L_i] for each load $L_i \in \mathbf{L}$. In our implementation, 100 data points are considered for each load.

8.3.4.2 Regression Analysis

This step involves finding equation parameters (slope and intercept) for branch current vs load change data points generated in the previous step. It is safe to assume linear relationship between branch currents and load demand since the power factor remains constant. The sensitive loads can be classified by observing the slope of the equation, $|I_j| = m|L_i| + C$, where $|I_j|$ are the absolute value of current flowing through j^{th} branch and $|L_i|$ is the i^{th} load (magnitude) ; and m, c are the equation parameters. For a given load, if the slope is positive for any branch current, then the load is considered to be sensitive.

Table 8.1: Timing analysis for IEEE 14 Bus system

Parameters	**IEEE 14 Bus System**
Variable count	20
B_T, B_{T_R} construction time (secs)	18.80
Average time for true positive cases (secs)	0.006
Average time for true negative cases (secs)	3.5×10^{-5}

8.3.4.3 Starting Point Estimation

In this step, we estimate the starting value for each decision variable, $\Delta L_i \in \Delta \mathbf{L}$ as described in the equations (7) and (8), where I_j is the current in j^{th} branch, (C^i_j, m^i_j) are parameters of the learned regression model that relates branch current I_j and load L_i. S_i is the set of branches (indices) for which the load L_i is classified as sensitive.

$$\mathbf{L}_i^{'} = [\frac{I_j - C^i_j}{m^i_j}], \quad \forall j \in S_i \tag{8.7}$$

$$\Delta L_i = \frac{|L_i| - \min(\mathbf{L}_i^{'})}{|L_i|} \tag{8.8}$$

8.4 Evaluation

In order to validate the accuracy of our approach, IEEE 14 bus system [171] is used. The system consists of 14 buses, 5 generators, 11 loads and 20 branches (transmission lines and transformers). As per the algorithm 14, a total of 400 critical and 600 non-critical outages are identified with k ranging from [1,3]. The former set of 400 outage combinations are referred to as true positive cases and the latter are called true negatives. The blackout criteria used is 40% of total system load.

Table 8.1 summaries the results of the experiments. The size of the boolean vector (state) is 20 (equal to number of branches). As shown in the Table 8.1, a small amount of overhead, 18 secs is added for constructing BDDs. On an average, fixed point computation i.e. identification of cascade progression, for true positive cases, takes 7 milliseconds whereas 0.03 milliseconds for true negatives. Figure 8.2(Left) show the response time of all the true positive and negative cases. These experiments are performed on a 1.7 GHz Intel Core i7 machine with 8 GB RAM.

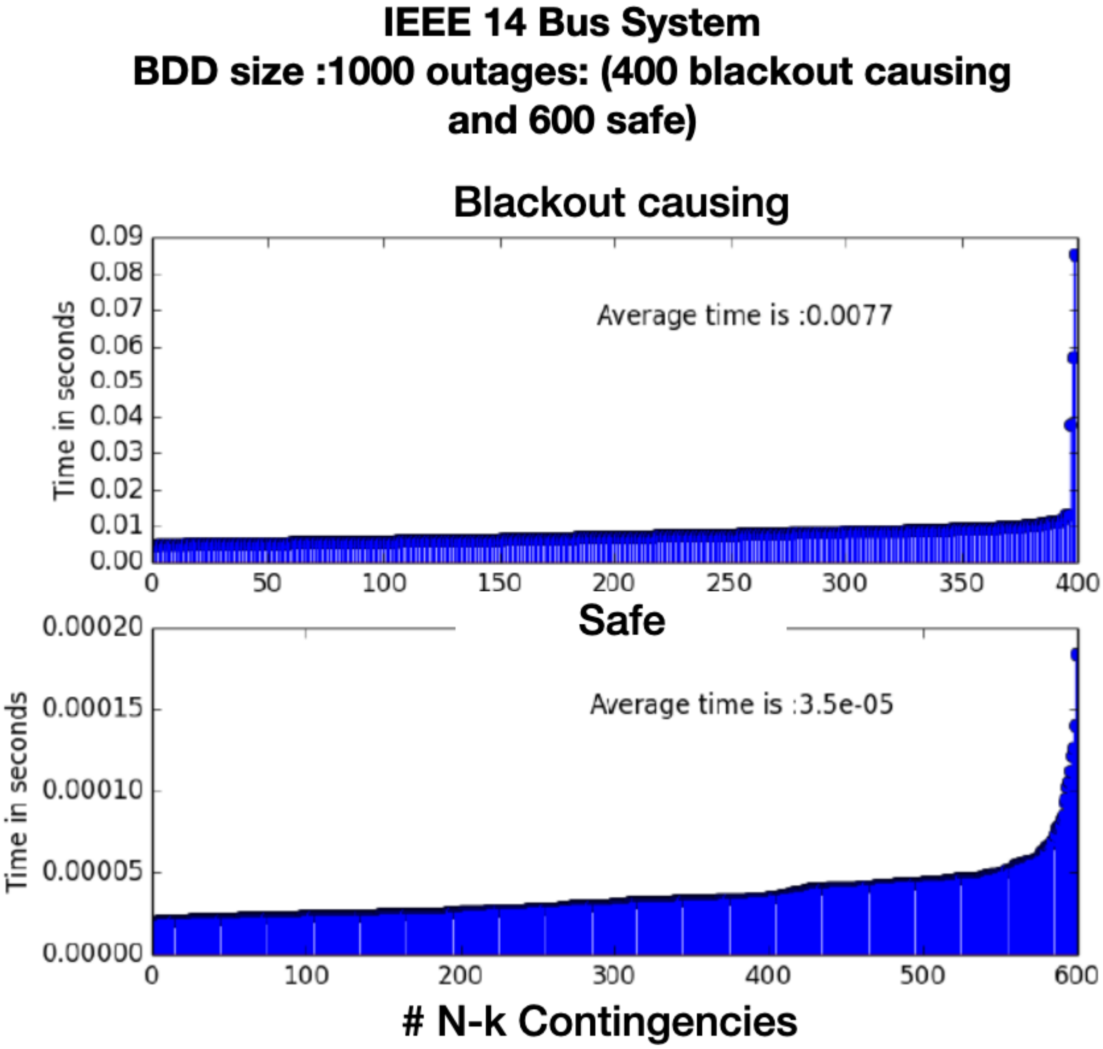


Figure 8.2: Response Time for set membership and fixed point calculation i.e. cascade progression for 1000 different combinations of branch outages for IEEE 14 Bus System. The figure on top shows the time taken for 400 actual cascade causing outages. The figure on the bottom shows the time taken to respond to the 600 True Negative or safe outage combinations

Table 8.2: Solver parameters

Parameter	Value
Problem type	Non Linear
Solver	Sequential Least SQuares Programming
Derivative Calculation	Forward finite difference
Step Size	2500.00
Max Iterations	1000

The optimization routine was able to find a solution for all 400 cases with an average of 29 iterations. Figure 8.4 lists the % load demand reduction. In all cases, load curtailment is restricted to less than 20 % of the net system load (constraint 6) with an average load reduction of approximately 8%. Table 8.2 lists the solver parameters used for the experiments.

Scalability Analysis: Since power systems are large networks its imperative to discuss the impact of scale on our approach. The presented approach consists of 3 major computational tasks 1) *N-k Contingency Analysis*, for small values of k (1, 2, 3 in our case), this process has approximately polynomial run time complexity as the number of combinations increases polynomially with increase in the size of network (N) as well as the time required for solving power flow [172]. 2) *BDD encoding and prognostics*, BDD encoding of singe outage depends upon the size of the network. In order to find the relationship between number of branches and set-membership time (identifying whether a given state is critical or not), 400 true positive and 600 true negative outages are identified for a larger IEEE 39 bus system [173] with 46 branch variables. As the number of branch variables doubles, the set-membership time roughly doubled as shown in figure 8.3. 3) *Optimization*, the number of constraints (one per branch) and design or control variables (loads) increases linearly with the increase in the size of the power network. It is well known that the performance of the optimizer is greatly affected by starting point estimate and the size of the input space of the problem. The sensitivity analysis routine uses full factorial based analysis to estimate a starting point for each load to prevent cascade. For large systems the number of sensitive loads might be very large. A bound on number of control variables (loads) can be placed that can reduce the search space for the optimization problem. If the number of sensitive loads is reduced to 2 (Average number of control variables in first experiment was 7), then the average number of iteration have reduced from 29 to 25 with a slight increase in percentage load reduction from 8.12 to 8.19 %.

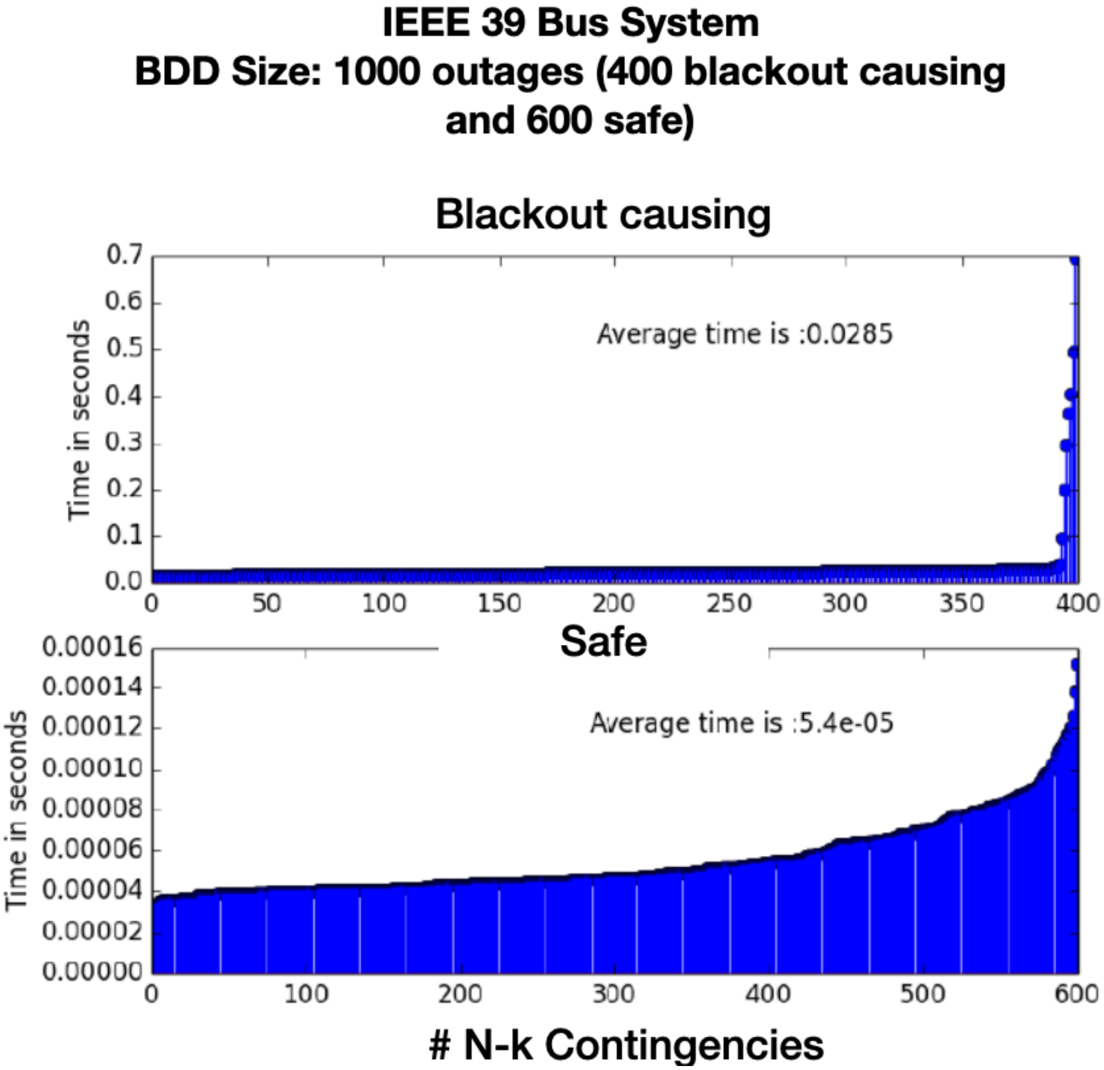


Figure 8.3: Response Time for set membership and fixed point calculation for IEEE 39 Bus System

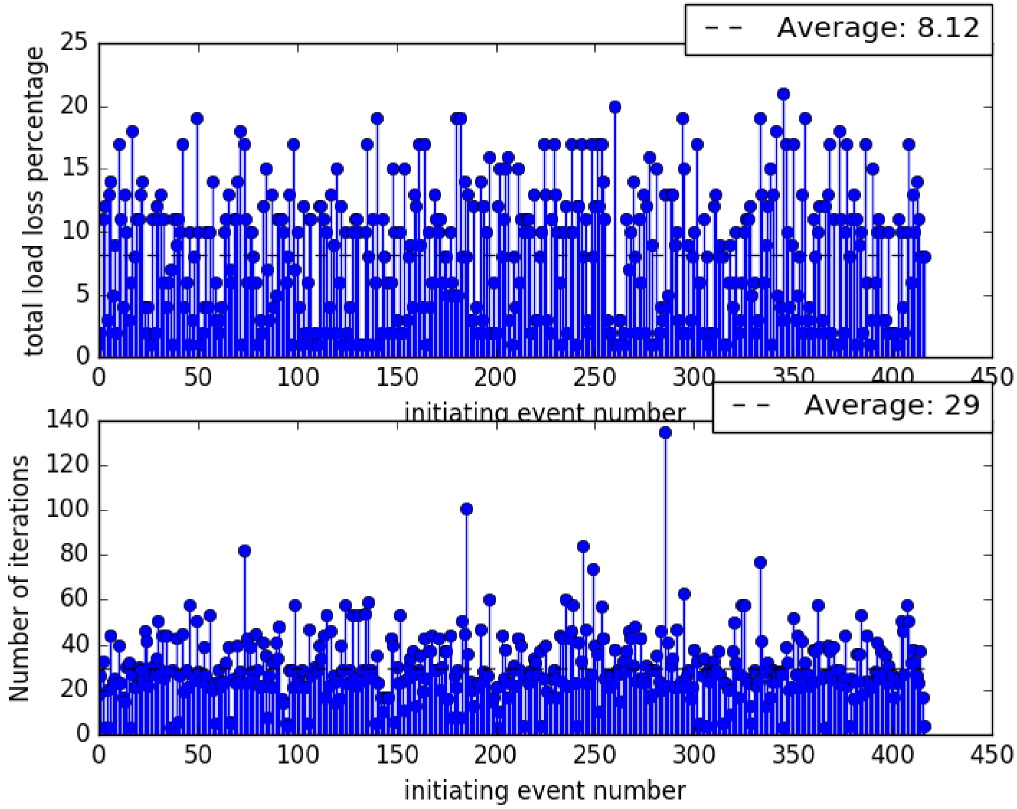


Figure 8.4: The figure on the top shows the net load loss (percentage) due to load control actions and the bottom one shows the number of iterations taken by the optimization engine to find a solution

8.5 Summary

In this chapter, we described the problem of simulating fault cascades in power systems and presented a novel way of storing the useful information from cascade simulations. We showed with the help of simple cascade model, the generation of binary decision diagrams. We presented a detailed algorithm to encode the results of N-k contingencies as BDDs and utilization of BDDs for prognosis of future cascades (if any) given the current system state. We also presented an extensible optimization methodology based on OpenMDAO and OpenDSS to identify load control actions to avoid cascading outages.

8.6 Contributions

The preliminary results for the cascade mitigation work are first published in the workshop paper titled, *A Systematic Approach of Identifying Optimal Load Control Actions for Arresting Cascading Failures in Power Systems* [174] followed by the cascade prognostics and mitigation results in the conference paper titled, *A Binary Decision Diagram Based Cascade Prognostics Scheme For Power Systems* [175].

Chapter 9

Conclusions and Future Work

In this thesis, we briefly described the composition of a cyber physical energy system (CPES) and also established the need of a new fault modeling as well as diagnosis framework by discussing the drawbacks of existing methodologies. To this end, we presented a novel graph-based, qualitative fault modeling formalism, Temporal Causal Diagrams (TCDs) capable of capturing fault effect propagation in physical and cyber sub-system while considering detection and stuck faults. We defined and verified the execution semantics of a TCD fault model by translating it to a network of UPPAAL Timed Automata (TA). We designed and validated component TCD fault models for representing fault effect propagation in transmission lines and its associated protection assemblies. We also demonstrated an automatic process of creating a system level TCD model from the power network topology by composing the individual component TCD models with the help of a polynomial time algorithm. Lastly, we presented a hierarchical diagnosis framework which is capable of diagnosing physical and cyber faults while considering out of order event reception. In the end, we validated the diagnosis framework with simulated traces from a variety of standard IEEE power networks.

We envision the TCD based diagnosis framework can be extended in following dimensions:

- TCD fault model semantics

  1. Extending the TCD fault model execution semantics to consider AND type discrepancies and intermittent faults.
  2. Developing a standalone TCD fault pattern simulator, as the current implementation relies on UPPAAL's concrete simulator for executing a TCD fault model, which places constraints on the size.

- TCD for CPES

  1. Creating a library of TCD fault models for power system equipment such as buses, transformers, lines, generators and their associated protection devices.

2. Add more uses cases from Smart distribution systems and microgrid domain.

- TCD diagnosis

1. Explore other architectural designs such as Decentralized or Distributed architecture to improve the scalability of the reasoning algorithm.

2. Apply multiprocessing techniques to take advantage of current multi-core CPUs to speed up the TFPG reasoner response to different events.

Chapter 10

Publications

Book Chapters

* **Chhokra, Ajay**, Abhishek Dubey, Nagabhushan Mahadevan, Saqib Hasan, and Gabor Karsai. *Diagnosis in Cyber-Physical Systems with Fault Protection Assemblies*, pages 201–225. Springer International Publishing, Cham, 2018.

Journal Articles

* Shashank Shekhar, **Ajay Chhokra**, Hongyang Sun, Aniruddha Gokhale, Abhishek Dubey, Xenofon Koutsoukos, and Gabor Karsai. Urmila: Dynamically trading-off fog and edge resources for performance and mobility-aware iot services. *Journal of Systems Architecture*, 107:101710, 2020.

* **Chhokra, Ajay D**, Nagabhushan Mahadevan, Abhishek Dubey, and Gabor Karsai. Hierarchical reasoning about faults in cyber-physical energy systems using temporal causal diagrams. *International Journal of Prognostics and Health Management*, 9, 2018.

Conference Papers

* Yogesh D. Barve, Shashank Shekhar, **Chhokra, Ajay**, Shweta Khare, Anirban Bhattacharjee, Zhuangwei Kang, Hongyang Sun, and Aniruddha Gokhale. Fecbench: A holistic interference-aware approach for application performance modeling. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pages 211–221, 2019.

* Anirban Bhattacharjee, **Chhokra, Ajay Dev**, Zhuangwei Kang, Hongyang Sun, Aniruddha Gokhale, and Gabor Karsai. Barista: Efficient and scalable serverless serving system for deep learning prediction services. In *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pages 23–33, 2019.

* Anirban Bhattacharjee, **Chhokra, Ajay Dev**, Hongyang Sun, Shashank Shekhar, Aniruddha Gokhale, Gabor Karsai, and Abhishek Dubey. Deep-edge: An efficient framework for deep learning model update

on heterogeneous edge. In *2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC)*, pages 75–84, 2020.

* Saqib Hasan, **Chhokra, Ajay**, Abhishek Dubey, Nagabhushan Mahadevan, Gabor Karsai, Rishabh Jain, and Srdjan Lukic. A simulation testbed for cascade analysis. In *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2017.

* R. Jain, S. M. Lukic, **A. Chhokra**, N. Mahadevan, A. Dubey, and G. Karsai. An improved distance relay model with directional element, and memory polarization for tcd based fault propagation studies. In *2015 North American Power Symposium (NAPS)*, pages 1–6, Oct 2015.

* Nagabhushan Mahadevan, Abhishek Dubey, **Chhokra, Ajay**, Huangcheng Guo, and Gabor Karsai. Using temporal causal models to isolate failures in power system protection devices. *IEEE Instrumentation & Measurement Magazine*, 18(4):28–39, 2015.

* Shashank Shekhar, **Chhokra, Ajay**, Hongyang Sun, Aniruddha Gokhale, Abhishek Dubey, and Xenofon Koutsoukos. Urmila: A performance and mobility-aware fog/edge resource management middleware. In *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*, pages 118–125, 2019.

* Shashank Shekhar, **Chhokra, Ajay Dev**, Anirban Bhattacharjee, Guillaume Aupy, and Aniruddha Gokhale. Indices: Exploiting edge resources for performance-aware cloud-hosted services. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pages 75–80, 2017.

* **Chhokra, Ajay**, Sherif Abdelwahed, Abhishek Dubey, Sandeep Neema, and Gabor Karsai. From system modeling to formal verification. In *2015 Electronic System Level Synthesis Conference (ESLsyn)*, pages 41–46. IEEE, 2015.

* **Chhokra, Ajay**, Saqib Hasan, Abhishek Dubey, and Gabor Karsai. A binary decision diagram based cascade prognostics scheme for power systems. In *2020 American Control Conference (ACC)*, pages 3011–3016, 2020.

* **Chhokra, Ajay**, Nagabhushan Mahadevan, Abhishek Dubey, and Gabor Karsai. Qualitative fault modeling in safety critical cyber physical systems. In *Proceedings of the 12th System Analysis and Modelling Conference*, SAM '20, page 128137, New York, NY, USA, 2020. Association for Computing Machinery.

* **Chhokra, Ajay Dev**, Abhishek Dubey, Nagbhushan Mahadevan, Daniel Allen Balasubramanian, and Gabor Karsai. Towards Diagnosing Cascading Outages in Cyber Physical Energy Systems using Temporal

Causal Models. In *Annual Conference of the Prognostics and Health Management Society 2017*, page 16, St. Petersberg, Florida, 2017. PHM Society.

Workshop Papers

* S. Hasan, A. Dubey, **A. Chhokra**, N. Mahadevan, G. Karsai, and X. Koutsoukos. A modeling framework to integrate exogenous tools for identifying critical components in power systems. In *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6, April 2017.

* **A. Chhokra**, A. Dubey, N. Mahadevan, and G. Karsai. A component-based approach for modeling failure propagations in power systems. In *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6, April 2015.

* **Chhokra, Ajay**, Amogh Kulkarni, Saqib Hasan, Abhishek Dubey, Nagabhushan Mahadevan, and Gabor Karsai. A systematic approach of identifying optimal load control actions for arresting cascading failures in power systems. In *Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, pages 41–46, New York, NY, USA, 2017. ACM.

Poster and Work in Progress Presentations

* Yogesh Barve, Shashank Shekhar, **Chhokra, Ajay**, Shweta Khare, Anirban Bhattacharjee, and Aniruddha Gokhale. Fecbench: An extensible framework for pinpointing sources of performance interference in the cloud-edge resource spectrum. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 331–333, 2018.

* A. Chhokra, S. Hasan, A. Dubey, N. Mahadevan, and G. Karsai. Wip abstract: Diagnostics and prognostics using temporal causal models for cyber physical energy systems. In *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, pages 87–88, April 2017.

* Ajay Chhokra, Abhishek Dubey, Nagbhushan Mahadevan, and Gabor Karsai. Distributed reasoning for diagnosing cascading outages in cyber physical energy systems: Poster abstract. In *Proceedings of the 7th International Conference on Cyber-Physical Systems*, ICCPS '16, pages 33:1–33:1, Piscataway, NJ, USA, 2016. IEEE Press.

* Shashank Shekhar, **Chhokra, Ajay**, Hongyang Sun, Aniruddha Gokhale, Abhishek Dubey, and Xenofon Koutsokos. Supporting fog/edge-based cognitive assistance iot services for the visually impaired: Poster

abstract. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, page 275276, New York, NY, USA, 2019. Association for Computing Machinery.

Appendix A

US Power Grid

In the early days of power system industry, individual companies operated isolated electrical systems. Most systems were not interconnected with neighboring systems. The first interconnected systems in mainland U.S.A (Texas) were Texas Power & Light and West Texas Systems, that occurred in 1924. There are number of advantages to an interconnected systems such as, reduction in the total generation capacity required, reduced power production costs and enhanced reliability. Eventually power systems began to interconnect with their neighboring systems.

Today, the electrical grid that powers mainland North America is divided into four regions or interconnections (two minors and two majors). An interconnection, also known as a wide area synchronous grid, is a region of interconnected AC power systems operating at the same frequency and phase with one another, though not with other interconnections. These regions are overseen by NERC, a non-profit corporation consisting of industry expert with an authority to establish and enforce reliability standards. In addition to reliability standards, NERC also publishes operating studies and statistics [21].

In 2007, compliance with approved NERC reliability standards became mandatory and NERC delegated its compliance monitoring and enforcing responsibility to eight regional entities: *Florida Reliability Coordinating Council* (FRCC) [176], *Midwest Reliability Organization* (MRO) [177], *Northeast Power Coordinating Council* (NPCC) [178], *Reliability First* (RF) [179], *South East Reliability Corporation* (SERC) [180], *Southwest Power Pool* (SPP) [181], *Texas Reliability Entity* (TRE) [182], *Western Electricity Coordinating Council* (WECC) [183]. The four interconnections in North America and their respective regional reliability entities are illustrated in Figure A.1 and briefly described as follows:

1. *The Eastern Interconnection:* It is the largest interconnection with a peak load of 600,000 MW that reaches from Central Canada eastward to the Atlantic coast (excluding Quebec), south to Florida, and back west to the foot of the Rockies (excluding most of Texas). It

is tied to the Western Interconnection with six DC tie-lines[1], to the Texas Interconnection with two DC, and to the Quebec Interconnection with four DC and a variable-frequency transformer. Six regional reliability coordinators facilitate reliable power grid operations in this interconnection by working with the electrical energy industry that operate within its boundaries.

2. *The Western Interconnection:* It is the second largest interconnection with peak load of 125,000 MW. It extends from Canada to Mexico and includes the provinces of Alberta and British Columbia, the northern portion of Baja California, Mexico, and all or portions of the 14 western states between. WECC is the only regional entity operating in the interconnection.

3. *The ERCOT Interconnection:* This interconnection covers the majority of the state of Texas. The peak load of the Texas Interconnection is about 70,000 MW. Aprart from two DC tie lines connection with Eastern Interconnection, there are three other DC links to Mexico. TRE is the only regional entity that operates in the interconnection.

4. *The Quebec Interconnection:* It is the smallest interconnection with a peak load of 30,000 MW. This interconnection too, has only one regional entity, NPCC, that monitors, enforce and creates standards.

[1] A tie-line is a transmission line that connects an interconnection to its neighboring interconnection

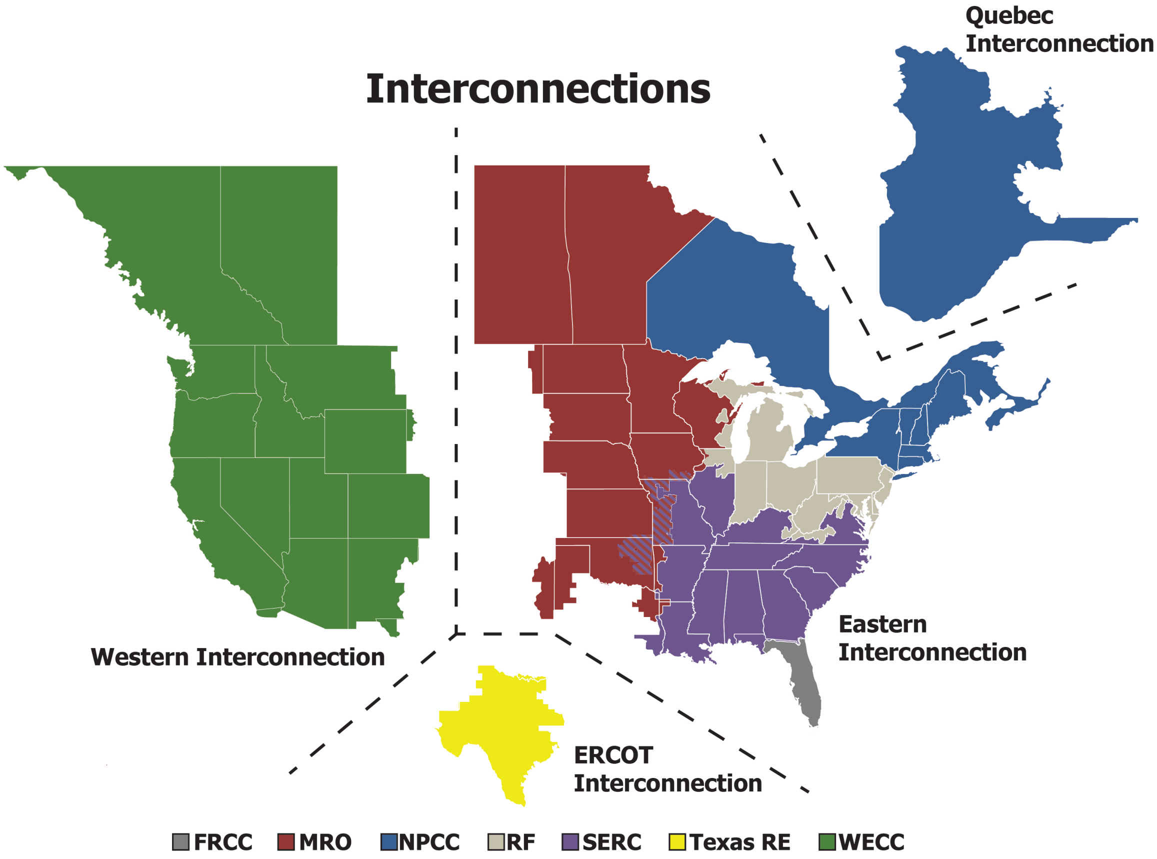


Figure A.1: Interconnections, and their respective Regional Reliability Councils, reprinted from [2]

Appendix B

Energy Market Participants

In addition to matching system demand with generation, these coordinators must also be able to respond quickly to ever-changing system conditions, including rapidly increasing or decreasing demand or sudden loss of generation. To meet this requirement, reliability coordinators procure and reserve additional capacity from certain generators that can be used in case of emergencies. These capacity reserves are called *Ancillary services*. For the market to operate reliably and efficiently, the various market participants need to work closely together and operate according to standard market protocols. Figure B.1, illustrates the main market participants and their interactions in a regional competitive retail market. Description of these market participants is as follows:

- *Resource Entities* (RE): REs are the only entities that can own generation or contract for instructed demand. They negotiate privately with other market participants to sell their energy (or demand) and communicate to reliability coordinators through their qualified scheduling entities.

- *Load Serving Entities* (LSE): LSEs are the only ones allowed to sell electricity to consumers. It forecasts customer load and negotiate privately with other market participants. Similar to REs, LSEs communicate to reliability coordinators indirectly[1] through their QSE.

- *Transmission and Distribution Service Providers* (TDSP): TDSPs provide the electricity transportation infrastructure, and work with reliability coordinators to jointly manage the transmission system. TDSPs directly communicate with reliability coordinators to ensure grid safety.

- *Qualified Scheduling Entities* (QSE): Reliability coordinators interact with markets through QSEs. They serve as the primary information provider of supply and demand. Reliability coordinators communicate all operational instructions to QSE, which is passed to appropriate

[1] LSEs will interact directly with reliability council when they need to submit switching requests, where customers choose a new LSE

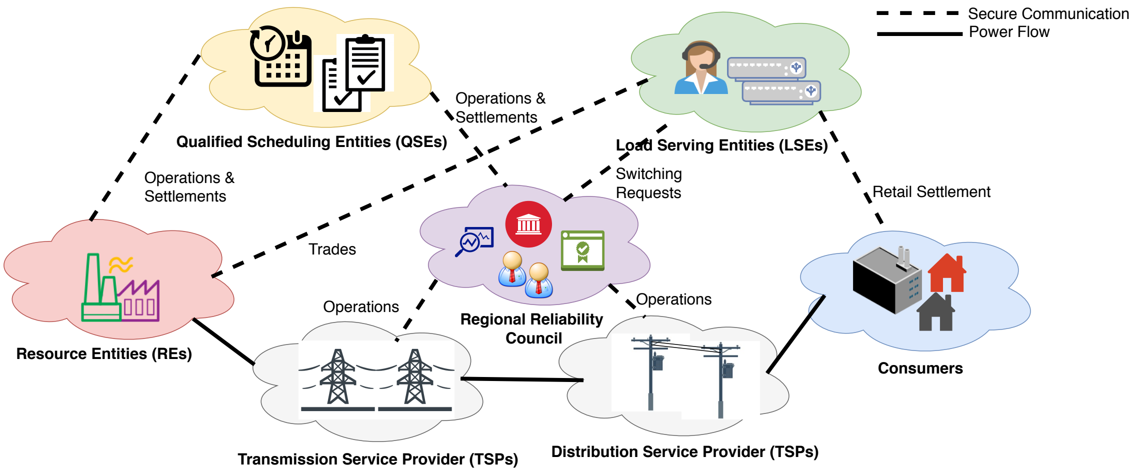


Figure B.1: Energy market participants and their interactions

entities. QSE submit bids and offers on behalf of REs and LSEs. A QSE is also responsible for submitting a operating plan for all resources it represents and offering or procuring ancillary services as needed to serve their represented load.

Appendix C

Mis-operation Causes

The reasons of protection system mis-operations categorized by NERC are as follows:

- **AC system**: This category includes mis-operations due to problems in the ac inputs to the protection system. Examples would include mis-operations associated with CT saturation, loss of potential etc.

- **Communication failures**: This category includes mis-operations due to failures in the communication systems associated with protection schemes. Examples would include mis-operations caused by loss of carrier, spurious transfer trips associated with noise, loss of fiber optic communication equipment, or microwave problems associated with weather conditions.

- **DC system**: This category includes mis-operations due to problems in the dc control circuits. These include problems in trip wiring to breakers, or loss of dc power to a relay.

- **Incorrect settings**: This category includes mis-operations due to issued setting errors, including those caused by modeling errors.

- **Logic errors**: This category includes mis-operations due to issued logic setting errors associated with programming microprocessor relay inputs, outputs, business logic, or protection function mapping to communication or physical I/O points.

- **Relay malfunctions**: This category includes mis-operations due to improper operation of the relays themselves. These may be due to component failures, physical damage to a device, firmware problems, or manufacturer errors. Examples would include mis-operations caused by changes in relay characteristic. Failures of auxiliary tripping relays fall under this category.

Appendix D

Physical Fault Summary

Table D.1: Physical component faults and protection schemes

Component	Fault	Causes	Effect	Protection Scheme
Generator	Overload (Fault Code 49)	Increased power on the generators load side	Stator winding overheating	Thermal image relay (keeping track of temperature) or over current relay.
	Unbalanced loads (Fault Code 46)	Sudden loss or connection of heavy loads, or poor distribution of loads	Generators full capacity cannot be utilized, rise of negative sequence components (rotation in reverse direction) leading to heavy currents in the rotor	Negative sequence over current relay (unsymmetrical loads would give rise to negative sequence components)
	Reverse power conditions (Fault Code 32)	Parallel operation of a generator with other units may force motor behavior (due to load unbalance or poor load sharing between generators)	Generator behaves as motor and draws power from the network, turbine connected to generator will be damaged due to winding overheating	Directional power relay with reverse power setting option
	Out-of-Step (Fault Code 78)	Loss of synchronism due to line switching, connection/disconnection of heavy loads, electrical faults, etc	Winding stress, high rotor iron currents, pulsating torques, mechanical resonances	Out of step protection relay which tracks the impedance calculated from measured voltage and current. In case of fault; there is nearly a step change in voltage or current.

Component	Fault	Causes	Effect	Protection Scheme
	frequency variations (Fault Code 81)	Improper speed control, grid disturbance or sudden load cut off	Severe speed changes will cause over fluxing , serious damage to the turbine generator set	Frequency protection relay which tracks frequency and trips the breaker in case of abnormal frequencies
	under or over voltage (Fault Code 27, 59)	System disturbance or malfunctioning AVR	Over fluxing and winding insulation failure	Over/ Under voltage relay with pre-set voltage limits defined in the settings
	Internal Faults (Fault Code 87)	Phase to phase and 3 phase to ground	gives rise to large amount of currents that can damage the winding	Differential protection on each side of generator
	Stator Earth Fault (Fault Code 64)	Winding insulation failure or inter-turn insulation failure	Thermal and magnetic imbalance and damage to rotor metallic parts	voltage relay detect earth fault
	Loss of Field (Fault Code 40)	Loss of exciter source, open or short circuit at the field winding	Loss of synchrmosim between the rotor and stator fluxes, draws reactive power from the grid and provokes severe torque oscillations	Impedance relay is used to implement this technique
	Rotor Earth fault (Fault Code 61F)	Winding insulation failure or inter-turn insulation failure	Thermal and magnetic imbalance and damage to rotor metallic parts	Voltage relay energized by neutral VT
Transmission Line	Over voltage (Fault Code 59)	Lightning or line switching	Give rise to transient over-voltages which can damage the insulation	Surge Arrestors/ Overvoltage relay with preset voltage limits defined in settings
	Power Swing Blocking (Fault Code 68)	Line switching, generator disconnection, addition/loss of load	Loss of synchronism between a generator and the rest of the system as seen by the measured voltages, phase sequence, phase angles, frequencies resulting in swing in power flows	A blocking relay provides this protection and has the same type of characteristic as a distance relay

Component	Fault	Causes	Effect	Protection Scheme
	Over-current (Fault Code 50/51)	Due to short circuit, single phase to ground or phase to phase faults. Can occur due to tree limbs falling on lines, etc	Gives rise to heavy current that flows through the winding conductor and causing overheating of the conductor which will deteriorate it	An over current protection relay which also serves as a back up for distance protection is used. In case the distance protection (primary protection) malfunctions, over-current protection will send trip commands
	Earth Fault (Fault Code 50N/51N)	Direct connection to ground of one or more phases	Gives rise to higher voltages on other lines and stresses the insulation of cables and other equipment connected to the system	Over-current relay that continuously monitors the current through the neutral and sends trip signals to the breaker upon fault detection
	Phase and ground faults (Fault Code 21)	Short circuit between phases or direct connection with ground	Reduction in overall line impedance (V/I) due to fault conditions. Fault current can overheat the transmission line and can cause damage to the conductor.	Distance protection relay serves as a primary protection for transmission lines. It keeps track of the line impedance and sends trip signal to the breaker if the line impedance changes (due to fault)
Transformer	Overload (Fault Code 58)	Increased power on the secondary side of the transformer	Transformer overheating	Thermal image relay (keeping track of temperature) / over current relay
	Internal Faults (Fault Codes 50/51, 50N/51N, 87)	Internal faults can be short circuits, or earth faults, or overloading	Can cause damage to transformer windings	Differential protection with CTs on each side of transformer (Unit Protection), Over-current relays

Component	Fault	Causes	Effect	Protection Scheme
	Loss of directional sensitivity (Fault Code 67/67N)	Fault in nearby (parallel) feeder/bay causing tripping in the healthier feeder/bay due to poor selectivity of the relay	Tripping of additional feeders, thus pushing the system towards larger outages	Directional Overcurrent relay detects the direction of current flows in to and flow out from the protected unit. A trip signal will be sent to breakers if direction of flow-in and flow-out current are not the same
	Breaker Failure (Fault Code 50BF)	Breaker malfunctioning	Unable to isolate faulty equipment due to tripping failures (longer existence of fault currents, thus more damage to equipment)	Breaker failure relay which operates with its algorithm to try to open the breaker, otherwise it sends trip command to nearby breakers to isolate the faulted equipment to stop feeding fault currents
Load	Loss of synchronism (synchronous machines only) (Fault Code 55)	Increase in load causes a decrease in the busbar voltage, or due to decrease in the field current that causes the motor torque to decrease	Damage occurs to the dampers and rotor windings due to loss of synchronism	Power factor relay that responds to the change in power factor that occurs when there is pole slipping (weakening of synchronizing torque to maintain synchronism under the same load)
	Under-voltage (Fault Code 27)	System disturbance or load increase	Under voltage results in over-currents which can damage insulation	Under voltage relay with pre-defined voltage limits defined in the relays settings
	Short circuit (Fault Code 50/51)	Phase to phase short circuit in the winding , at the motor terminals or between cables	Destroy the machine due to over-heating and electro-dynamic forces created by the high currents	Over-current relay with a preset value which sends a trip signal if the current exceeds its preset value.

Component	Fault	Causes	Effect	Protection Scheme
	Overload (Fault Code 49)	Increase of load torque, or decrease in the motor torque due to busbar voltage or decrease in DC Field current	High currents drawn by the motor affects insulation, and thus reduces the machines life expectancy	Thermal image relay (keeping track of temperature and has a thermal time constant) / over-current relay
	Earth fault (Fault Code 50N/51N)	Machine insulation damage	Results in a fault current that flows from windings to earth via stator lamination	Over-current relays with neutral module.

Appendix E

2003 US-Northeastern Blackout Events Summary

Table E.1: Cascade Event Sequence

Phase	Time	Event
Phase 1	12:15:00 EDT	MISOs state estimator software solution was compromised, and MISOs single contingency reliability assessment became unavailable.
	13:31:34 EDT	Eastlake Unit 5 generation tripped in northern Ohio
	14:02 EDT	Stuart-Atlanta 345-kV transmission line tripped in southern Ohio
Phase 2	14:14 EDT	FE alarm and logging software failed. Neither FEs control room operators nor FEs IT EMS support personnel were aware of the alarm failure.
	14:20 EDT	Several FE remote EMS consoles failed. FEs Information Technology (IT) engineer was computer auto-paged
	14:27:16 EDT	Star-South Canton 345-kV transmission line tripped and successfully reclosed.
	14:32 EDT	AEP called FE control room about AEP indication of Star-South Canton 345-kV line trip and reclosure. FE had no alarm or log of this line trip
	14:41 EDT	The primary FE control system server hosting the alarm function failed. Its applications and functions were passed over to a backup computer. FEs IT engineer was auto-paged.
	14:54 EDT	The FE back-up computer failed and all functions that were running on it stopped. FEs IT engineer was auto-paged.
Phase 3	15:05:41 EDT	Harding-Chamberlin 345-kV line tripped.
	15:31-33 EDT	MISO called PJM to determine if PJM had seen the Stuart-Atlanta 345-kV line outage. PJM confirmed Stuart-Atlanta was out.
	15:32:03 EDT	Hanna-Juniper 345-kV line tripped.
Phase 4	15:39:17 EDT	Pleasant Valley-West Akron 138-kV line tripped and reclosed at both ends after sagging into an underlying distribution line.
	15:42:05 EDT	Pleasant Valley-West Akron 138-kV West line tripped and reclosed.
	15:44:40 EDT	Pleasant Valley-West Akron 138-kV West line tripped and locked out.
	15:42:49 EDT	Canton Central-Cloverdale 138-kV line tripped on fault and reclosed.
	15:45:39 EDT	Canton Central-Cloverdale 138-kV line tripped on fault and locked out.
	15:42:53 EDT	Cloverdale-Torrey 138-kV line tripped.

Phase	Time	Event
	15:44:12 EDT	East Lima-New Liberty 138-kV line tripped from sagging into an underlying distribution line.
	15:44:32 EDT	Babb-West Akron 138-kV line tripped on ground fault and locked out.
	15:45:40 EDT	Canton Central 345/138 kV transformer tripped and locked out due to 138 kV circuit breaker operating multiple times, which then opened the line to FEs Cloverdale station.
	15:51:41 EDT	East Lima-N. Findlay 138-kV line tripped, likely due to sagging line, and reclosed at East Lima end only.
	15:58:47 EDT	Chamberlin-West Akron 138- kV line tripped.
	15:59:00 EDT	West Akron 138-kV bus tripped, and cleared bus section circuit breakers at West Akron 138 kV.
	15:59:00 EDT	West Akron-Aetna 138-kV line opened.
	15:59:00 EDT	Barberton 138-kV line opened at West Akron end only. West Akron-B18 138-kV tie breaker opened, affecting West Akron 138/12-kV transformers 3, 4 and 5 fed from Barberton.
	15:59:00 EDT	West Akron-Granger-Stoney-Brunswick-West Medina opened.
	15:59:00 EDT	West Akron-Pleasant Valley 138-kV East line (Q-22) opened
	15:59:00 EDT	West Akron-Rosemont-Pine-Wadsworth 138-kV line opened.
	16:05:55 EDT	Dale-West Canton 138-kV line tripped due to sag into a tree, reclosed at West Canton only
	16:05:57 EDT	Sammis-Star 345-kV line tripped
	16:06:02 EDT	Star-Urban 138-kV line tripped
	16:06:09 EDT	Richland-Ridgeville-Napoleon- Stryker 138-kV line tripped on overload and locked out at all terminals
	16:08:58 EDT	Ohio Central-Wooster 138-kV line tripped
Phase 5	16:05:57 EDT	Sammis-Star 345-kV tripped by zone 3 relay.
	16:08:59 EDT	Galion-Ohio Central-Muskingum 345-kV line tripped on zone 3 relay.
	16:09:06 EDT	East Lima-Fostoria Central 345-kV line tripped on zone 3 relay, causing major power swings through New York and Ontario into Michigan.
	16:09:08 EDT	Michigan Cogeneration Venture plant reduction of 300 MW (from 1,263 MW to 963 MW)
	16:09:17 EDT	Avon Lake 7 unit trips (82 MW) 16:09:17 EDT: Burger 3, 4, and 5 units trip (355 MW total)
	16:09:30 EDT	Kinder Morgan units 3, 6 and 7 trip (209 MW total)
Phase 6	16:10:36.2 EDT	Argenta-Battle Creek 345-kV line tripped

Phase	Time	Event
	16:10:36.3 EDT	Argenta-Tompkins 345-kV line tripped
	16:10:36.8 EDT	Battle Creek-Oneida 345-kV line tripped
	16:10:37 EDT	Sumpter Units 1, 2, 3, and 4 units tripped on under-voltage (300MW near Detroit)
	16:10:37.5 EDT	MCV Plant output dropped from 963 MW to 109 MW on over-current protection
	16:10:38.2 EDT	Hampton-Pontiac 345-kV line tripped.
	16:10:38.4 EDT	Thetford-Jewell 345-kV line tripped.
	16:10:38.6 EDT	Erie West-Ashtabula-Perry 345-kV line tripped at Perry.
	16:10:38.6 EDT	Large power surge to serve loads in eastern Michigan and northern Ohio swept across Pennsylvania, New Jersey, and New York through Ontario into Michigan.
	16:10:39.5 EDT	Bay Shore-Monroe 345-kV line tripped.
	16:10:39.6 EDT	Allen Junction-Majestic- Monroe 345-kV line tripped.
	16:10:39 EDT	Homer City-Watercure Road 345 kV line tripped.
	16:10:39 EDT	Homer City-Stolle Road 345 kV line tripped.
	16:10:40 EDT	Majestic-Lemoyne 345-kV line.
	16:10:40 EDT	Lakeshore unit 18 (156 MW, near Cleveland) tripped on under-frequency.
	16:10:41 EDT	Belle River unit 1 tripped (637 MW) on out-of-step.
	16:10:41 EDT	St. Clair unit 7 tripped (221 MW, DTE unit) on high voltage.
	16:10:41.7 EDT	Eastlake 1, 2, and 3 units (304 MW total, near Cleveland) tripped on under-frequency.
	16:10:41.7 EDT	Avon Lake unit 9 (580 MW, near Cleveland) tripped on under-frequency.
	16:10:41.7 EDT	Perry 1 nuclear unit (1,223 MW, near Cleveland) tripped on underfrequency.
	16:10:41.8 EDT	Fostoria Central-Galion 345-kV line.
	16:10:41.9 EDT	Beaver-Davis Besse 345-kV line.
	16:10:42 EDT	Bay Shore Units 1-4 (551 MW near Toledo) tripped on over-excitation.
	16:10:42 EDT	Ashtabula unit 5 (184 MW, near Cleveland) tripped on under-frequency.
	16:10:42 EDT	Greenwood unit 1 tripped (253 MW) on low voltage, high current.
	16:10:42 EDT	Trenton Channel units 7A, 8 and 9 tripped (648 MW).
	16:10:43 EDT	West Lorain units (296 MW) tripped on under-voltage.
	16:10:43 EDT	Keith-Waterman, 230-kV line tripped.
	16:10:44 EDT	South Ripley-Erie East 230 kV, East Towanda-Hillside 230 kV and South Ripley-Dunkirk 230 kV lines tripped.
	16:10:45 EDT	Wawa-Marathon 230-kV and Branchburg-Ramapo 500-kV lines tripped.

Phase	Time	Event
Phase 7	16:10:47 EDT	New York-New England upstate transmission lines disconnected
	16:10:49 EDT	New York transmission system split along Total East interface
	16:10:50 EDT	The Ontario system just west of Niagara Falls and west of St. Lawrence separated from the western New York island.
	16:11:22 EDT	Southwest Connecticut separated from New York City.
	16:11:57 EDT	Remaining transmission lines between Ontario and eastern Michigan separated. By this point most portions of the affected area were blacked out

Appendix F

Power System Control

Energy management is the process of monitoring, coordinating, and controlling the generation, transmission, and distribution of electrical energy in power systems. A characteristic of a power system is that the load, i.e., the electric power consumption, varies significantly both in short and in the long time scales. Since the transmission system provides negligible energy storage capabilities, supply (power produced by generators) and demand (power consumed by loads) must be balanced by either generation or load side at all time instants. Imbalance in supply and demand leads to frequency deviations that if too large will have severe impacts on the system operation. In addition to keeping the above mentioned balance, the delivered electricity must conform to certain quality criteria. This means that the voltage magnitude, frequency, and wave shape must be controlled within specified limits.

In general, instrumentation and control system in power system is performed with the participation of large number of devices organized in multiple levels as shown in figure F.1. The ground level consists of power system apparatus (physical components) that generate, transport and distribute power. The next level is the Process level, the lowest level of instrumentation and control devices (cyber components). The devices in this level are physically connected to power systems and are sensing their current status. The devices include current transformers, voltage transformer, thermal resistance detectors and circuit breakers. The following level is the device, unit or bay[1] level comprised of Integrated Electronic Devices (IEDs). An IED is a device incorporating one or more processors that has the capability to receive or send data/control from or to an external source. With its enhanced microprocessor and communication technology, IEDs provides self and external circuit monitoring, primary and secondary protection, real-time synchronization for event reporting, along with remote and local control and data acquisition for use in network analysis. The power system IEDs include protective relays, meters, digital fault recorders, load tap change controller,

[1] A bay refers to an area where power system device such as feeder breaker and all of the instrumentation and control devices associated with it are located.

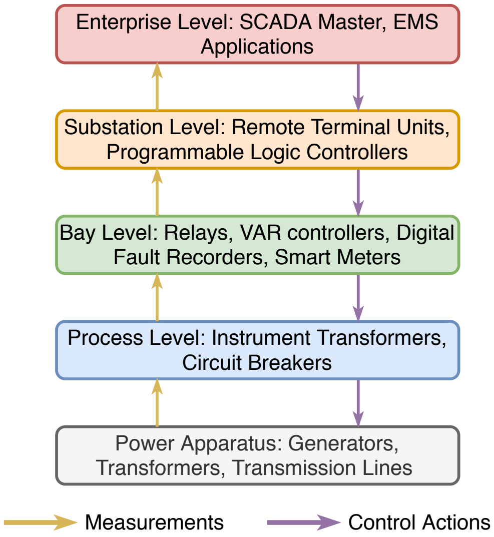

Enterprise Level: SCADA Master, EMS Applications
Substation Level: Remote Terminal Units, Programmable Logic Controllers
Bay Level: Relays, VAR controllers, Digital Fault Recorders, Smart Meters
Process Level: Instrument Transformers, Circuit Breakers
Power Apparatus: Generators, Transformers, Transmission Lines
Measurements
Control Actions

Figure F.1: Multi-Level Control in Power Systems

VAR controllers. The next level consists of substation controllers that perform data acquisition and control (remote) IEDs and contain local I/O. These devices contain data for the entire station and thus this level referred to as Station level. The devices in this level include Remote Terminal Units (RTUs) and Programmable Logic Controller (PLC). An RTU is a device that can be installed in a remote location, collect sensor data and forward it to another device without processing it. A PLC is an extension of RTU as it has the ability to perform user defined processing of the sensor data to extract useful information. The last level in the hierarchy is known as Enterprise level, it consists of communication front-end that acquires collected data from various station level devices, power network analysis and energy management application servers. Enterprise level devices are hosted in multiple control centers typically called energy control centers. Instrumentation devices in the rest of the levels except power apparatus are hosted in various substations.

From functionality point of view energy management is composed of two main components: 1) *Supervisory Control and Data Acquisition* (SCADA), 2) *Energy Management Systems* (EMS) that are described in the following sub-sections

F.0.1 Supervisory Control and Data Acquisition

Supervisory Control and Data Acquisition (SCADA) refer to a system that collects data from various sensors/ equipment in remote locations and then transfers to a central node to be processed

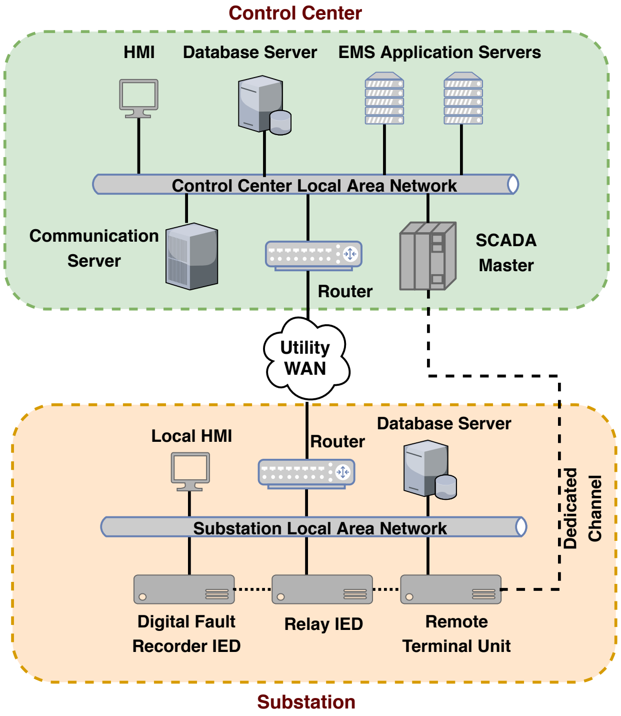


Figure F.2: Control center and substation architecture without security

for determining control actions for managing the state of the equipment by computer aided tools or an operator. In power systems data is transferred from various substations to control center and includes measurements (voltage, current, frequency, active and reactive power flows), relay status bits and breaker state. SCADA system allows continuous monitoring and real time remote control of power systems. Traditionally, a master station at control center polls data from remote terminal units located at different substations which in turn collect data from different devices every 2-4 seconds depending upon the criticality and availability of data [184]. However, after widespread use of IEDs and adoption of substation integration, operators can directly access and control IED state with a latency of 40-100 milliseconds [14]. Figure F.2 shows an architecture of a modern day control center and substation SCADA system. The main elements of the SCADA system are described as follows:

- Human Machine Interface (HMI): The SCADA HMI is a core component of a remote monitoring and controlling system as it presents data collected from RTUs and IEDs. It also allows an operator to control power equipment through RTUs and IEDs. SCADA HMI consists of a map board and multi video display unit workstations.

- Application Servers: Modern SCADA master systems have both software and hardware in a distributed architecture. The processing power is distributed among different computers called Application Servers. From a functional point of view, these servers can be categorized into two classes:
  - Database Application Servers: These servers support the database that contains all historical data. The data from these databases is used for performing operational control and security planning in power systems.
  - Advanced Energy management Application Servers: These servers support all Energy Management Systems (EMS) applications such as Automatic General Control, Economic Dispatch, Security Analysis (described in next section). The main characteristic of these servers is its processing power. More than one server may be used for these applications.
- Communication Front End: This system is used for data acquisition from RTUs, PLCs and field equipment. It provides functions such as acquiring the RTU data, protocol conversion, security check, temporary storage of analog and digital data, and detection of analog value and digital state changes.
- External Communication server: This server provides data exchange with other control centers. A standard protocol, such as IEC 60870-6 (TASE.2) [185] is used to exchange real-time and archive data.
- Remote Terminal Unit (RTU): A RTU is a microprocessor based electronic device that interfaces objects in the physical world to a distributed control systems or SCADA by transmitting telemetry data to the system, and by using messages from the supervisory system to control connected objects.
- Data Concentrator: A data concentrator collects the required data from all substation IEDs, including RTUs. Unlike Database application servers, data concentrators are located in substations.
- Integrated Electronic Devices (IEDs): These are microprocessor-based controllers with advanced communication capabilities. IEDs receive data from power equipment and issue control commands, such as tripping circuit breakers if they sense voltage, current, or frequency anomalies. Common types of IEDs include protective relaying devices, On Load Tap Changer

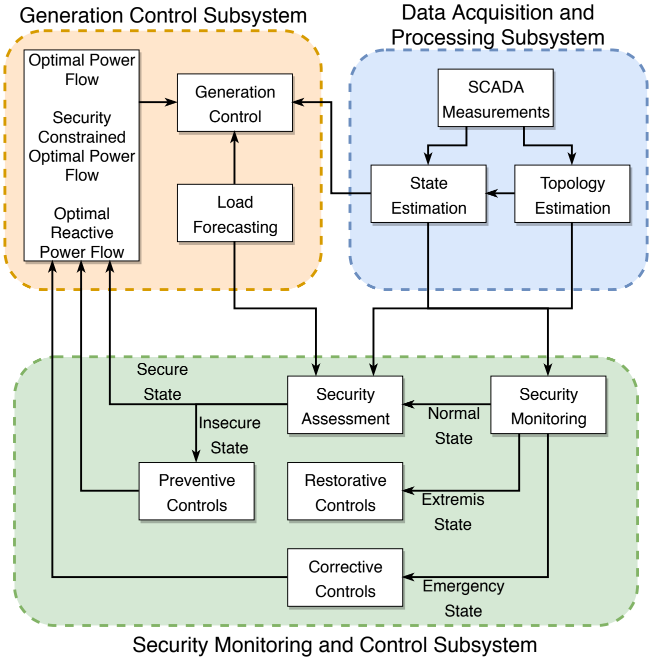


Figure F.3: Functional Diagram of an Energy Management System

controllers, circuit breaker controllers, capacitor bank switches, recloser controllers, voltage regulators, etc.

- Remote Access Controller: It enables remote access to the substation IEDs for remote configuration, access, and data retrieval.

F.0.2 Energy Management System

An Energy Management Systems (EMS) is a system of computer-aided tools used by operators to monitor, control, and optimize the performance of the generation and/or transmission system. These tools can be grouped into three distinct groups resulting in the three subsystems: (a) the data processing subsystem, (b) the security monitoring and control subsystem, and (c) the generation control subsystem. The three subsystems are illustrated in Figure F.3. A brief description is given next.

F.0.2.1 Data Processing Subsystem

The objective of the data processing subsystem is to obtain an accurate estimate of the operating state of the system. This is achieved with data acquired through a large number of remote terminal units. The remote terminal units collect analog measurements (voltage magnitude, power flows, etc.) and status variables (status of breakers, switches, etc.) and transmit this data to the computers of the energy management system via the communication network. There, the topology of the network is updated and the state of the system is estimated.

Topology Processing: The topology of the network characterizes the connectivity between buses (nodes), the shunt elements at each bus, and which generators are connected. This information comes to the EMS from the SCADA in the form of status indicators for each circuit breaker and switch at all buses. This information is referred to as the *bus section breaker-switch* data and provides a mapping of individual bus sections at each substation and how they are connected. The bus section breaker-switch model is converted into *bus-branch* model. A bus-branch model shows the connectivity between transmission lines, transformers, and buses which is essential for performing load flow and state estimation studies.

State Estimation: State estimators perform statistical analysis using a set of m imperfect redundant data telemetered from the power system to determine the state of the system. The state of the system is a function of n state variables. Although the state estimation solution is not a true representation of the system, it is the best possible representation based on the telemeter-ed measurements. Also, it is necessary to have the number of measurements greater than the number of states $m \geq n$ to yield a representation of the complete state of the system. This is known as the *observability* criterion. Typically, m is two to three times the value of n, allowing for a considerable amount of redundancy in the measurement set. Complex nodal voltages are the most commonly used state variables. Turn ratios of transformers with taps that change under operating conditions are also treated as state variables.

Since a measurement is not exact, it can be expressed with an error component as shown in Equation F.1, where z is the measured value, z_T is the true value, and v is the measurement error that represents uncertainty in the measurement. It's a common practice to model v as a random variable with Gaussian probability distribution. Active and reactive power flows, voltage magnitudes are normally used as measurements in state estimators. In general, the measured value, as expressed in Equation F.1, can be related to the state vector, x, by Equation F.2, where $h(x)$ is a vector of nonlinear functions relating the measurements to the state variables. For an arbitrary pair of buses, i, j, power flow from i to j is given by equation eqs. (F.3) and (F.4), where $|V_i|$, $|V_j|$ are the magnitudes of voltage at buses i and j respectively, δ_{ij} is the phase angle difference between bus i and j, g_{ij} and b_{ij} are conductance and susceptance of the line $i-j$ respectively and g_{i_sh} and b_{i_sh} are the shunt conductance and susceptance at bus i respectively.

$$z = z_T + v \tag{F.1}$$

$$z = h(x) + v \tag{F.2}$$

$$P_{ij} = |V_i|^2\left(g_{ij} + g_{i_sh}\right) - |V_i||V_j|\left[g_{ij}cos\left(\delta_{ij}\right) + b_{ij}sin\left(\delta_{ij}\right)\right] \tag{F.3}$$

$$Q_{ij} = |V_i|^2\left(b_{ij} + b_{i_sh}\right) - |V_i||V_j|\left[g_{ij}sin\left(\delta_{ij}\right) + b_{ij}cos\left(\delta_{ij}\right)\right] \tag{F.4}$$

Using equations eqs. (F.3) and (F.4), equation F.2 can be rewritten as F.5, which expresses the measurements (reactive and active power flow) entirely in terms of network parameters (conductance and susceptance, known a priory) and system states (bus voltage and phase angle).

$$\begin{bmatrix} P_{ij} \\ Q_{ij} \end{bmatrix} = \begin{bmatrix} |V_i|^2\left(g_{ij} + g_{i_sh}\right) - |V_i||V_j|\left[g_{ij}cos\left(\delta_{ij}\right) + b_{ij}sin\left(\delta_{ij}\right)\right] \\ |V_i|^2\left(b_{ij} + b_{i_sh}\right) - |V_i||V_j|\left[g_{ij}sin\left(\delta_{ij}\right) + b_{ij}cos\left(\delta_{ij}\right)\right] \end{bmatrix} + \begin{bmatrix} v_{P_ij} \\ v_{Q_ij} \end{bmatrix} \tag{F.5}$$

The most common approach to solving the state estimation problem is using the method of Weighted Least Square (WLS). This is accomplished by identifying the values of the state variables that minimize the performance index, J (the weighted sum of square errors), as shown in equation F.6, where weighting factor, R, is the diagonal covariance matrix of the measurements and error, e is the difference between the true measured value, z_T, and the estimated measured value z. Using equation F.5, F.6 can be re-written as F.7 where the weights are defined by the inverse of the measurements variances. As a result, measurements of higher quality have smaller variances that correspond to their weights having higher values, while measurements with poor quality have smaller weights due to the correspondingly higher variance values.

$$J = e^T R^{-1} e \tag{F.6}$$

$$J = \left(z_T - h(x) \right)^T R^{-1} \left(z_T - h(x) \right) \tag{F.7}$$

F.0.2.2 Security Monitoring and Control Subsystems

Security in power systems is defined as the ability to withstand imminent disturbances (contingencies) without interruption to customer service. It relates to the robustness of the system to imminent disturbances and, hence, depends on the system operating condition as well as the probability of disturbances. The functions of this subsystem can be grouped into three classes: (1) Security monitoring, (2) Security Assessment, and (3) Security controls.

Security Monitoring classifies the state of the system as normal or abnormal based on real time measurements. At any time, power system should satisfy two types of requirements: (1) Operating constraints, O, such as limits on system frequency, limits on bus voltage magnitude, limits on circuit loading, etc. These operating constraints are expressed as a set of inequality constraints defined over state and control variables as shown in equation F.8. (2) Load constraints, D, which express the fact that any customer switching into the system must be served. They are represented with a set of equality constraints, i.e., the power flow equations as shown in equation F.9.

$$h(x,u) \leq 0 \tag{F.8}$$

$$g(x,u) = 0 \tag{F.9}$$

In terms of above expressions, the operating states of a power system are classified into: (l) secure, (2) normal but insecure or vulnerable, (3) emergency, (4) extremis, and (5) restorative as follows:

- *Secure*: All load and operating constraints are satisfied for the system and for any foreseeable and probable contingency.
- *Normal but insecure*: All load and operating constraints are satisfied for the present system, but not for one or more foreseeable contingencies.
- *Emergency*: All load constraints are satisfied, but one or more operating constraints are vio-

lated.

- *Extremis*: One or more load constraints are violated, and one or more operating constraints are violated.
- *Restorative*: All operating constraints are satisfied, but one or more loads are disconnected.

Security Assessment is the evaluation of data, provided by security monitoring, to estimate the relative robustness (security level) of the system in its present state ,i.e., determination of whether the system is in the secure or insecure operating state. Security assessment involves evaluating the impacts of unplanned outages on power systems. This process is called *contingency analysis*, where a *contingency* is the loss or failure of a small part of the power system, or the loss/failure of individual equipment such as a generator or transformer. Contingency analysis consists of 3 steps:

- *Contingency Definition*: It involves preparing a list of probable contingencies. N-1 is the standard security criteria, which stipulates that the power system must remain secure in the event of losing any single component of the system, i.e., assuming a system has N components, the system must remain secure even if it operates with N-1 components.
- *Contingency Selection*: This process involves evaluating the risk associated with each contingency by simulating the outage. Usually, fast power flow solution techniques such as DC power flow are used to quickly assess the risks associated with each contingency until no violations of operating constraints are observed.
- *Contingency Ranking*: This process ranks the set of contingencies according to the security criteria such as generation demand imbalance.

Security Controls: The overall objective of system operation is to steer the system in such a way as to operate in a secure state at every instant of time. Occasionally, however, the system deviates from the secure operation. In this case, controls are exercised to return the system operation to a secure state. Depending on the type of insecurity, different controls must be exercised. These controls are characterized as *preventive*, *corrective*, *emergency*, and *restorative*.

- Preventive Controls are actions which bring a normal but insecure operating state to a secure state. Preventive actions consist of generation rescheduling that involves longer time scales. Security-constrained optimal power flow is an example of rescheduling the generations in the system to ensure secure operation.
- Corrective Controls are actions which bring an emergency operating state to a normal state

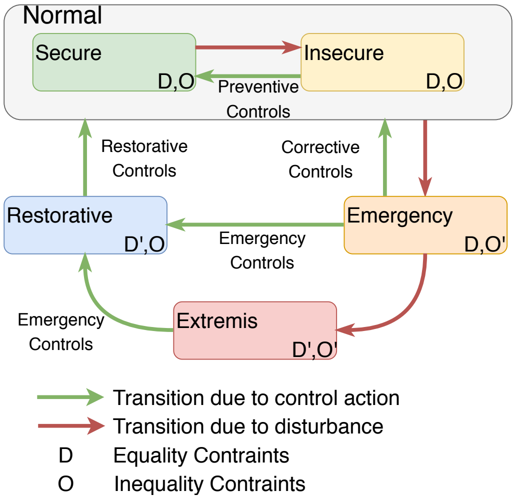


Figure F.4: Power System Operating States

(secure or vulnerable). Corrective actions constitute switching of VAR compensating devices, changing transformer taps, etc., are mainly automatic, and involve short duration.

- Emergency Controls are actions which bring an emergency or extremis operating state to a restorative state. Emergency actions include controlled generator and load shedding, system splitting or islanding.
- Restorative Controls are actions which bring a restorative operating state to a normal state (secure or insecure). Restorative actions include re-synchronization, automatic load transfer and automatic feeder restoration.

A summary of operating states and security controls is illustrated in Figure F.4.

F.0.2.3 Automatic Generation Control Subsystem

This subsystem manages the energy generation, controls the frequency and the power transactions (net interchange) of the system, and optimizes the operation of the system. There is a hierarchical structure within this system. The first layer performs economic dispatch and VAR dispatch based on load forecasting models. The second layer performs generation control based on the references obtained from the first layer.

Economic Dispatch: It is an optimization process that determines the operation of the least-cost available generators, given (a) the total electric demand, and (b) the minimum and maximum operating limits of each generator. Equations eqs. (F.10) to (F.12) present the formulation of the economic dispatch problem as an optimization problem.

$$\min_{P_{G_1},\ldots,P_{G_k}} \quad \sum_{i \in M} c_i \cdot P_{G_i} \tag{F.10}$$

$$\sum_{i \in M} P_{G_i} = P_D \tag{F.11}$$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \qquad \forall i \in M \tag{F.12}$$

The objective function, equation F.10 minimizes the total power generation cost of k generators, where set $M = \{m_1, \ldots, m_k\}$ is the set of controllable generators, c_i is the marginal cost of i^{th} generator, $m_i \in M$, and P_{G_i} is the amount of power it generates. Equation F.12 requires that all generators must not violate their minimum or maximum limits, while equation F.11 stipulates that all generated power must be equal to the electricity demand, P_D. The economic dispatch has some drawbacks because of the underlying assumption of the loss-less and unrestricted flow of electricity from point A to point B. This means that it neglects all network constraints, including transmission line limits, line congestion, and transmission losses.

Optimal Power Flow (OPF) is an extension of economic dispatch. Its objective is to minimize the total cost of electricity generation while maintaining the electric power system within safe operating limits. The power system is modeled as a set of **N** buses connected by a set of **L** branches. Controllable generators are located at $\mathbf{M} \subseteq \mathbf{N}$ system buses. The operating cost of each generator is a (typically quadratic) function of its real output power: $c_i \cdot P_{G_i}$. Equations (F.13) to (F.20) present formulation of optimal power flow as an optimization problem

$$\min_{P_{G_1},\ldots,P_{G_k}} \quad \sum_{i\in M} c_i \cdot P_{G_i} \tag{F.13}$$

$$P_{G_i} - P_{D_i} = V_i \sum_{j\in N} V_j \big(G_{ij}\cos\big(\delta_i - \delta_j\big) + B_{ij}\sin\big(\delta_i - \delta_j\big)\big) \qquad \forall i \in M, \tag{F.14}$$

$$Q_{G_i} - Q_{D_i} = V_i \sum_{j\in N} V_j \big(G_{ij}\sin\big(\delta_i - \delta_j\big) - B_{ij}\cos\big(\delta_i - \delta_j\big)\big) \qquad \forall i \in M, \tag{F.15}$$

$$P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max} \qquad \forall i \in M, \tag{F.16}$$

$$Q_{G_i}^{\min} \leq Q_{G_i} \leq Q_{G_i}^{\max} \qquad \forall i \in M, \tag{F.17}$$

$$V_i^{\min} \leq Vi \leq V_i^{\max} \qquad \forall i \in N, \tag{F.18}$$

$$\delta_i^{\min} \leq \delta i \leq \delta_i^{\max} \qquad \forall i \in N, \tag{F.19}$$

$$|V_i - V_j| y_{ij} \leq I_{ij}^{\max} \qquad \forall i, j \in L \tag{F.20}$$

Equations (F.14) and (F.15) are the only equality constraints in the optimal power flow formulation, and they denote the AC power flow equations. Equation (F.16) refer to the active power bounds of the generators, while eq. (F.17) depicts the generators' reactive power bounds. Equation (F.18) denotes the maximum and minimum allowable limits for the bus voltage magnitudes (no complex numbers here), and eq. (F.19) refer to maximum and minimum allowable limits for the voltage angles. Equation (F.20) represents the constraints imposed on the branch current due to thermal limit of the conductor.

The AC power flow equations are non-linear which results in a non-linear non-convex problem. Non-convex problems are in general much harder to solve, and there is no guarantee that the solver can find the global minimum. The usual procedure is to linearize the AC power flow equations by making two assumptions (1) Bus voltages remain constant at nominal value (2) Voltage angle differences are small[2]. The resulting power flow equations are called DC power flow and corresponding optimal power flow is referred to as DC optimal power flow.

The main limitation of Optimal power flow formulation is that it focuses on the optimization of a single system configuration at the time while the system operator needs to know: (i) how robust the system is with respect to various credible contingencies and (ii) how to meet operating constraints

[2] The assumptions of constant voltage and small angle differences are appropriate for lightly loaded systems

for probable contingencies. The first requirement can be tackled by performing security analysis at the optimal power flow solution. However, the second requirement led to the formulation of the Security Constrained Optimal Power Flow (SCOPF) problem as a natural extension of the optimal power flow which takes into account pre-contingency (base case) constraints and also (steady-state) post-contingency constraints together.

SCOPF also referred to as Security Constrained Economic Dispatch (SCED). It seeks an optimal solution that remains feasible under any of the pre-specified set of likely contingency events. SCOPF formulations typically have the same objective function and decision variables as the classic formulation OPF formulation. However, they introduce N_c additional sets of state variables x and accompanying sets of power flow constraints, where N_c is the number of contingencies. SCOPF can be expressed in a general way as shown in eqs. (F.21) to (F.23)

$$\min_{x_0,\ldots,x_c,u_0} f\left(x_0, u_0\right) \tag{F.21}$$

$$g_k\left(x_k, u_0\right) = 0 \qquad k = 0, \ldots, c \tag{F.22}$$

$$h_k\left(x_k, u_0\right) \leq 0 \qquad k = 0, \ldots, c \tag{F.23}$$

where f is a (real-valued) function representing the objective to optimize, g_k, h_k are the sets of equality and inequality constraints for the k-th system configuration (k = 0 corresponds to the base case, while k = $1, \ldots, c$ corresponds to the k^{th} post-contingency state, c being the number of contingencies considered), x_k is the vector of state variables (i.e., real and imaginary part of voltage at all buses) for the k-th system topology and u_0 is the vector of base case control/decision variables (e.g., active and reactive generator powers, controllable transformer ratio, shunt element reactance, load apparent power, etc.).

VAR Dispatch: VAR dispatch or control seeks to optimize the system reactive power generation in order to minimize the total system losses. It is also known as *Optimal Reactive Power Flow* (ORPF). In ORPF, the system real power generation is determined a priori, from the outcome of ED. A basic ORPF formulation is illustrated by eqs. (F.24) to (F.31)

$$\min_{Q_{G_1},\ldots,Q_{G_k}} \quad P_1 \tag{F.24}$$

$$P_{G_i} - P_{D_i} = V_i \sum_{j \in N} V_j \big(G_{ij} \cos\big(\delta_i - \delta_j\big) + B_{ij} \sin\big(\delta_i - \delta_j\big)\big) \qquad \forall i \in M, \tag{F.25}$$

$$Q_{G_i} - Q_{D_i} = V_i \sum_{j \in N} V_j \big(G_{ij} \sin\big(\delta_i - \delta_j\big) - B_{ij} \cos\big(\delta_i - \delta_j\big)\big) \qquad \forall i \in M, \tag{F.26}$$

$$P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max} \qquad \forall i \in G, \tag{F.27}$$

$$Q_{G_i}^{\min} \leq Q_{G_i} \leq Q_{G_i}^{\max} \qquad \forall i \in G, \tag{F.28}$$

$$V_i^{\min} \leq Vi \leq V_i^{\max} \qquad \forall i \in N, \tag{F.29}$$

$$\delta_i^{\min} \leq \delta i \leq \delta_i^{\max} \qquad \forall i \in N, \tag{F.30}$$

$$|V_i - V_j| y_{ij} \leq I_{ij}^{\max} \qquad \forall i, j \in L \tag{F.31}$$

The formulation of OPRF is identical to the classical OPF except the number control variables and objective function. In ORPF, all real power load and generation is fixed except for the real power, P_1, at one bus called the slack bus. Minimizing P_1 is therefore equivalent to minimizing total system losses.

Generation Control: Generation control system is a feedback control system that regulates power output (real) of generators, maintain voltage level at generator terminals, stabilizes oscillations and maintain scheduled power interchange between control areas. The set-points or references are obtained through online optimal power flow or security constrained economic dispatch running on computers in control centers. Various control loops track these set points in near real time (1-5 minutes) to regulate frequency, voltage and real power of generators as shown in Figure F.5. These control loops are briefly described as follows:

1. *Primary Automatic Generator Control Loop*: The objective of this control loop is to regulate the real power output and the speed of the generator. It consists of the speed regulator (governor) of the prime mover. It uses the feedback of the generator speed (or frequency) and the real power output of the generator.
2. *Secondary Automatic Generation Control Loop*: The objective of the secondary automatic generation control loop is to regulate the net interchange, unit real power output, and speed

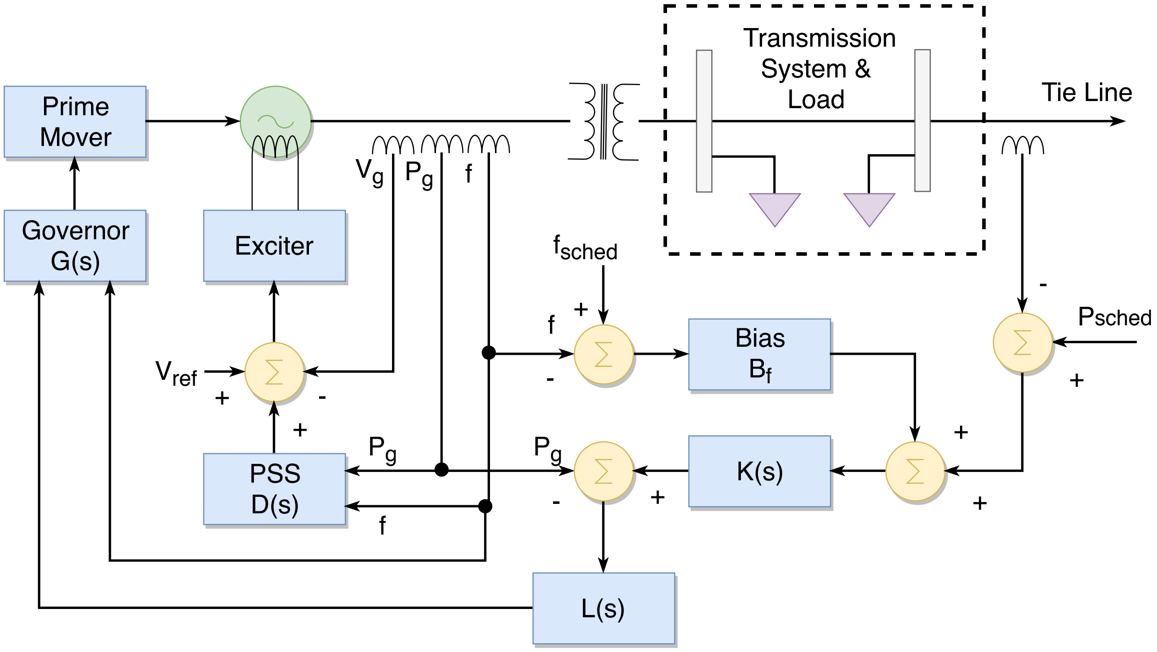


Figure F.5: Schematic Representation of Control Schemes for a Generating Unit

(frequency). It consists of a feedback system which injects a signal into the speed regulator (governor). The signal, referred to as the Unit Control Error (UCE), is constructed from measurements of frequency, interchange schedule, unit real power output, etc. Reference quantities for this control loop are: (a) Scheduled interchange of real power, P_{sched}, (b) Scheduled frequency, f_{sched}. This control loop uses integral feedback of frequency and therefore regulates the system real time.

3. *Power System Stabilizer* (PSS) *Loop*: The objective of this control loop is to slow down the oscillations of the generator following a disturbance. It consists of a feedback system which injects a stabilizing signal into the exciter system. Feedback quantities: frequency, f, real power, P_g.

4. *Voltage Control Loop*: The objective of this control loop is to regulate the voltage at the terminals of the generator. It consists of the voltage regulator and exciter system. Inputs to this control loop are the reference voltage, V_{ref}, which may be selected by the system dispatcher or automatically by computers (VAR dispatch), and the actual voltage at the terminals of the generator, V_g.

Appendix G

TCTL Grammar

The language of well-formed formulas for TCTL is generated by the following grammar:

$$\phi ::= \bot \,|\, \top \,|\, p \,|\, \textbf{not}\phi \,|\, \phi\,\textbf{or}\,\phi \,|\, \phi\,\textbf{and}\,\phi \,|\, \phi\,\textbf{imply}\,\phi \,|\, \textbf{A[ ]}\,\phi \,|\, \textbf{A<>}\,\phi \,|\, \textbf{E[ ]}\,\phi \,|\, \textbf{E<>} \,|\, \phi \, - - > \, \phi \quad \text{(G.1)}$$

where p is a set of atomic propositions defined over state labels, discrete and clock variables, **not**, **or**, **and**, **imply** are logical operators and **A[ ]**, **A** $<>$, **E[ ]**, **E** $<>$, $- - >$ are temporal operators. A temporal operator is a combination of path and state operators defined as follows:

- **E** $<> p$: p is true in at least one state reachable state along any path.
- **E** [ ] p : There exists a path in which p is true in all states.
- **A** $<> p$: p will eventually become true in some state along all paths.
- **A** [ ] p : p is true in all reachable states.
- $p - - > q$: In all paths, if p becomes true then q will eventually becomes true. This operator is equivalent to **A [ ]** (p **imply** (**A** $<>$ q)).

BIBLIOGRAPHY

[1] G. Benmouyal, M. Meisinger, J. Burnworth, W.A. Elmore, K. Freirich, P.A. Kotos, P.R. Leblanc, P.J. Lerley, J.E. McConnell, J. Mizener, J. Pinto de Sa, R. Ramaswami, M.S. Sachdev, W.M. Strang, J.E. Waldron, S. Watansiriroch, and S.E. Zocholl. IEEE standard inverse-time characteristic equations for overcurrent relays. *IEEE Transactions on Power Delivery*, 14(3):868–872, 7 1999.

[2] North American Electric Reliability Corporation. Interconnections, 2018. https://www.nerc.com/AboutNERC/keyplayers/PublishingImages/Interconnections%2024JUL18.jpg.

[3] Edward A Lee. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors*, 15(3):4837–4869, 2015.

[4] Rajeev Alur. *Principles of Cyber-Physical Systems - Rajeev Alur - Google Books*.

[5] Peter Palensky, Edmund Widl, and Atiyah Elsheikh. Simulating Cyber-Physical Energy Systems: Challenges, Tools and Methods. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(3):318–326, 3 2014.

[6] Vasso Reppa, Marios M. Polycarpou, and Christos G. Panayiotou. Distributed sensor fault diagnosis for a network of interconnected cyberphysical systems. *IEEE Transactions on Control of Network Systems*, 2(1):11–23, 3 2015.

[7] North American Electric Reliability Corporation. https://www.nerc.com/Pages/default.aspx.

[8] Major Event Analysis Reports. Available at https://www.nerc.com/pa/rrm/ea/Pages/Major-Event-Reports.aspx.

[9] J. Duncan Glover, S. Sarma Mulukutla, and Thomas Overbye. Power System Analysis &amp; Design, SI Version - Google Books.

[10] High-Speed Line Protection, Automation, and Control System Major Features and Benefits SEL-421 Protection and Automation System. Technical report.

[11] Mirrored Bits Communications. Technical report.

[12] Distributed Network Protocol.

[13] MODBUS Application Protocol Specification V1.1b3 Modbus. Technical report, 2012.

[14] Stanley H. Horowitz and Arun G. Phadke. *Power system relaying*. Wiley/Research Studies Press, 2008.

[15] High Voltage Products. Technical report.

[16] SEL. SEL-421-4,-5 Protection and Automation System. Technical report.

[17] SEL. SEL-700G Family of Generator Protection Relays. Technical report.

[18] Introduction to Misoperation Information Data Analysis System (MIDAS). Technical report.

[19] NERC State of Reliability 2018. Technical report, 2018.

[20] Protection System Misoperations. https://www.nerc.com/pa/RAPA/Pages/Misoperations.aspx.

[21] NERC State of Reliability. Technical report, 2018. https://www.nerc.com/pa/RAPA/PA/Performance{%}20Analysis{%}20DL/NERC_2018_SOR_06202018_Final.pdf.

[22] Edmund O Schweitzer III and Jeff Roberts. Distance relay element design. In *proceedings of the 46th Annual Conference for Protective Relay Engineers, College Station, TX*, 1993.

[23] Hank Miller, John Burger, Normann Fischer, and Bogdan Kasztenny. Modern line current differential protection solutions. In *2010 63rd Annual Conference for Protective Relay Engineers*, pages 1–25. IEEE, 3 2010.

[24] Mike McDonald Chairman Demetrios Tziouvaras Vice Chairman Apostolov, Alex Benmouyal, Gabriel Brunello, Gustavo Buanno, Art Darlington, Al Elmore, Walt Fink, Charlie Holbach, Juergen Horton, Randy Johnson, Gerald Kemp, Peter Kennedy, Bill Kim, Chul-Hwan Khan, Shoukat Lowe, Bill Kobet, Gary Nagpal, Mukesh Plumptre, Frank Schroeder, and Mark Turner. POWER SWING AND OUT-OF-STEP CONSIDERATIONS ON TRANSMISSION LINES. Technical report, IEEE PSRC WG D6, 2005.

[25] P.M. Anderson and M. Mirheydar. An adaptive method for setting underfrequency load shedding relays. *IEEE Transactions on Power Systems*, 7(2):647–655, 5 1992.

[26] Institute of Electrical and Electronics Engineers., IEEE Industry Applications Society. Industrial and Commercial Power Systems Department., IEEE Standards Board., and American National Standards Institute. *IEEE recommended practice for protection and coordination of industrial and commercial power systems*. Institute of Electrical and Electronics Engineers, 2001.

[27] U.S.-Canada Power System Outage Task Force. Causes of the August 14th Blackout in the United States and Canada. Technical report, NERC, 2003.

[28] Jie Chen and Ron J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems - Jie Chen, R.J. Patton - Google Books*. SPRINGER SCIENCE+BUSINESS MEDIA, LLC, 1 edition, 1999.

[29] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. Smart Grid The New and Improved Power Grid: A Survey. *IEEE Communications Surveys & Tutorials*, 14(4):944–980, 24 2012.

[30] V.H. Ferreira, R. Zanghi, M.Z. Fortes, G.G. Sotelo, R.B.M. Silva, J.C.S. Souza, C.H.C. Guimarães, and S. Gomes. A survey on intelligent system application to fault diagnosis in electric power system transmission lines. *Electric Power Systems Research*, 136:135–153, 7 2016.

[31] Abhisek Ukil and Rastko ivanovi. Application of Abrupt Change Detection in Power Systems Disturbance Analysis and Relay Performance Monitoring. *IEEE Transactions on Power Delivery*, 22(1):59–66, 1 2007.

[32] A.A. Girgis and M.B. Johns. A hybrid expert system for faulted section identification, fault type classification and selection of fault location algorithms. *IEEE Transactions on Power Delivery*, 4(2):978–985, 4 1989.

[33] Rujiroj Leelaruji and Luigi Vanfretti. State-of-the-art in the industrial implementation of protective relay functions, communication mechanism and synchronized phasor capabilities for

electric power systems protection. *Renewable and Sustainable Energy Reviews*, 16(7):4385–4395, 9 2012.

[34] Chen-Ching Liu and Tharam Dillon. State-of-the-art of expert system applications to power systems. *International Journal of Electrical Power & Energy Systems*, 14(2-3):86–96, 4 1992.

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

[36] B. Jeyasurya, S.S. Venkata, S.V. Vadari, and J. Postforoosh. Fault diagnosis using substation computer. *IEEE Transactions on Power Delivery*, 5(2):1195–1201, 4 1990.

[37] A. Hertz and P. Fauquembergue. Fault diagnosis at substations based on sequential event recorders. *Proceedings of the IEEE*, 80(5):684–688, 5 1992.

[38] J.R. McDonald, G.M. Burt, and D.J. Young. Alarm processing and fault diagnosis using knowledge based systems for transmission and distribution network control. *IEEE Transactions on Power Systems*, 7(3):1292–1298, 1992.

[39] M. Kezunovic, P. Spasojevic, C.W. Fromen, and D.R. Sevcik. An expert system for transmission substation event analysis. *IEEE Transactions on Power Delivery*, 8(4):1942–1949, 1993.

[40] Zhu Yongli, Y.H. Yang, B.W. Hogg, W.Q. Zhang, and S. Gao. An expert system for power systems fault analysis. *IEEE Transactions on Power Systems*, 9(1):503–509, 1994.

[41] M.E. Vazquez, L. Oscar, M. Chacon, J. Hector, and F. Altuve. A knowledge-based system for on-line diagnosis of power system fault allocation. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1148–1153. IEEE.

[42] Hanjin Miao, M. Sforna, and Chen-Ching Liu. A new logic-based alarm analyzer for on-line operational environment. *IEEE Transactions on Power Systems*, 11(3):1600–1606, 1996.

[43] C.S. Chang and S.G. Kerk. Fault signal filtering for improving fault section estimation.

In *2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.00CH37077)*, volume 4, pages 2545–2550. IEEE.

[44] Juhwan Jung, Chen-Ching Liu, Mingguo Hong, M. Gallanti, and G. Tornielli. Multiple hypotheses and their credibility in on-line fault diagnosis. *IEEE Transactions on Power Delivery*, 16(2):225–230, 4 2001.

[45] Ching-Lai Hor, Peter A. Crossley, and Simon J. Watson. Building Knowledge for Substation-Based Decision Support Using Rough Sets. *IEEE Transactions on Power Delivery*, 22(3):1372–1379, 7 2007.

[46] T.S. Sidhu, O. Cruder, and G.J. Huff. An abductive inference technique for fault diagnosis in electrical power transmission networks. *IEEE Transactions on Power Delivery*, 12(1):515–522, 1997.

[47] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 6 1965.

[48] ME El-Hawary. Fuzzy system theory in electrical power engineering, 1998.

[49] R.J. Sárfi, M.M.A. Salama, and A.Y. Chikhani. Applications of fuzzy sets theory in power systems planning and operation: a critical review to assist in implementation. *Electric Power Systems Research*, 39(2):89–101, 11 1996.

[50] A. Ferrero, S. Sangiovanni, and E. Zappitelli. A fuzzy-set approach to fault-type identification in digital relaying. *IEEE Transactions on Power Delivery*, 10(1):169–175, 1995.

[51] Ngoc Tran-Huynh and N Hoonchareon. Generalized sagittal diagram for fault equipment identification within a transmission system. In *TENCON 2010 - 2010 IEEE Region 10 Conference*, pages 1266–1271. IEEE, 11 2010.

[52] R.N. Mahanty and P.B. Dutta Gupta. A fuzzy logic based fault classification approach using current samples only. *Electric Power Systems Research*, 77(5-6):501–507, 4 2007.

[53] C. Aguilera, E. Orduna, and G. Ratta. Fault detection, classification and faulted phase selection approach based on high-frequency voltage signals applied to a series-compensated line. *IEE Proceedings - Generation, Transmission and Distribution*, 153(4):469, 2006.

[54] Doaa khalil Ibrahim, El Sayed Tag Eldin, Essam M. Aboul-Zahab, and Saber Mohamed Saleh. Real time evaluation of DWT-based high impedance fault detection in EHV transmission. *Electric Power Systems Research*, 80(8):907–914, 8 2010.

[55] D. Chanda, N.K. Kishore, and A.K. Sinha. Application of wavelet multiresolution analysis for identification and classification of faults on transmission lines. *Electric Power Systems Research*, 73(3):323–333, 3 2005.

[56] D. Chanda, N.K. Kishore, and A.K. Sinha. A wavelet multiresolution analysis for location of faults on transmission lines. *International Journal of Electrical Power & Energy Systems*, 25(1):59–69, 1 2003.

[57] J. Liang, S. Elangovan, and J.B.X. Devotta. A wavelet multiresolution analysis approach to fault detection and classification in transmission lines. *International Journal of Electrical Power & Energy Systems*, 20(5):327–332, 6 1998.

[58] Chul-Hwan Kim, Hyun Kim, Young-Hun Ko, Sung-Hyun Byun, R.K. Aggarwal, and A.T. Johns. A novel fault-detection technique of high-impedance arcing faults in transmission lines using the wavelet transform. *IEEE Transactions on Power Delivery*, 17(4):921–929, 10 2002.

[59] Zhengyou He, Ling Fu, Sheng Lin, and Zhiqian Bo. Fault Detection and Classification in EHV Transmission Line Based on Wavelet Singular Entropy. *IEEE Transactions on Power Delivery*, 25(4):2156–2163, 10 2010.

[60] M. García-Gracia, A. Montañés, N. El Halabi, and M.P. Comech. High resistive zero-crossing instant faults detection and location scheme based on wavelet analysis. *Electric Power Systems Research*, 92:138–144, 11 2012.

[61] Hosung Jung, Young Park, Moonseob Han, Changmu Lee, Hyunjune Park, and Myongchul Shin. Novel technique for fault location estimation on parallel transmission lines using wavelet. *International Journal of Electrical Power & Energy Systems*, 29(1):76–82, 1 2007.

[62] A.M. El-Zonkoly and H. Desouki. Wavelet entropy based algorithm for fault detection and

classification in FACTS compensated transmission line. *International Journal of Electrical Power & Energy Systems*, 33(8):1368–1374, 10 2011.

[63] S El Safty and A El-Zonkoly. Applying wavelet entropy principle in fault classification. *International Journal of Electrical Power & Energy Systems*, 31(10):604–607, 2009.

[64] F.E. Prez, R. Aguilar, E. Ordua, J. Jger, and G. Guidi. High-speed non-unit transmission line protection using single-phase measurements and an adaptive wavelet: zone detection and fault classification. *IET Generation, Transmission & Distribution*, 6(7):593, 2012.

[65] J. Upendar, C.P. Gupta, and G.K. Singh. Statistical decision-tree based fault classification scheme for protection of power transmission lines. *International Journal of Electrical Power & Energy Systems*, 36(1):1–12, 3 2012.

[66] A Abur and F.H Magnago. Use of time delays between modal components in wavelet based fault location. *International Journal of Electrical Power & Energy Systems*, 22(6):397–403, 8 2000.

[67] E.E. Ngu and K. Ramar. A combined impedance and traveling wave based fault location method for multi-terminal transmission lines. *International Journal of Electrical Power & Energy Systems*, 33(10):1767–1775, 12 2011.

[68] D.M. Gilbert and I.F. Morrison. A statistical method for the detection of power system faults. *International Journal of Electrical Power & Energy Systems*, 19(4):269–275, 5 1997.

[69] N.G. Tarhuni, N.I. Elkalashy, and M. Lehtonen. Simplified probabilistic selectivity technique for earth fault detection in unearthed MV networks. *IET Generation, Transmission & Distribution*, 3(2):145–153, 2 2009.

[70] F.N. Chowdhury, J.P. Christensen, and J.L. Aravena. Power system fault detection and state estimation using Kalman filter with hypothesis testing. *IEEE Transactions on Power Delivery*, 6(3):1025–1030, 7 1991.

[71] J. Barros. Realtime fault detection and classification in power systems using microprocessors. *IEE Proceedings - Generation, Transmission and Distribution*, 141(4):315, 1994.

[72] Vitor Hugo Ferreira and Alexandre P. Alves da Silva. Toward Estimating Autonomous Neural Network-Based Electric Load Forecasters. *IEEE Transactions on Power Systems*, 22(4):1554–1562, 11 2007.

[73] Alexandre P. Alves da Silva, Vitor H. Ferreira, and Roberto M.G. Velasquez. Input space to neural network based load forecasters. *International Journal of Forecasting*, 24(4):616–629, 10 2008.

[74] Vitor Hugo Ferreira and Alexandre Pinto Alves da Silva. Chaos theory applied to input space representation of autonomous neural network-based short-term load forecasting models. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 22(6):585–597, 12 2011.

[75] Vitor Hugo Ferreira and Alexandre P. Alves da Silva. Autonomous Kernel Based Models for Short-Term Load Forecasting, 2012.

[76] Vidya Sagar S. Vankayala and Nutakki D. Rao. Artificial neural networks and their applications to power systemsa bibliographical survey. *Electric Power Systems Research*, 28(1):67–79, 10 1993.

[77] Whei-Min Lin, Chin-Der Yang, Jia-Hong Lin, and Ming-Tong Tsay. A fault classification method by RBF neural network with OLS learning procedure. *IEEE Transactions on Power Delivery*, 16(4):473–477, 2001.

[78] T. Dalstein and B. Kulicke. Neural network approach to fault classification for high speed protective relaying. *IEEE Transactions on Power Delivery*, 10(2):1002–1011, 4 1995.

[79] R.N. Mahanty and P.B. Dutta Gupta. Application of RBF neural network to fault classification and location in transmission lines. *IEE Proceedings - Generation, Transmission and Distribution*, 151(2):201, 2004.

[80] Z. Chen and J.-C. Maun. Artificial neural network approach to single-ended fault locator for transmission lines. *IEEE Transactions on Power Systems*, 15(1):370–375, 2000.

[81] A.J. Mazon, I. Zamora, J.F. Miñambres, M.A. Zorrozua, J.J. Barandiaran, and K. Sagastabeitia. A new approach to fault location in two-terminal transmission lines using artificial neural networks. *Electric Power Systems Research*, 56(3):261–266, 12 2000.

[82] G. Cardoso, J.G. Rolim, and H.H. Zurn. Application of Neural-Network Modules to Electric Power System Fault Section Estimation. *IEEE Transactions on Power Delivery*, 19(3):1034–1041, 7 2004.

[83] He Zhengyou, Gao Shibin, Chen Xiaoqin, Zhang Jun, Bo Zhiqian, and Qian Qingquan. Study of a new method for power system transients classification based on wavelet entropy and neural network. *International Journal of Electrical Power & Energy Systems*, 33(3):402–410, 3 2011.

[84] Sami Ekici, Selcuk Yildirim, and Mustafa Poyraz. Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition. *Expert Systems with Applications*, 34(4):2937–2944, 5 2008.

[85] Damitha K. Ranaweera. Comparison of neural network models for fault diagnosis of power systems. *Electric Power Systems Research*, 29(2):99–104, 3 1994.

[86] Jefferson Morais, Yomara Pires, Claudomir Cardoso, and Aldebaro Klautau. A Framework for Evaluating Automatic Classification of Underlying Causes of Disturbances and Its Application to Short-Circuit Faults. *IEEE Transactions on Power Delivery*, 25(4):2083–2094, 10 2010.

[87] Y.H. Song, Q.X. Xuan, and A.T. Johns. Comparison studies of five neural network based fault classifiers for complex transmission lines. *Electric Power Systems Research*, 43(2):125–132, 11 1997.

[88] G.K Purushothama, A.U Narendranath, D Thukaram, and K Parthasarathy. ANN applications in fault locators. *International Journal of Electrical Power & Energy Systems*, 23(6):491–506, 8 2001.

[89] A. de Souza Gomes, M. A. Costa, T. G. A. de Faria, and W. M. Caminhas. Detection and Classification of Faults in Power Transmission Lines Using Functional Analysis and Computational Intelligence. *IEEE Transactions on Power Delivery*, 28(3):1402–1413, 7 2013.

[90] Paula Renatha N. da Silva, Martin Max L.C. Negrão, Petrnio Vieira, and Miguel A. Sanz-

Bobi. A new methodology of fault location for predictive maintenance of transmission lines. *International Journal of Electrical Power & Energy Systems*, 42(1):568–574, 11 2012.

[91] S.R. Samantaray and P.K. Dash. Pattern recognition based digital relaying for advanced series compensated line. *International Journal of Electrical Power & Energy Systems*, 30(2):102–112, 2 2008.

[92] A.E. Lazzaretti, V.H. Ferreira, H.V. Neto, R.J. Riella, and J. Omori. Classification of Events in Distribution Networks using Autonomous Neural Models. In *2009 15th International Conference on Intelligent System Applications to Power Systems*, pages 1–6. IEEE, 11 2009.

[93] Urmil B. Parikh, Biswarup Das, and Rudraprakash Maheshwari. Fault classification technique for series compensated transmission line using support vector machine. *International Journal of Electrical Power & Energy Systems*, 32(6):629–636, 7 2010.

[94] B. Ravikumar, D. Thukaram, and H.P. Khincha. Application of support vector machines for fault diagnosis in power transmission system. *IET Generation, Transmission & Distribution*, 2(1):119, 2008.

[95] Electromagnetic Transient Program. https://etap.com/product/electromagnetic-transient-program.

[96] Hanif Livani and Cansin Yaman Evrenosoglu. A Fault Classification and Localization Method for Three-Terminal Circuits Using Machine Learning. *IEEE Transactions on Power Delivery*, 28(4):2282–2290, 10 2013.

[97] V. Malathi, N.S. Marimuthu, and S. Baskar. Intelligent approaches using support vector machine and extreme learning machine for transmission line protection. *Neurocomputing*, 73(10-12):2160–2167, 6 2010.

[98] R. Salat and S. Osowski. Accurate Fault Location in the Power Transmission Line Using Support Vector Machine Approach. *IEEE Transactions on Power Systems*, 19(2):979–986, 5 2004.

[99] U.B. Parikh, Biswarup Das, and R.P. Prakash Maheshwari. Combined Wavelet-SVM Technique for Fault Zone Detection in a Series Compensated Transmission Line. *IEEE Transactions on Power Delivery*, 23(4):1789–1794, 10 2008.

[100] Sami Ekici. Support Vector Machines for classification and locating faults on transmission lines. *Applied Soft Computing*, 12(6):1650–1658, 6 2012.

[101] A.A. Yusuff, C. Fei, A.A. Jimoh, and J.L. Munda. Fault location in a series compensated transmission line based on wavelet packet decomposition and support vector regression. *Electric Power Systems Research*, 81(7):1258–1265, 7 2011.

[102] S.R. Samantaray, P.K. Dash, and G. Panda. Distance relaying for transmission line using support vector machine and radial basis function neural network. *International Journal of Electrical Power & Energy Systems*, 29(7):551–556, 9 2007.

[103] A.A. Yusuff, A.A. Jimoh, and J.L. Munda. Determinant-based feature extraction for fault detection and classification for power transmission lines. *IET Generation, Transmission & Distribution*, 5(12):1259, 2011.

[104] J.A. Morales, E. Orduña, and C. Rehtanz. Classification of lightning stroke on transmission line using multi-resolution analysis and machine learning. *International Journal of Electrical Power & Energy Systems*, 58:19–31, 6 2014.

[105] Nan Zhang and Mladen Kezunovic. Transmission Line Boundary Protection Using Wavelet Transform and Neural Network. *IEEE Transactions on Power Delivery*, 22(2):859–869, 4 2007.

[106] Carlos Eduardo de Morais Pereira and Luiz Cera Zanetta. An optimisation approach for fault location in transmission lines using one terminal data. *International Journal of Electrical Power & Energy Systems*, 29(4):290–296, 5 2007.

[107] Simulink - Simulation and Model-Based Design. https://www.mathworks.com/products/simulink.html.

[108] Fushuan Wen and Zhenxiang Han. Fault section estimation in power systems using a genetic algorithm. *Electric Power Systems Research*, 34(3):165–172, 9 1995.

[109] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R.G. Harley. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Transactions on Evolutionary Computation*, 12(2):171–195, 4 2008.

[110] M.R. AlRashidi and M.E. El-Hawary. A Survey of Particle Swarm Optimization Applications in Electric Power Systems. *IEEE Transactions on Evolutionary Computation*, 13(4):913–918, 8 2009.

[111] Xiangning Lin, Shuohao Ke, Zhengtian Li, Hanli Weng, and Xionghui Han. A Fault Diagnosis Method of Power Systems Based on Improved Objective Function and Genetic Algorithm-Tabu Search. *IEEE Transactions on Power Delivery*, 25(3):1268–1274, 7 2010.

[112] Carlos Eduardo de Morais Pereira and Luiz Cera Zanetta. An optimisation approach for fault location in transmission lines using one terminal data. *International Journal of Electrical Power & Energy Systems*, 29(4):290–296, 5 2007.

[113] L. Wei, W. Guo, F. Wen, G. Ledwich, Z. Liao, and J. Xin. Waveform matching approach for fault diagnosis of a high-voltage transmission line employing harmony search algorithm. *IET Generation, Transmission & Distribution*, 4(7):801, 2010.

[114] Shanshan Luo, Mladen Kezunovic, and Don R. Sevick. Locating faults in the transmission network using sparse field measurements, simulation data and genetic algorithm. *Electric Power Systems Research*, 71(2):169–177, 10 2004.

[115] Ying-Xin Wu, Xiang-ning Lin, Shi-hong Miao, Pei Liu, Dong-qing Wang, and De-bin Chen. Application of Family Eugenics Based Evolution Algorithms to Electric Power System Fault Section Estimation. In *2005 IEEE/PES Transmission &amp; Distribution Conference &amp; Exposition: Asia and Pacific*, pages 1–5. IEEE.

[116] Wenxin Guo, Fushuan Wen, Gerard Ledwich, Zhiwei Liao, Xiangzhen He, and Jiansheng Huang. A new analytic approach for power system fault diagnosis employing the temporal information of alarm messages. *International Journal of Electrical Power & Energy Systems*, 43(1):1204–1212, 12 2012.

[117] Wenxin Guo, Fushuan Wen, Gerard Ledwich, Zhiwei Liao, Xiangzhen He, and Junhui Liang. 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, 7 2010.

[118] Bo Hu, Jinhua She, and Ryuichi Yokoyama. Hierarchical Fault Diagnosis for Power Systems Based on Equivalent-Input-Disturbance Approach. *IEEE Transactions on Industrial Electronics*, 60(8):3529–3538, 8 2013.

[119] J. Shiozaki, B. Shibata, H. Matsuyama, and E. O'shima. Fault diagnosis of chemical processes utilizing signed directed graphs-improvement by using temporal information. *IEEE Transactions on Industrial Electronics*, 36(4):469–474, 1989.

[120] Anibal Bregon, Belarmino Pulido, Gautam Biswas, and Xenofon Koutsoukos. GENERATING POSSIBLE CONFLICTS FROM BOND GRAPHS USING TEMPORAL CAUSAL GRAPHS. Technical report.

[121] Yiannis Papadopoulos. Model-based system monitoring and diagnosis of failures using statecharts and fault trees. *Reliability Engineering & System Safety*, 81(3):325–341, 9 2003.

[122] Z. Yongli, H. Limin, and L. Jinling. Bayesian Networks-Based Approach for Power Systems Fault Diagnosis. *IEEE Transactions on Power Delivery*, 21(2):634–639, 4 2006.

[123] Y Sekine, H Okamoto, and T Shibamoto. Fault section estimation using cause-effect network. In *2nd Symposium Expert System Application to Power Systems Conference*, pages 277–282, 1989.

[124] Wen-Hui Chen, Chih-Wen Liu, and Men-Shen Tsai. Fault diagnosis in distribution substations using CE-nets via Boolean rule matrix transformations. In *2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134)*, volume 1, pages 416–420. IEEE.

[125] W. H. Chen, C. W. Liu, and M. S. Tsai. Fast Fault Section Estimation in Distribution Substations Using Matrix-Based Cause-Effect Networks. *IEEE Power Engineering Review*, 21(8):61–61, 8 2001.

[126] Wen-Hui Chen, Chih-Wen Liu, and Men-Shen Tsai. On-line fault diagnosis of distribution substations using hybrid cause-effect network and fuzzy rule-based method. *IEEE Transactions on Power Delivery*, 15(2):710–717, 4 2000.

[127] Wen-Hui Chen. Fault Section Estimation Using Fuzzy Matrix-Based Reasoning Methods. *IEEE Transactions on Power Delivery*, 26(1):205–213, 1 2011.

[128] Wen-Hui Chen, Shun-Hung Tsai, and Hsien-I. Lin. Fault Section Estimation for Power Networks Using Logic Cause-Effect Models. *IEEE Transactions on Power Delivery*, 26(2):963–971, 4 2011.

[129] Wen-Hui Chen. Online Fault Diagnosis for Power Transmission Networks Using Fuzzy Digraph Models. *IEEE Transactions on Power Delivery*, 27(2):688–698, 4 2012.

[130] K.L. Lo, H.S. Ng, and J. Trecat. Power systems fault diagnosis using Petri nets. *IEE Proceedings - Generation, Transmission and Distribution*, 144(3):231, 1997.

[131] Chun-Ling Yang, Akmiko Yokoyama, and Yasuji Sekine. Fault Section Estimation of Power System Using Colored and Timed Petri Nets. *Electrical Engineering in Japan*, 115(2):89–101, 9 2007.

[132] J. Sun, S.-Y. Qin, and Y.H. Song. Fault Diagnosis of Electric Power Systems Based on Fuzzy Petri Nets. *IEEE Transactions on Power Systems*, 19(4):2053–2059, 11 2004.

[133] Xu Luo and Mladen Kezunovic. Implementing Fuzzy Reasoning Petri-Nets for Fault Section Estimation. *undefined*, 2008.

[134] Alireza Tavakholi Ghainani, Abdullah Asuhaimi Mohd Zin, and Nur Ain Maiza Ismail. Fuzzy Timing Petri Net for Fault Diagnosis in Power System. *Mathematical Problems in Engineering*, 2012:1–12, 9 2012.

[135] Sherif Abdelwahed and Gabor Karsai. Notions of diagnosability for timed failure propagation graphs. In *AUTOTESTCON (Proceedings)*, pages 643–648, 2007.

[136] S. Abdelwahed, G. Karsai, N. Mahadevan, and S.C. Ofsthun. Practical Implementation of

Diagnosis Systems Using Timed Failure Propagation Graph Models. *IEEE Transactions on Instrumentation and Measurement*, 58(2):240–247, 2 2009.

[137] Pavel Krčál, Leonid Mokrushin, P. S. Thiagarajan, and Wang Yi. Timed vs. Time-Triggered Automata. pages 340–354. Springer, Berlin, Heidelberg, 2004.

[138] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 4 1994.

[139] Gerd Behrmann, Alexandre David, and Kim G. Larsen. A Tutorial on UPPAAL. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3185:200–236, 2004.

[140] TCD. https://github.com/chhokrad/tcd.git.

[141] N. Mahadevan, A. Dubey, A. Chhokra, H. Guo, and G. Karsai. Using temporal causal models to isolate failures in power system protection devices. *IEEE Instrumentation Measurement Magazine*, 18(4):28–39, August 2015.

[142] Ajay Chhokra, Nagabhushan Mahadevan, Abhishek Dubey, and Gabor Karsai. Qualitative fault modeling in safety critical cyber physical systems. In *Proceedings of the 12th System Analysis and Modelling Conference*, SAM '20, page 128137, New York, NY, USA, 2020. Association for Computing Machinery.

[143] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Snchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[144] PyPower. https://github.com/rwl/pypower.

[145] Leon Thurner, Alexander Scheidler, Florian Schfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. Pandapoweran open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, 2018.

[146] Working Group. Common format for exchange of solved load flow data. *IEEE Transactions on Power Apparatus and Systems*, PAS-92(6):1916–1925, 1973.

[147] Ajay Chhokra, Carlos Barreto, Abhishek Dubey, Gabor Karsai, and Xenofon Koutsoukos. Power-attack: A comprehensive tool-chain for modeling and simulating attacks in power systems.

[148] Ajay Chhokra, Abhishek Dubey, Nagahbhushan Mahadevan, and Gabor Karsai. A component-based approach for modeling failure propagations in power systems. In *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6, 2015.

[149] Ajay Chhokra, Nagabhushan Mahadevan, Abhishek Dubey, Daniel Balasubramanian, and Gabor Karsai. Towards diagnosing cascading outages in cyber physical energy systems using temporal causal models. volume 9, 2017.

[150] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D.C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, 3 1996.

[151] MATLAB. *9.7.0.1190202 (R2019b)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[152] Texas A&M University. Electric grid test case repository.

[153] Sel-311c: Transmission protection system, 2016.

[154] Ajay D Chhokra, Nagabhushan Mahadevan, Abhishek Dubey, Saqib Hasan, Daniel Balasubramanian, and Gabor Karsai. Hierarchical reasoning about faults in cyber-physical energy systems using temporal causal diagrams. *International Journal of Prognostics and Health Management*, 9(1), 2018.

[155] Duncan J Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[156] Laurent Hébert-Dufresne, Pierre-André Noël, Vincent Marceau, Antoine Allard, and Louis J

Dubé. Propagation dynamics on networks featuring complex topologies. *Physical Review E*, 82(3):036115, 2010.

[157] Peter S Bearman, James Moody, and Katherine Stovel. Chains of affection: The structure of adolescent romantic and sexual networks1. *American journal of sociology*, 110(1):44–91, 2004.

[158] Paolo Crucitti, Vito Latora, and Massimo Marchiori. A topological analysis of the italian electric power grid. *Physica A: Statistical mechanics and its applications*, 338(1):92–97, 2004.

[159] Margaret J Eppstein and Paul DH Hines. A random chemistry algorithm for identifying collections of multiple contingencies that initiate cascading failure. *IEEE Transactions on Power Systems*, 27(3):1698–1705, 2012.

[160] Benjamin A Carreras, Vickie E Lynch, Ian Dobson, and David E Newman. Critical points and transitions in an electric power transmission model for cascading failure blackouts. *Chaos: An interdisciplinary journal of nonlinear science*, 12(4):985–994, 2002.

[161] Dusko P Nedic, Ian Dobson, Daniel S Kirschen, Benjamin A Carreras, and Vickie E Lynch. Criticality in a cascading failure blackout model. *International Journal of Electrical Power & Energy Systems*, 28(9):627–633, 2006.

[162] MP Bhavaraju and NE Nour. Trelss: A computer program for transmission reliability evaluation of large-scale systems. Technical report, Electric Power Research Inst., Palo Alto, CA (United States); Public Service Electric and Gas Co., Newark, NJ (United States), 1992.

[163] Jiajia Song, Eduardo Cotilla-Sanchez, Goodarz Ghanavati, and Paul DH Hines. Dynamic modeling of cascading failure in power systems. *IEEE Transactions on Power Systems*, 31(3):2085–2095, 2016.

[164] Paul Hines and Sarosh Talukdar. Controlling cascading failures with cooperative autonomous agents. *International journal of critical infrastructures*, 3(1-2):192–220, 2006.

[165] A Berizzi. The italian 2003 blackout. In *Power Engineering Society General Meeting, 2004. IEEE*, pages 1673–1679. IEEE, 2004.

[166] Juhwan Jung, Chen-Ching Liu, Steven L Tanimoto, and Vijay Vittal. Adaptation in load shedding under vulnerable operating conditions. *IEEE Transactions on Power Systems*, 17(4):1199–1205, 2002.

[167] Mads R Almassalkhi and Ian A Hiskens. Model-predictive cascade mitigation in electric power systems with storage and renewablespart i: theory and implementation. *IEEE Transactions on Power Systems*, 30(1):67–77, 2015.

[168] S. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, 27(06):509–516, jun 1978.

[169] Justin Gray, Kenneth T Moore, and Bret A Naylor. Openmdao: An open source framework for multidisciplinary analysis and optimization. In *AIAA/ISSMO Multidisciplinary Analysis Optimization Conference Proceedings*, volume 5, 2010.

[170] Roger C Dugan. Reference guide: The open distribution system simulator (opendss). *Electric Power Research Institute, Inc*, 2012.

[171] Iraj Dabbagchi. Ieee 14-bus test system.

[172] F. L. Alvarado. Computational complexity in power systems. *IEEE Transactions on Power Apparatus and Systems*, 95(4):1028–1037, July 1976.

[173] T Athay, R Podmore, and S Virmani. A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, (2):573–584, 1979.

[174] Ajay Chhokra, Amogh Kulkarni, Saqib Hasan, Abhishek Dubey, Nagabhushan Mahadevan, and Gabor Karsai. A systematic approach of identifying optimal load control actions for arresting cascading failures in power systems. In *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, CPSR-SG'17, page 4146, New York, NY, USA, 2017. Association for Computing Machinery.

[175] Ajay Chhokra, Saqib Hasan, Abhishek Dubey, and Gabor Karsai. A binary decision diagram based cascade prognostics scheme for power systems. In *2020 American Control Conference (ACC)*, pages 3011–3016, 2020.

[176] Florida Reliability Coordinating Council, Inc. https://www.frcc.com/AboutUs/SitePages/Home.aspx.

[177] Midwest Reliability Organization. https://www.mro.net/about/Pages/default.aspx.

[178] Northeast Power Coordinating Council, Inc. https://www.npcc.org/About/default.aspx.

[179] Reliability First. https://www.rfirst.org/about/Pages/AboutUs.aspx.

[180] South East Reliability Corporation. https://www.serc1.org/about-serc.

[181] Southwest Power Pool. https://www.spp.org/about-us/.

[182] Texas Reliability Entity. https://www.texasre.org/Pages/About-Us.aspx.

[183] Western Electricity Coordinating Council. https://www.wecc.biz/Pages/AboutWECC.aspx.

[184] IEEE Power Engineering Society. Substations Committee., Institute of Electrical and Electronics Engineers., and IEEE-SA Standards Board. *IEEE standard for SCADA and automation systems*. Institute of Electrical and Electronics Engineers, 2008.

[185] IEC 60870-6-503:2014, 2014.

Acronyms

AC Alternating Current. 6, 7, 153

CART Classification and Regression Tree. 34

CB Circuit Breaker. 7, 8, 11

CCT Critical Clearing Time. 24

CPES Cyber-Physical Energy System. v, 1–3, 42–44, 102

CPS Cyber-Physical System. ix, 1, 2, 4, 43

CPU Central Processing Unit. 11

CT Current Transformer. 9, 19, 20

DC Direct Current. 154

DFR Digital Fault Recorders. 28

DNP3 Distributed Network Protocol. 11

ED Economic Dispatch. 180

ELM Extreme Learning Machine. 35

EMF Electromotive Force. 20

EMS Energy Management Systems. 1, 169, 171, 172

EMTP Electro-Magnetic Transients Program. 37

FRCC Florida Reliability Coordinating Council. 153

HMI Human Machine Interface. 170

I/O Input/Output. 11

IED Integrated Electronic Device. 28, 168–172

LSE Load Serving Entities. 156, 157

MIDAS Misoperation Information and Data Analysis System. 15

MLP Multi-Layer Perceptron. 35, 36

MRO Midwest Reliability Organization. 153

NERC North American Electric Reliability Corporation. 3, 16, 23, 25, 26, 42, 153, 158

NPCC Northeast Power Coordinating Council. 153, 154

OOS Out-Of-Step. 21

OPF Optimal Power Flow. 178, 180

ORPF Optimal Reactive Power Flow. 180

OST Out-of-Step Tripping. 21

PBS Power Swing Block. 20, 21

PLC Programmable Logic Controller. 169, 171

PN Petr Nets. 40

PSO Particle Swarm Optimization. 38

PSS Power System Stabilizer. 182

PT Potential Transformer. 9

QSE Qualified Scheduling Entities. 156, 157

RBF Radial basis Function. 35–37

RE Resource Entities. 156, 157

RF Reliability First. 153

RNN Recurrent Neural Network. 35

RTU Remote Terminal Unit. 42, 169–171

SCADA Supervisory Control and Data Acquisition. 1, 28, 41, 169–171, 173

SCED Security Constrained Economic Dispatch. 180

SCOPF Security Constrained Optimal Power Flow. 180

SDG Signed Directed Graph. 39

SERC South East Reliability Corporation. 153

Sf$_6$ Sulfur hexafluoride. 12

SPP Southwest Power Pool. 153

SVM Support Vector Machines. 36, 37

TCD Temporal Causal Diagram. 45

TCG Temporal Causal Graph. 39

TDSP Transmission and Distribution Service Providers. 156

TFPG Temporal Fault Propagation Graph. 44, 45

TRE Texas Reliability Entity. 153, 154

TTA Time Triggered Automata. 45

UCE Unit Control Error. 182

VT Voltage Transformer. 9

WECC Western Electricity Coordinating Council. 153, 154

WLS Weighted Least Square. 174

WSCC Western System Coordinating Council. 98