# AN ABSTRACT OF A THESIS

# CONTINUOUS SURVEILLANCE DESIGN FOR CRITICAL SMART GRID INFRASTRUCTURE USING UNMANNED AERIAL VEHICLE (UAV)

Rahat Masum

Master of Science in Computer Science

A smart grid is a widely distributed engineering system with overhead transmission lines often running through deep forests, long rivers, coastal and hilly areas, and busy cities. Physical damage to those power lines, from natural calamities or technical failures, will disrupt the functional integrity of the grid. To ensure system recovery and the continuation of secure operational flow when those phenomena happen, the grid operator must immediately take steps to nullify the impacts and repair the problems, even if those occurs in hardly reachable remote areas. Emerging unmanned aerial vehicles (UAVs) show great potential to replace traditional human patrols for regularly monitoring critical situations involving the safety of the grid. The critical lines can be monitored by a fleet of UAVs to ensure a resilient surveillance system. The proposed approach first considers the $n$-1 contingency analysis to find the criticality of a transmission line from its performance index, which is calculated using linear sensitivity factors. We divide a line into smaller segments forming multiple inspection points on the line. Then, we propose a formal framework that verifies whether a given set of UAVs can perform continuous surveillance of the grid satisfying various requirements, particularly the monitoring and resiliency specifications. The verification process ultimately provides a trajectory plan for the UAVs, including the refueling schedules. The resiliency requirement of inspecting a point is expressed in terms of a $k$-property specifying that if $k$ UAVs fail or compromised still there is a UAV to collect the data at the point within a threshold time, while the points that are under resilient surveillance cover a required percentage of the grid's overall criticality. We evaluate the proposed framework on synthetic data based on various IEEE test bus systems. With the solution model, we have implemented a graphical simulation UX using Unity3D to present the routing of UAVs as a surveillance scenario.

# CONTINUOUS SURVEILLANCE DESIGN FOR CRITICAL SMART GRID INFRASTRUCTURE USING UNMANNED AERIAL VEHICLE (UAV)

---

A Thesis

Presented to

the Faculty of the College of Graduate Studies

Tennessee Technological University

by

Rahat Masum

---

In Partial Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE

Computer Science

---

May 2019

**CERTIFICATE OF APPROVAL OF THESIS**

**CONTINUOUS SURVEILLANCE DESIGN FOR CRITICAL SMART GRID INFRASTRUCTURE USING UNMANNED AERIAL VEHICLE (UAV)**

by

Rahat Masum

Graduate Advisory Committee:

| | |
|---|---|
| Michael Rogers, Chairperson | Date |
| Mohammad Ashiqur Rahman | Date |
| Ambareen Siraj | Date |

Approved for the Faculty:

Mark Stephens
Dean
College of Graduate Studies

Date

## DEDICATION

This thesis is dedicated to my parents,

who have always been to my support during the dark nights, the sleepless mornings and the frustrated hours of insane pressurized situations. Without their kind words and unconditional love, I would not be able to cross the journey.

## ACKNOWLEDGMENTS

I am grateful to my supervisor Dr. Mohammad Ashiqur Rahman for guiding me with proper directions to this work. It was my pleasure to work with such a hard-working person who had got the best out of me, at the same time allowed me to make this paper to be my own work. I am thankful to Dr. Ambareen Siraj for all the support she has been providing me from the very beginning of my Masters program. My heartiest respect to Dr. Mike Rogers for giving his kind consent and time to be the chairperson of my committee. I want to mention my student colleague Mr. Jakaria, Mr. Mehedi, and Mr. Amarjit for always cheering me with their friendly behavior and helping attitude whenever I became stuck at any problem. I am grateful to my friend Mr. Hasan Shahriar to help me with Matlab analysis. I would like to appreciate the overall academic and administrative support I have got from Dr. Ghafoor and our honorable Chair of the department Dr. Gannod. I express my gratitude to the Administrative Associate of our department Ms. Megan Cooper for her continuous cooperation from the first day of my journey at Tennessee Tech.

## TABLE OF CONTENTS

Page

Chapter Page

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF PROGRAMS

Program Page

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Smart grid is responsible for transmission, distribution and control management for electricity supply over homes, factories, hospitals etc. Overhead power lines provide the primary engineering infrastructure, connecting the substations and the generation facilities, to transmit electrical energy along long distances. These transmission lines are distributed from busy cities to remote country areas, often running through coastal ranges, deep forests, long rivers, and mountains. Natural calamities or technical errors can incur physical damages to the overhead power lines. A damage to this infrastructure, *i.e.*, one or more broken lines will hamper the necessary energy transmission. The failure in transmission will lead the system into an unstable or outage state. The situation becomes worse when it is delayed to fix the damage. The power line infrastructure health is important to be monitored. Spatial properties like temperature, elongation, and wind induced conductor motion are important to be taken care of for the optimal infrastructure health [1]. An unanticipated natural catastrophe (*e.g.*, wildfire) or extreme weather can weaken the condition of one or more power lines, which can cause ultimate breakage of these lines. A frequent monitoring of the health can minimize this possibility by taking mitigating actions.

The power line surveillance is traditionally event-based, especially when the control

center detects an outage. Traditionally, in such events, technicians are sent on vehicles for damage assessment tasks. However, this inspection mechanism is not only time-inefficient causing delays in recovery but also unsafe, especially during disastrous situations or hardly reachable remote areas [2]. An alternative approach is sending trained inspectors flying with helicopters to assess the lines for damages using binoculars or cameras. Although this approach is being used widely, it is still safety-critical in the event of extreme weather and the overall process of necessary actions is time consuming for remote areas. Apart from these drawbacks, the event-based surveillance delays the immediate decision making process and response time during hazardous situations. The power line health maintenance also needs frequent updates on the spatial parameters, along with the electrical properties [3]. Hence, **continuous** monitoring of the critical transmission lines is advantageous, often a necessity. However, human patrol based, even using helicopters, are infeasible for continuous monitoring due to its high operating cost, along with safety factors. In this condition, unmanned aerial vehicles (UAVs) can provide a feasible means [4], [5], [6].

The emergence of UAVs, particularly along with the rapid advancement of the corresponding technology and the increasing cost-efficient availability of UAVs, makes the UAV-based surveillance the perfect solution for continuous inspection, along with event-based cases, of the overhead transmission lines. This approach is not only free from the safety-critical issues like those with sending human patrols but also capable of providing timely, inexpensive, and wide-range of supports.

Transmission lines in a grid possess critical overload situation due to a transmission

line trip and generation outage. The overloaded lines can cause subsequent trips in a cascaded fashion if necessary recovery steps are not taken on time. Hence, it is important to analyze the impact of a line trip (or multiple trips) on the system's stability. The contingency analysis, a core component of the energy management system (EMS) in a smart grid, performs this analysis and selects the operating points (*e.g.*, generation dispatches at different generators) that keep the system in a stable situation even in contingencies – transmission line or generation source failures. Typically, $n-1$ (or $n-1-1$) contingency analysis is done where one node failure (or subsequent failure of two nodes) is considered among $n$, the total number of transmission lines (or generation sources) [7, 8]. Because of the connectivity between the buses and various loads and generations at the buses different transmission lines often impact the system differently. Hence, some transmission lines can be highly critical with respect to their individual impacts while some others may not be critical at all. The continuous monitoring of the transmission lines should consider their respective **criticality** weights into account.

The continuous surveillance includes various requirements. While the goal of the surveillance is to cover the critical points, the primary need of the continuous inspection is to maintain the data freshness - the subsequent data (*e.g.*, images) collection at a point on the transmission lines should be within a threshold time frame. The surveillance coverage requirement may include all the critical points or some of them that together cover at least a certain portion of the system's overall criticality.

While a UAV can fail or malfunction due to technical errors, it is vulnerable to cyber

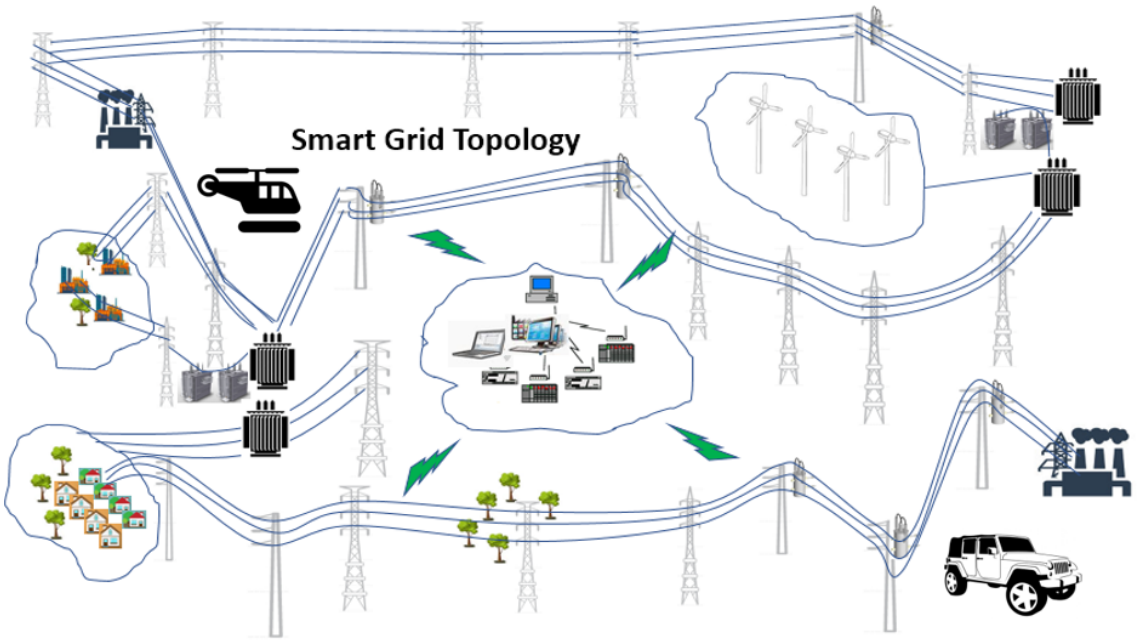


Figure 1.1: Overhead power transmission lines surveillance in a smart grid.

attacks, along with physical attacks, especially in an adversarial environment. Therefore, the surveillance **resiliency** is important. If one or more UAVs fail or are compromised, the properly functioning and uncompromised UAVs may not be able to collect and report data satisfying a minimal data freshness.

Appropriate deployment of UAVs depends on the feasibility of a trajectory plan for the UAVs that satisfies the surveillance requirement, $k-$resiliency property, and cost-effective fuel usage and refueling making the scenario a NP-hard problem. This research solves this problem by providing the trajectory plan, along with the refueling schedule, for each UAV, assuming that each UAV is connected with a base control center and deliver captured monitoring data to the base while flying over critical power lines.

## 1.2 Contribution

In summary, the major contributions of our work are as follows:

- We propose a formal model to synthesize a plan for continuous surveillance of overhead transmission lines using UAVs, satisfying the surveillance requirements. The plan includes trajectories and refueling schedules for the UAVs, given the bus system and a set of UAVs.

- We define the criticality of a transmission leveraging the contingency analysis, a core EMS control routine. To analyze the critical conditions of any lines we have followed $n-1$ contingency scenario and find out the impacts of the line outage using Shift Factors and Line Outage Distribution Factors.

- We define the resiliency property with respect to a parameter $k$, specifying a data point is under resilient surveillance if even if $k$ UAVs are unavailable or compromised. In our design $k$ is considered as multiple visits (*i.e.*, $k = 2$ times) of UAVs from the fleet.

- We implement the proposed formal model using SMT [9] and demonstrate the solution on an example case study, adapted from an IEEE test bus system [10].

- We have created one graphical simulation using Unity3d framework to present the outcome of the solution as UAV trajectopry routes and refueling plans for the assigned

fleet over IEEE-14 bus test system. The simulation is implemented based on real-time solution model and cartesian coordinate design as a view point perspective.

## 1.3 Organization

The rest of this work is organized as follows: Chapter 2 contains an in-brief discussion of the motivation for this work and the existing literature reviews; divided in different point of views. In Chapter 3, we have out solution model framework from a high level view. From the proposed solution model, Chapter 4 presents the critical analysis of transmission lines based on the methodologies used in the work. Chapter 5 possesses the formal model design for the proposed continuous surveillance model, along with the $k-$resiliency constraint and analysis. Our implementation approach and sample case study has been discussed in details in Chapter 6. An extensive evaluation has been done in Chapter 7 according to scalability and characteristics analysis. The graphics simulation for the implemented solution model is described in Chapter 8. Finally, Chapter 9 concludes our work with a future direction to the extension for scholarly publications in future.

# CHAPTER 2

# BACKGROUND AND LITERATURE REVIEW

## 2.1 Background

We briefly discuss about the motivation and background study for the power transmission line surveillance. Then, we present a related literature review for the line criticality analysis and the existing practice of UAV involvement to this concern.

### 2.1.1 Power Transmission Line Surveillance

Cyber-physical systems (CPSs) like smart grids are identified as critical infrastructures (CIs) due to the essential services they provide to the nation. A smart grid is physically a combination of generation, transmission, and distribution systems that distribute the electric power from the generation sources, through the substations (buses) to homes, industries, hospitals, offices, schools, colleges, shops, gas stations, and presently, plug-in-electric vehicles. The power transmission, to, from, and between the substations is performed via an engineering topology of overhead power transmission lines. A power transmission line is supported by towers, usually made of wood, steel, aluminum, or concrete. The overhead power lines transmit electrical energy along large distances. According to the report - "Between June 2011 and February 2013, Recovery Act funds have been used to deploy 343 advanced grid sensors, upgrade 3,000 distribution circuits with digital technology, in-

stall 6.2 million smart meters and invest in 16 energy storage projects (EOP 2013)" [11]. One or more transmission line failures (*i.e.*, tripping/breakage) can cause power outages, resulting in load shedding and, in extreme situations, blackouts for days. We are motivated at this point to provide a continuous surveillance model that will help understand the circumstances of the power lines regularly.

The transmission line trips can occur for several reasons, including equipment aging and natural disasters. Due to aging, 70% of the transmission lines, including other machines, have an average age of 25 to 30 years [12]. Hence, these components need to be inspected regularly in order to maintain uninterrupted power transmission to power consumers. Severe weather conditions like hurricanes, tornadoes, and wildfires can cause severe destruction of the smart grid's physical infrastructure. The damage may include leaning poles, broken wires, tree encroachment over lines, broken switches, blown transformers, etc. These damages affect the electric supply, and the longer the recovery stage, the larger the financial loss. In this regard, the recent Hurricane Maria incident in Puerto Rico provides an opportunity to use the surveillance model. This calamity destroy the power grid and leave many roads impassable for the recovery workers [13], [14].

Although natural calamities cannot be controlled in spite of forecasts, damage assessment needs to be addressed as far out and as quickly as possible. In normal weather conditions, incidents like animal attacks, strong winds, and construction work can also trip one or more power lines. A regular, continuous monitoring process will help quickly receive information about the incidents lines through a time efficient, safe surveillance task. In ad-

dition to the above scenarios, adversaries can attack the control center by making injurious operating decisions, which can overload the transmission lines, ultimately leading to the destruction of equipment. A frequent or continuous surveillance may detect the potential failure and save the line from being tripped, and thereby the system from an outage.

### 2.1.2 Contingency Analysis

EMS is the core component of the bulk energy management in a smart grid and consists of several interdependent computational modules [7], [8]. Some of them can be seen as state estimation (SE), optimal power flow (OPF), contingency analysis (CA), and automatic generation control (AGC). These modules are interdependent on each other, *e.g.*, the output of SE and OPF is used to find out the power flows over the transmission lines to provide a secure and optimal flows within the line capacity.

EMS executes these modules based on the measurement data received from the field devices and accordingly control the grid (*e.g.*, generation set-points) for operational security and economic efficiency. Contingency analysis (CA) is one of the core EMS modules. The goal of CA is to operate the power system securely by analyzing the system subject to a contingency (*e.g.*, transmission line outage) and determine the set-points that will allow system operation without violation of constraints. Typically, $n-1$ contingency analysis is done where $n$ is the total number of nodes (either transmission lines or generation sources) and one node failure is considered [7, 8]. The analysis considers the failure of each transmission line and checks its impact on the system, *e.g.*, if the rest of the transmission lines

Table 2.1: Related Works for Power Line Monitoring and UAV Surveillance

| Area of Discussion | References |
|---|---|
| Contingency Analysis | [7] [8] |
| Surveillance Based on UAVs | [4] [5] [15] [16] |
| Overhead Line Surveillance | [2] [6] [17] [18] [19] [20] [21] |
| Optimal Surveillance | [22] [23] [24] |
| Security of UAV in Surveillance | [25] [26] |

become overloaded due to the considered trip situation. If there are overloaded cases, the generation setting points (which are usually optimized by executing another control module, namely optimal power flow (OPF)) are typically changed within the constraints, to mitigate the overloaded situation. Optimizing the set-points while mitigating or minimizing the impact of contingencies is done combining CA with OPF, namely security constrained OPF (SCOPF). Such analysis is often performed considering the $n-2$ scenario (*i.e.*, two node failures).

## 2.2 Literature Review

We discuss the existing literature related to this research in different categories. Table 2.1 presents a summarized list of the existing scholarly articles as per the categorized discussion.

### 2.2.1 UAV-Based Surveillance Technique

Srinivasan *et al.*, presented the idea of video surveillance using cameras and sensors leveraging unmanned aircraft systems in a project with the Florida Department of Transportation [4]. The video images contain the traffic information on the roads and are

transmitted by UAVs using the microwave IP network. The captured video contains information of traffic on the roads and received by wireless means of communication. At the same time, the UAVs capture weather information data with diferent sensors. Through a 2.4 GHz wireless link channel, these informations are passed to the ground station via transmission lines. Issues regarding the physical layer, communications and network layer have been mentioned as data security, integrity, reliability, signaling, and mobility while receiving from the microwave towers with an installed antena support [15]. The swarm of quad-rotor UAVs, equipped with GPS location set and surveillance capabilities, can take direction from the ground control station and determine the necessary altitude calculations for surveillance using the airborne sensors. Jaimes *et al.*, provided some aspects of the real-time image recognition task based on the videos sent from the preassigned GPS coordinate-programmed UAVs [16]. The work flow was designed in a way that UAV can go to the corresponding surveillance area and surveillances for the specific time and communicate to the ground station. The communication is done by the deployed sensors once the UAVs reached to the specific coordinate positions. Through the communication channel, collected data will be analyzed by image recognition methods to make a decision on further survveillance. After each flight completeion, UAVs will provide the live videos recorded during the flight to the ground station. A master-follower approach of UAVs was acquired through the flight paths. This can be achieved by three approaches as: (i) infinite time monitoring by every UAVs for corresponding area, (ii) independent flight mode of the UAVs, or (iii) all UAVs at the same time to do the surveillance in a simultaneous manner. UAVs were

used in a project by Moreno *et al.*, to monitor marine environments in the Mexican seashore using cheap sensors and less power-consuming buoys [5]. UAVs are treated as a form of mobile data collector having long-range communication capabilities. Since sea shoreline is a vast area to be convered, the project showed extraction for rescue purpose in the shoreline. In this project, periodic data messages from the storage of buoys were transmitted through UAVs. These data are not only for human rescue mission but also about the environmental parameters (*e.g.*, water temparature, speed of wind, wind pressure, humidity), which were used for further analysis. As the UAVs does not contain unlimited source of power/battery, the authors has discussed the return module of the UAVs as well. nce any task is complete for a UAV or, it is low in battery power, it will return to the base for next action coordinate. This work has provided a good direction to monitor large surveillance space covering any hardly reachable areas by placing the buoys into small vessel so that UAVs can fly and communicate via LoRa communication protocol.

### 2.2.2 Overhead Line Surveillance

Power line inspection requires a variety of components and situations to be surveilled. Any disaster or intentional activity can damage the transmission system that include leaning poles, broken wires, tree encroachment over lines in the forest, broken switches etc. Traditionally, electric utilities send technicians by vehicles to the potential damaged areas for inspecting the towers. This process is very time consuming and resource engaging. Ma *et al.*, discussed an alternative approach, in which trained inspectors are sent by helicopters

to inspect lines using binoculars or cameras and record data to a log book for further analysis [2]. However, this approach can be unsafe for the human patrols in extreme weather conditions and can be time consuming in remote, unpassable areas. Several works later discussed the use of UAV-based monitoring instead of traditional human patrol methods relying on human monitoring. A project named Hydro-Quebec LineScout Technology, being conducted by the Hydro-Quebec Transmission Network, presented remotely-controlled mobile robots/UAVs that are capable of performing basic power line inspection and maintenance tasks [17], [18]. Li *et al.*, presented a knowledge-based power line detection method from the captured images so that UAVs can be utilized for surveillance and inspection systems [19]. Later, the authors improved the power line detection efficiency by developing a pulse-coupled neural filter that eradicates noise data from any captured image by UAV so that straight lines can be detected [27]. A study about the low budget UAV-based power line monitoring solutions that are capable of detecting transmission lines is presented in [6]. Pagnano *et al.*, proposed an automated surveillance mechanism for the transmission lines, leveraging UAVs and robots that rolled on the wires [20]. This real-time inspection methodology uses the image and signal data processing, allowing the detection of faults or abnormalities on the lines. The solutionmainlyfocused on vertical take-off and landing (VTOL). An unmanned aerial system for high voltage power line inspection is presented in [21]. For real-time error reporting, quadcopters were equipped with color cameras controlled from the ground control station. Inspection is done by detecting hot spots captured by the aerial video components through qualitative inspections. The payloads and flight

control system being on-board to the UAS helped finding environmental parameters for power line joint detections.

### 2.2.3 Optimal Surveillance

In a complex urban environment, Semsch *et al.*, proposed a two-stage mechanism: (i) constructing a coverage of point set and, (ii) exploring trajectory based on UAV motion through the points, which can control flights for autonomous multi-UAVs to provide to allow maximum surveillance [22]. The UAVs are equipped with on-board cameras to gather information through the sensors. The mechanism looks for a set of paths so that, in case of obstacles, a minimum area will be uncovered, giving the adversaries fewer opportunities for exploitation. The authors presented the trajectory planning by the minimization of time with the maximization of coverage area. The time optimization is done by taking average time between two consecutive instances when an arbitrary surface point area can be seen. During this tim,e data record, the work has considered occulsions in presence of high-rise buildings. Lim *et al.*, targeted the challenges associated with using UAVs to scan power lines from a distance and send the damage data to the control center [23]. The authors presented a solution to minimize overall inspection time and cost by prepositioning the UAVs optimally. The solution shows the operation of UAVs with thermal FLIR camera, CCTV camera, and GPS. In this approach, the inspection problem is divided into two phases: the first phase corresponds to deploying the UAVs in the target domain, and the second one finds the optimal route to the nodes over the topology. However, the approach

is event-based, allowing for monitoring disaster situations with the prepositioned UAVs to assess the damage due to the extreme weather. Deng *et al.*, proposed a multi-platform cooperative UAV system as well as a multi-model communication system for power line inspections in China [24]. The authors considered three design challenges, such as (i) power supply of battery as per the patrol distance and time duration from the ground control station, (ii) communication range and datalink capacity for real-time data delivery, and (iii) the delay between image capturing and the corresponding analysis at the control center.

### 2.2.4 Security of UAV in Surveillance

The UAV monitoring system can succumb to cyber attacks that allows sensitive data to be collected by the adversaries. Birnbaum *et al.*, presented a real-time behavioral monitoring procedure that can convert a flight plan to behavioral profiling [25]. The analyzer tries to validate the actual flight with the standard profile, allowing the detection of any unplanned deviation due to attacks or failures. These events include hardware or communication channel failures, malicious hardware, compromised sensors, false data injections, and denial service attacks. These attacks may invalidate the work of surveillance by the UAVs making an economic loss and in some cases manipulation of captured data if accessible and not securely encrypted. Abbaspour *et al.*, demonstrated the safety-critical issues and corresponding detection mechanisms for the UAV-based surveillance with respect to several faults and sensor-spoofing attacks [26]. In particular, the sensor-spoofing attack has been discussed and a detection mechanism is proposed using adaptive neural network along

with an embedded Kalman Filter. The proposed solution detects the injection of false data in the on-board sensors and mentioned methodologies tries to detect the FDI (False Data Injection) attack with faster timely manners.

While the existing literature presented the techniques for UAV-based surveillance of power transmission lines, the trajectory planning for continuous surveillance is important, especially when different lines can have different criticality levels, the surveillance area is typically large, and the resource is often limited. Since some UAVs can fail or be compromised, resiliency of the surveillance against such events is crucial. A formal modeling of efficient routing problem has been discussed by Rahman *et al.*, using *Plug-in Hybrid Electric Vehicles* [28]. The vehicle navigation problem has been solved to satisfy the fuel requirements and routing plans. However, automated synthesis of a resilient surveillance plan, *i.e.*, trajectories of a fleet of UAVs, including their refueling, is a combinatorially hard problem. This research provides a solution to this complex problem by formally modeling the continuous and resilient surveillance properties as a constraint satisfaction problem. Due to any failure of UAVs from the fleet we will continue our surveillance to the point that will make our critical points covered obtaining a threshold score for the surveillance.

# CHAPTER 3

# RESILIENT SOLUTION APPROACH

A fleet of UAVs will be sent over the critical lines to capture images deliverable to the control station via communication channels and protocols to analyze power line situations. The existing cellular network can be leveraged for communication. Fig. 3.1 shows the proposed solution framwork for the resilient surveillance planning model.

Contingency analysis using Shift Factor (SF), discussed in Chapter 4, allows critical lines to be selected from Performance Index (PI) calculation. Transmission lines get critical weights (descending) applying K-Means clustering over PI values. UAVs are placed over lines, and a formal model assigns a route that provides surveillance satisfying the threshold

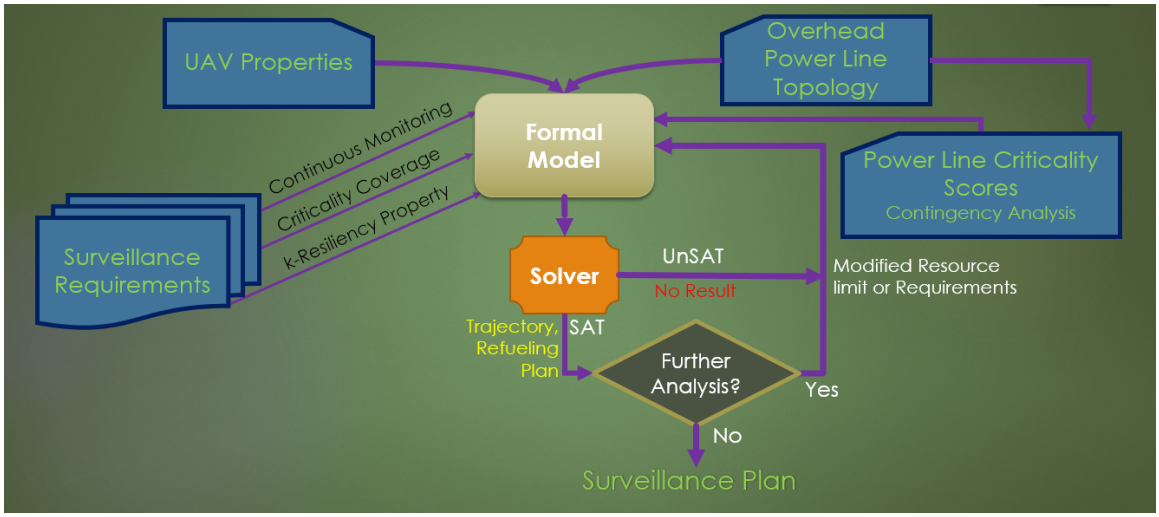


Figure 3.1: The solution approach for resilient surveillance planning and analysis.

time and fuel constraints for the routing paths of the assigned UAVs. Lines Inspected by the fleet of UAVs add critical weights to the surveillance score for analyzing costs, which is further calculated from the $k$ failed or compromised UAVs in the surveillance based on the percent of critical line coverage.

According to Fig. 3.1, the solver model in the middle of the workflow does all the constraint satisfactory tasks. This model is provided with several inputs/requirements to design the solution tree. UAV properties in the diagram contains the initial placement, fuel capacity and refueling steps information. while overhead power line topology refers to the bus network formation and line segment division, critical analysis of the lines primarily refers to the contingency analysis design for the topology. We have used $n-1$ contingency scenario to model the required monitoring case analysis. We have developed a scoring/ranking algorithm for the critical transmission line division which is implemented at this section as well. The final input to the solver model is the desired requirement for a continuous and resilient surveillance task. This requirements includes the time-step analysis for a continuos monitoring based on the estimated/targetted coverage of the critical lines in the system. With the $k$-resiliency design, the solution module satisfies the multiple visit scenarios in case of any failure of UAVs from the fleet to ensure the compact surveillance over the time period. After modeling all the requirements for the surveillance task, the solver may result in an unsatisfactory solution refering to no possible trajectory plans for the constraints set. thgis refers to the failure of covering the critical lines in the bus system based on the threshold time and resiliency requirements.

# CHAPTER 4

# CRITICAL POWER LINE ANALYSIS

The modeling of the transmission system can be considered having the physical topology. Topology primarily includes the substations (buses), the connectivity (through transmission lines/branches) among them, and the paths of the transmission lines (*i.e.*, the intermediate poles, geographical positions). The geographical topology of the transmission system is the map of the overhead power lines, *i.e.*, the physical routes of the transmission lines (through towers) connecting the substations (buses). The criticality of the transmission lines depends on the topology, electrical properties of the lines, and loads and generation dispatches at the buses. Hence, a change in the load or the generation dispatch impacts the criticality structure of the system. The criticality (computation) model is discussed bellow.

## 4.1 Line Criticality Model

We compute the criticality of a transmission line using power transfer distribution factors (PTDFs) and line outage distribution factors (LODFs) [7, 8]. PTDF gives the fraction/portion of power flowing over the lines in the bus system. The power that is sent into the topology network from a source bus $s$ to the receiving bus $r$; which flows over line $l$ connected from bus $i$ to bus $j$. The methodology we have used in this work is a similar technique known as shift factors (SFs).

### 4.1.1 Distribution Factor Calculation

Let $\mathbb{B}$ be the set of buses and $\mathbb{L}$ be the set of all lines in the system. $G_i$ and $L_i$ are the generation and the load, respectively, at bus $i$. $P_l$ is the power flow through line $l$ and $\bar{P}_l$ is the flow capacity of the line.

SFs determine the flows on branches and interfaces in the bus network. A shift factor is computed in a bus system for a particular line with a reference direction and a particular bus. SFs are the entries of matrix **SF** where the $\{l, i\}^{th}$ element ($SF_{l,i}$) is the sensitivity of the flow on line $l$ with respect to a resource (a power injection or a load) at bus $i$ [29]. It is calculated from the impedance matrix **Z** – an inverse of the admittance matrix. If $X_l$ is the reactance of line $l$ that connects buses $m$ and $n$, then $SF_{l,i}$ is calculated as follows:

$$SF_{l,i} = (Z_{m,i} - Z_{n,i})/X_l \tag{4.1}$$

The power flow on line $l$ ($P_l$) in the reference bus direction due to a power injection/generation $G_i$ (negative direction for a load/withdrawal $L_i$) at bus $i$ can be calculated as $P_{l,i} = SF_{l,i} \times G_i$, while the total power flow $P_l$ is calculated from the all power injections/generations ($G_i$s) and loads ($L_i$s).

$$P_l = \sum_{i \in \mathbb{B}} SF_{l,i} \times (G_i - Li) \tag{4.2}$$

In case of $n$-1 contingencies, outage in one line will affect the power flow in other lines. The impact of the outage can be calculated from line outage distribution factors (LODFs). LODF is a two dimentional matrix ($m \times n$) similar to SFs except for the dimensions. It is a (*line* by *line*) matrix whereas, SF is a (*line* by *bus*) matrix excluding the reference

(slack) bus. For LODFs, during $n$-1 contingency, the line which contains the outage, gets the factor as 0 (zero) since this line is no longer carrying any power flow from the buses it is connected with. With the changed amount of the line power flows due to the outage, all the other transmision lines now get the additional (increase/decrease) power flows (portion) which was initially carried out by the line in outage. These additional flows will be added (regardless of its sign/direction) to the original flows of all the other transmission lines in the bus network. The direction depends on the signs of the corresponding LODF element from the matrix calculated.

If $LODF_{l,l'}$ defines the LODF for line $l$ after an outage on line $l'$, the change in the power flow ($\Delta P_l$) on line $l$ due to the failure of line $l'$ can be found as follows:

$$\Delta P_l = LODF_{l,l'} \times P_l \tag{4.3}$$

From the calculated change in flows, modified amount of powers through the lines $P_{l,l'}$ after the contingency (outage) can be found as follows:

$$P_{l,l'} = \Delta P_l + P_l \tag{4.4}$$

Since we are considering the $n$-1 scenario, for each of the line from the total $n$ lines in the bus topology, Equation 4.4 is calculated. These calculations help to find out all the possible recovery aprroaches from the flow analysis to prevent discurption in the system due to any line outage. The grid operator can set the generation constraints accordingly to maintain the powerflow without overloading from the different values of $P_{l,l'}$.

### 4.1.2 Performance Index (PI) Calculation

The sensitivity factors give reasonably close estimates of real power flows in the lines in the event of a line (or generator) outage. Based on the results of the sensitivity analysis, all the outage cases are ranked according to a performance index (PI) calculation. If $\bar{P}_l$ is the capacity of line $l$ and $n$ is a suitable index, for a contingency, *e.g.*, the outage of line $l'$, the simplest form of PI (here $PI_{l'}$) will be as follows [8]:

$$PI_{l'} = \sum_{l \in \mathbb{L}} \left( P_l / \bar{P}_l \right)^{2n} \tag{4.5}$$

A larger PI for a line shows that it has higher criticality than that of a line with a smaller PI; in this way, PI values help rank the transmission lines per their critical sensitivities. There are improved ways of calculating the PI for better understanding of the critical contingencies [30]. The resilient surveillance model does not depend on the criticality computation mechanism.

## 4.2 Weight Assignement for Critical Lines

According to the PI values, we rank the transmission lines. We assign a criticality weight to each line according to its PI rank. To provide the weight assignment, we develop a mechanism based on a clustering method (refer to Appendix B for interested readers). In particular, the technique forms a set of clusters from the PIs following the $K$-means clustering method [31]. According to given PI values, the criticality weights of the transmission lines ($LC_l$s) are returned. The lines with the PI values falling within the same

cluster receive the same criticality score and the cluster mean is considered this score. The mechanism finds the minimum $K$ based on a threshold distance value. While the threshold value ensures a particular score be assigned to a set of lines whose PI values are close to each other (no more than a distance from the cluster centroid), the minimization of $K$ reduces the impact of small variances in PI values and helps in prioritizing the lines.

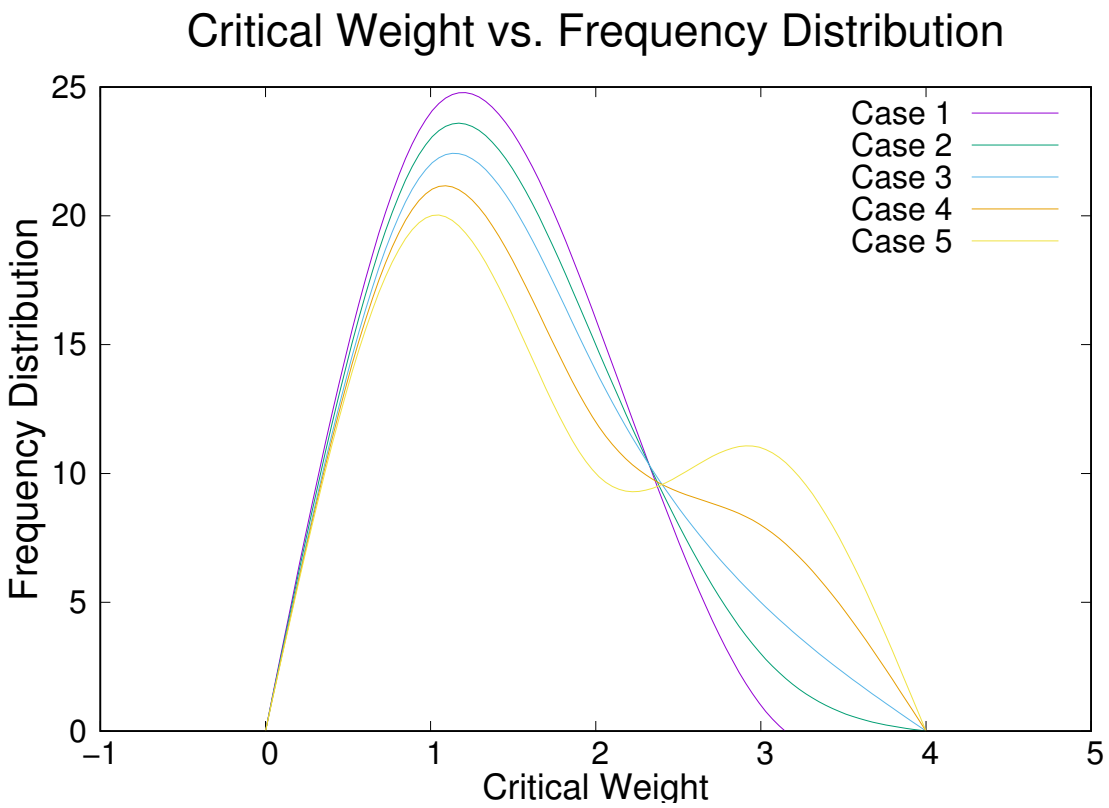


Figure 4.1: Impact of the critical weight distribution over the number of lines (frequency) for the synthetic cases.

In regard to this, Fig. 4.1 shows the overall weight distribution for the transmission lines. In each of the case there is a variation of different criticality distances (PI Score). The variation is shown as a function of number of lines to the critical distance for an IEEE-30

bus test system. Each curve in the figure holds the information about how many lines possess how much criticality from the overall weight assignments.

The cases mentioned in the curve plot are taken with different critical stages of the overall tranmission lines. Each case has variation with the ciritcal level while making the total summation of critical weights same. These curves represents the normal distribution of the critical weights over the 41 lines (for IEEE-30 bus). The cases are designed in a increasing manner from average criticality to the higher criticality distribution.

# CHAPTER 5

# CONTINUOUS AND RESILIENT SURVEILLANCE DESIGN

We have designed and solved the proposed solution framework shown in Fig. 3.1 for the continuous, resilient surveillance planning tasks. A fleet of UAVs will be sent over the critical lines to capture images deliverable to the control station via communication channels and protocols to analyze power line situations. The existing cellular network can be leveraged for communication. Fig. 5.1 shows the scenario where resilient surveillance is done using $k$ UAVs.

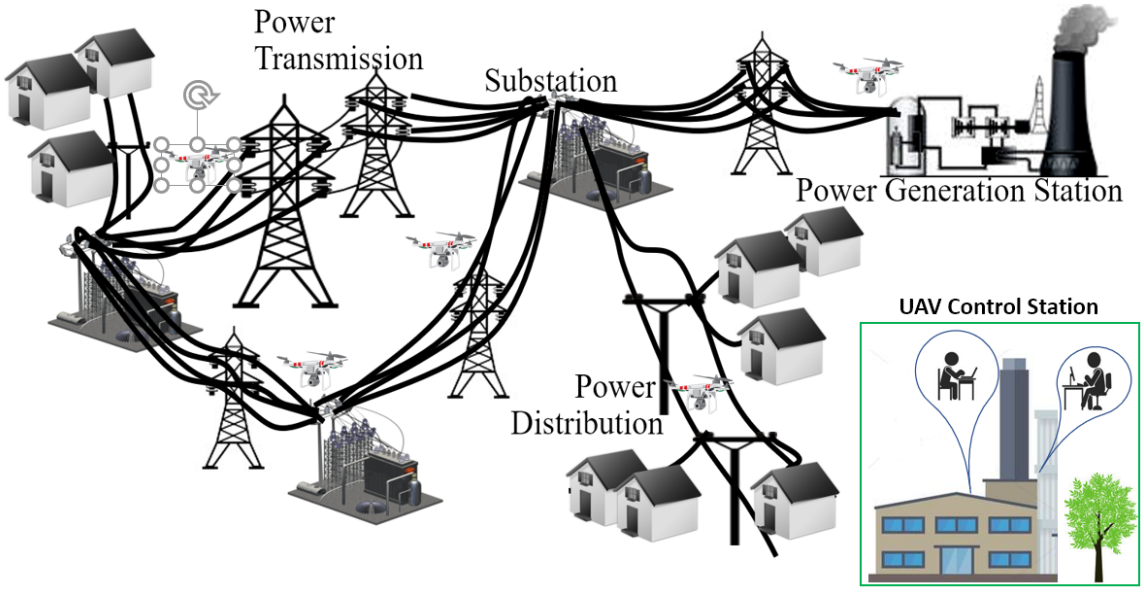


Figure 5.1: UAV surveillance operation over power transmission lines.

### 5.1 Primary System Model

We assume the UAVs will do the continuous surveillance of the transmission system by flying over (within a safe distance) the power lines [32]. A UAV's trajectory will follow the bus topology except when it needs to refuel. We define the surveillance of the transmission lines considering many surveilling points on the line topology/infrastructure. A line is divided into several segments, with each segment (link) connecting two points. We assume a fixed segment length throughout the modeling, and hence, the number of segments or points on a transmission line represent its physical length. The surveillance is modeled through visiting these points.

Each UAV is identified through an ID. A UAV possesses a set of properties like its average speed, fuel capacity, initial fuel, and starting position on the line topology. In this modeling, we also assume that all UAVs maintain the same speed on an average throughout the surveillance period irrespective of the altitude differences or necessary turns between the points. However, different UAVs may need different fuel consumption to maintain the speed.

Let $\mathbb{P}$ denote the set of points on the network and $\mathbb{U}$ be the set of UAVs to perform the surveillance. Moreover, $\mathbb{P}_l$ ($\mathbb{P}_l \subseteq \mathbb{P}$) is the set of points on line $l$ and $Seg_{p,p'}$ denotes if points $p$ and $p'$ are connected. Since the UAVs have the same speed, the same amount of time will be taken to cover each segment (*i.e.*, to fly from one point to another). We consider this required time as one time unit. The surveillance task will be modeled for a period of time, say $S$ time units/slots, where $s$ ($1 \leq s \leq S$) will identify a particular time

step.

Each UAV $u$ has a fuel capacity ($FuelCap_u$) and it starts surveillance with an initial fuel ($InitFuel_u$) from a specific point ($InitPoint_u$). Let $Fuel_{u,s}$ be the remaining fuel of UAV $u$ at step $s$, $FFuel_u$ be the fuel required to fly a segment (*i.e.*, fuel consumption at each time unit), and $HFuel_u$ be the fuel required for hovering during a time step. We use $TBase_p$ to denote the time steps to go to the base or the (closest) refueling station from point $p$ for refueling.

## 5.2 UAV Trajectory Model

Let $Visit_{u,p,s}$ denote whether UAV $u$ is visiting point $p$ at time $s$ ($s > 1$). UAV $u$ can visit point $p$ at $s$ ($> 1$) in two cases. In the first case, the UAV is already there (i.e., if it is hovering/loitering), and it has sufficient fuel for hovering during the time step. In the second case, the UAV is at point $p'$ that is connected to point $p$ ($Seg_{p',p}$ is true), and it has sufficient fuel to fly there. In both of the cases, the sufficient fuel requirement also includes the fuel needed to fly to the refueling station from the visited point ($p$). This constraint ensures that no UAV will be out of fuel and be stranded. We formalize the first case ($Hover_{u,p,s}$) as follows:

$$\begin{aligned} Hover_{u,p,s} \rightarrow \;& Visit_{u,p,s-1} \wedge (Fuel_{u,s} = Fuel_{u,s-1} - HFuel_u) \\ & \wedge (Fuel_{u,s} \geq FFuel \times TBase_p) \end{aligned} \tag{5.1}$$

Fig. 5.2 shows the time division within the surveillance period to satisfy the continuous surveillance task.


Figure 5.2: An illustration of the time steps within the surveillance period.

In the second case ($Fly_{u,p,s}$), the fuel consumption cost depends on the climbing angle of the segment ($Seg_{p',p}$). This impact of climbing angle (upward or downward) on the fuel consumption is abstracted using a ratio ($CRatio_{p',p}$) of the required cost to fly that segment over that of flying the same length of a horizontal segment. The following equation presents the corresponding formalization:

$$\begin{aligned} Fly_{u,p,s} \rightarrow & \bigvee_{p' \in \mathbb{P}, p' \neq p} Seg_{p',p} \wedge Visit_{u,p',s-1} \wedge \\ & (Fuel_{u,s} = Fuel_{u,s-1} - FFuel \times CRatio_{p',p}) \\ & (Fuel_{u,s} \geq FFuel \times TBase_p) \end{aligned} \tag{5.2}$$

Therefore, visiting a point $p$ at time $s$ by UAV $u$ ($Visit_{u,p,s}$ ) is defined as follows:

$$Visit_{u,p,s} \rightarrow Hover_{u,p,s} \vee Fly_{u,p,s} \tag{5.3}$$

We assume that if no UAV partially covers a segment partially, *i.e.*, if the UAV starts flying from a point over a segment, it reaches the end point of the segment. The same is true about hovering at a point during a time step. The initial location (at $s = 1$) of each UAV $u$ is identified at some point $p$ on the topology according to its initial (current/given) placement.

A point is connected with two or multiple points. At a particular time step, one UAV can only choose one segment. The constraint is formalized as follows:

$$\forall_{u,p,s}\, Visit_{u,p,s} \;\rightarrow \bigwedge_{p' \in \mathbb{P},\ p \neq p'} \neg\, Visit_{u,p',s} \tag{5.4}$$

$Visited_{p,s}$ denotes whether point $p$ has been visited by any UAV at step $s$. Hence:

$$Visited_{p,s} \rightarrow \bigvee_{u \in \mathbb{U}} Visit_{u,p,s} \tag{5.5}$$

## 5.3 UAV Refueling Model

The refueling of a UAV is modeled by abstracting its path from a point $p$ to the base or a refueling station and subsequently returning to a point $p'$. For simplicity of presenting the refueling model, we assume only one refueling station. The path distance from a point to the refueling center is often more than one time step ($TBase_p$). We define $ToRefuel_{u,p,s}$ to denote UAV $u$ is moving to the station from point $p$ at time $s$ for refueling, $FromRefuel_{u,p,s}$ to represent the return of UAV $u$ after the refueling to point $p$ at time $s$, and $Refuel_{u,s}$ to specify UAV $u$ is refueling at the station at time $s$. If $ToRefuel_{u,p,s}$ is true, then the following equation holds:

$$\begin{aligned} ToRefuel_{u,p,s} \rightarrow\ & Visit_{u,p,s} \wedge Refuel_{u,s+TBase_p} \wedge \\ & \left( \bigwedge_{p \in \mathbb{P}} \bigwedge_{s < s' \leq s + TBase_p} \neg\, Visit_{u,p,s'} \right) \end{aligned} \tag{5.6}$$

We also need to ensure that $Refuel_{u,s}$ is true only if there is a valid $ToRefuel_{u,p,s-TBase_p}$ for some point $p$. Similarly, if $Refuel_{u,p,s}$ is true, then the following equation must hold:

$$Refuel_{u,s} \rightarrow \bigvee_{p \in \mathbb{P}} Visit_{u,p,s+TBase_p} \wedge FromRefuel_{u,p,s+TBase_p} \wedge \left( \bigwedge_{p \in \mathbb{P}} \bigwedge_{s < s' < s+TBase_p} \neg Visit_{u,p,s'} \right) \tag{5.7}$$

It is also ensured that $FromRefuel_{u,p,s}$ is true only if there is a valid $Refuel_{u,s-TBase_p}$ and this $p$ is the only return point for this particular refueling.

We assume that a UAV refuels up to its capacity. The stored fuel after refueling, more appropriately after returning to perform surveillance, is computed as follows:

$$FromRefuel_{u,p,s} \rightarrow Fuel_{u,s} = FuelCap_u - FFuel_u \times TBase_p \tag{5.8}$$

### 5.4 Continuous Surveillance Model

The continuous surveillance for a point requires that it is always visited at least once within a (given) threshold period of time ($TC$). In other words, each pair of two consecutive visits to this point is done within $TC$. If $Surveilled_p$ denotes if point $p$ has been continuously surveilled in $S$. In this case, if point $p$ is visited at time $s$, then the next visit to this point needs to be at some time $s'$ within $TC$.

$$Surveilled_p \rightarrow \bigwedge_{1 \leq s \leq (S-TC)} Visited_{p,s} \rightarrow \bigvee_{s < s' \leq (s+TC)} Visited_{p,s'} \tag{5.9}$$

To make the continuous surveillance true from the beginning (to initiate the above equation to act for all points), $Surveilled_p$ must also ensure that starting from the beginning

within the threshold time there is at least one visit to point $p$:

$$Surveilled_p \rightarrow \bigvee_{1 \leq s \leq TC} Visited_{p,s} \tag{5.10}$$

## 5.5 Resilient Surveillance Model

A point is under $k-$resilient surveillance if it is visited by $k+1$ (different) UAVs within a (given) threshold time $TR$ throughout the surveillance period. We define that point $p$ is under resilient surveillance ($ResVisited_{p,s}$) at time $s$ (for the time period $s+TR$) if the following equation holds:

$$\begin{aligned} ResVisited_{p,s} \rightarrow\ & Visited_{p,s} \wedge \\ & (\sum_{u \in \mathbb{U}} VisitDuring_{u,p,s,TR} \geq (k+1)) \vee ((s+TR) \leq S)) \end{aligned} \tag{5.11}$$

Here, $VisitDuring_{u,p,s,TR}$ denotes that if point $p$ is visited by $u$ during the period from $s$ to $TR$. That is:

$$VisitDuring_{u,p,s,TR} \rightarrow ((s+TR) \leq S) \wedge \bigvee_{s \leq s' \leq s+TR} Visit_{u,p,s'} \tag{5.12}$$

Then, the continuous monitoring requirement is ensures for resilient surveillance by the following constraint:

$$\begin{aligned} ResSurveilled_p\ \rightarrow\ & Surveilled_p \wedge \\ & \bigwedge_{1 \leq s \leq (S-TR)} ResVisited_{p,s} \rightarrow \\ & \bigvee_{s < s' \leq (s+TR)} ResVisited_{p,s'} \end{aligned} \tag{5.13}$$

## 5.6 Criticality Coverage Requirements Model

The objective of the surveillance is to continuously monitor the transmission system (*e.g.*, power lines, generations, or other physical components) such that the surveilled points cover at least a threshold part of the overall criticality. We define the criticality coverage score as the criticality of the points under continuous surveillance over the total criticality of the system. We consider that a point on a line will have the same criticality weight ($PC_p$) as that of the line ($LC_l$) (see Section 4.2). However, a point at which two or more lines connected (*i.e.*, at a substation), the criticality weight will be the average of the corresponding lines' weights.

$$\forall_{p \in \mathbb{P}_l} PC_p = LC_l \tag{5.14}$$

If $CS$ denotes the minimum requirement of the criticality coverage score for the continuous surveillance, then the following should be satisfied:

$$\frac{\sum_{p \in \mathbb{P}} Surveilled_p \times PC_p}{\sum_{p \in \mathbb{P}} PC_p} \geq CS \tag{5.15}$$

For an arithmetic operation on an Boolean parameter, we assume Boolean "*true*" and "*false*" as integer 1 and 0, respectively. The resilient surveillance requirement is often different than that of the continuous surveillance because the former considers a contingency or attack scenario. If $RCS$ denotes the required criticality score under the resilient surveillance, then we have the following constraint:

$$\frac{\sum_{p \in \mathbb{P}} ResSurveilled_p \times PC_p}{\sum_{p \in \mathbb{P}} PC_p} \geq RCS \tag{5.16}$$

## 5.7 Surveillance Repetition Plan

Since the surveillance is continuous and the model considers a particular surveillance period, the operator may repeatedly follow the same trajectory plan or executing the model for continuous surveillance periods considering the last execution result as the input for the next run. This is because the model cannot be run efficiently for a non-trivial long period as the number of clauses grows exponentially (see later in Section 7).

In the first case, when the same trajectory plan will be executed constantly, we can achieve this requirement by ensuring a couple of constraints at the end of the surveillance period (*i.e.*, at time step $S$). Firstly, each UAV $u$ must return to its starting position ($InitPoint_u$). Secondly, the remaining fuel of the UAV at step $S$ needs to be equal to or greater than its initial stored fuel. The following equation formalizes these two constraints:

$$Visit_{u,InitPoint_u,S} \ \wedge \ (Fuel_{u,S} \geq InitFuel_u) \tag{5.17}$$

Lastly, we will need to ensure the continuous surveillance between the last visit to a point in one cycle and the first visit to the point in the next cycle. Hence, Equation (5.9) will be redefined as:

$$Surveilled_p \ \rightarrow \bigwedge_{1 \leq s \leq S} Visited_{p,s} \rightarrow \bigvee_{s < s' \leq (s+TC)} Visited_{p,(s'\%S)} \tag{5.18}$$

Similarly, Equation (5.13) will be updated.

# CHAPTER 6

# IMPLEMENTATION AND CASE STUDY

We briefly discuss the implementation of the model and present a synthetic case study illustrating the model's execution.

## 6.1 Implementation

We use the SMT logic [9] to encode the formalization presented in the previous section. We use primarily Boolean, integer, and real number terms in encoding. In the formal model, where the Boolean parameters are used in an arithmetic operation (*e.g.*, in an multiplication), we create a pseudo Integer term for each Boolean parameter to evaluate the term as 1 or 0 for Boolean "true" and "false", respectively and perform the operation. All the constraints are modeled to satisfy as per requirement of the Boolean variables/terms.

The encoded model is solved using Z3, an efficient SMT solver [33]. We use an appropriate API to perform the complete process of encoding and solving the model through developing a C# program. The solution to the model gives a result primarily as *sat* or *unsat*. In the case of *sat*, the solver provides the values of the terms, including the unknowns. From these value assignments, we receive the detailed trajectory planning that can successfully perform the surveillance satisfying the constraints. More specifically, the terms $Visit_{u,p,s}$s, $ToRefuel_{u,p,s}$s, $Refuel_{u,s}$s, $FromRefuel_{u,p,s}$s, and $Fuel_{u,s}$s provide the trajectory paths and

refueling plans, including remaining fuels.

In the case of *unsat*, we can conclude that there is no surveillance plan under the given number of UAVs and resiliency surveillance requirements. The developed program reads necessary inputs (*i.e.*, data about the bus topology, the transmission line infrastructure, the UAV fleet, and surveillance requirements) from a text file. The required outputs are printed on different text files. We also develop a 3D graphical interface, based on Unity [34], to demonstrate the surveillance plan which will be discussed in Chapter 8.

## 6.2 Example Case Study

Here we consider a synthetic overhead power line infrastructure based on the IEEE 14-bus test system. There are 14 buses and 20 lines in this test system. Fig. 6.1 shows the power line infrastructure of the IEEE-14 bus test system [10]. In our analysis each transmission line is divided into multiple segments where all the segments are assumed as equal length (distance). There are 65 equal-length (1 mile) segments, each connecting two points on the transmission lines. The points are numbered from 1 to 59. There is a fleet of 5 UAVs to perform the surveillance. The surveillance period is 91 time units/steps. As we assumed in the modeling, each segment is covered in a time step, although the fuel consumption/cost of covering a particular segment depends on the type (mileage property) of the UAV and the climbing angle for this segment.

The partial input file corresponding to this case study is presented in Table 6.1. According to the load and generation information of the buses, there are 5 generation buses

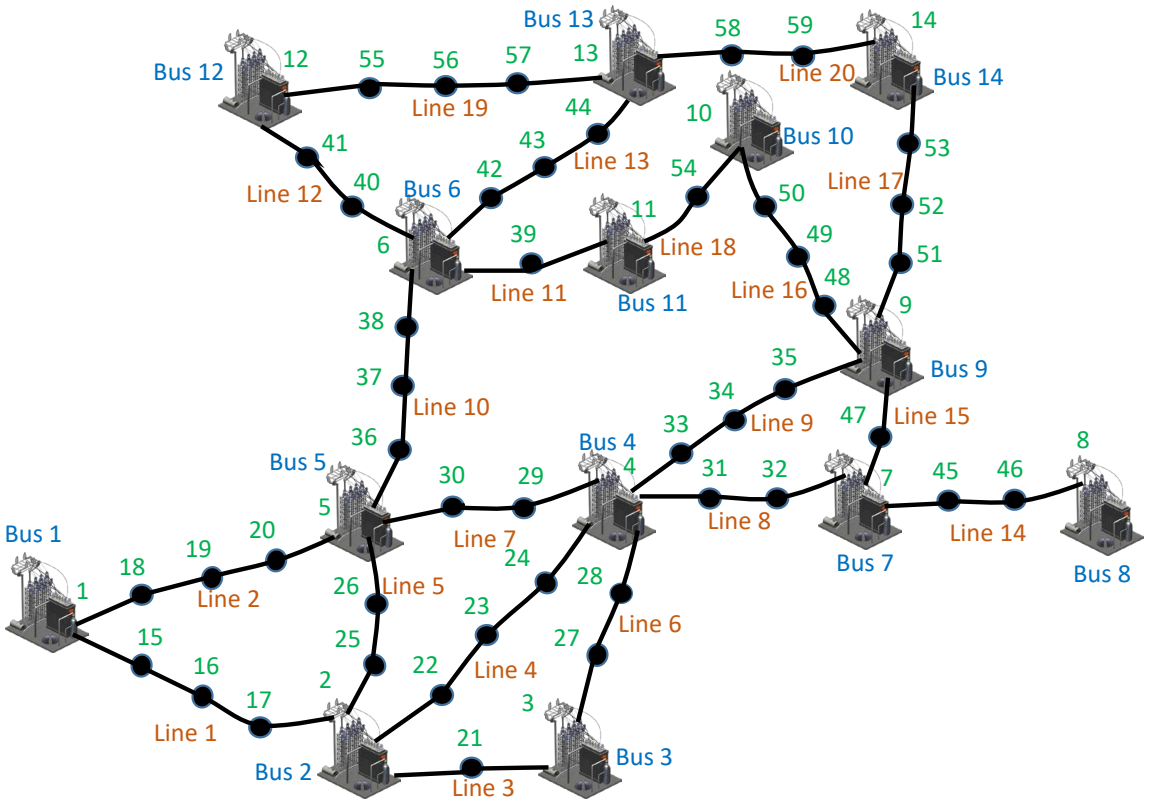


Figure 6.1: The example overhead transmission line infrastructure for the IEEE 14-bus test system.

(buses 1, 2, 3, 6, and 8). The rest of the buses are load buses. The transmission line information includes the end buses and the impedance (reactance) for each line.

Overhead transmission line infrastructure information includes the set of points that constitute each line. For an instance, line 1 is constituted of points 1, 15, 16, 17, and 2. The segment or link information more specifically provides information about the segment: the end points and the fuel cost ratio to fly from the first point to the second point in one time step. The ratio is taken over the cost of flying the same distance horizontally (climbing angle is zero). For example, the fuel cost required to fly the segment from point

Table 6.1: Case Study Input IEEE-14 bus system (Partial)

```
# Number of Buses, Lines, Points, and Segments, Segment Length in Time Units, Number of UAVs, Surveillance Period
14 20 59 65 1 5 91

# Load Information (Bus No, Load)
4 93.0
5 66.0
. . . . . .

# Generation Information(Bus No, Generation)
1 121.0
2 18.0
. . . . . .

# Transmission Line Info (From-Bus, To-Bus, Reactance)
1 2 0.05917
1 5 0.22304
2 3 0.19797
. . . . . .

# Maximum Criticality (PI Score) Distance
15%

# Line Point Set
1 15 16 17 2
1 18 19 20 5
2 22 23 24 4
. . . . . .

# Segments/Links (End Points, Fuel Cost Ratio)
1 15 1.0
15 16 0.95
16 17 1.0
. . . . . .

# UAV Properties (Initial Point, Stored Fuel, Fuel Capacity (Watt), Mileage (Fuel/Step), Hovering Cost (Fuel/Step))
10 1200 1500 15 3
5 600 1200 12 3
14 300 1500 15 6
. . . . . .

# Threshold Time between Two Consecutive Visits to a Point
25

# Resiliency Requirements (k, Threshold Time)
2 45

# Minimum Criticality Scores under Continuous Surveillance and Resilient Surveillance
80 50
```

1 to point 15 is regular, while the flying cost from point 15 to point 16 is less by 5%. The cost ratio is inverted if the flying direction is the opposite. The UAV information consists of a set of properties for each UAV, which includes its starting position, initial stored fuel, fuel capacity, and fuel costs per step for flying (when the climbing angle is zero) and hovering/loitering. It is worth mentioning that these property values are synthetic and driven from practical sources, particularly considering HyDrone UAVs [35], [36], [37].

As the surveillance requirements, we consider (i) continuous surveillance threshold as 25 time steps, (ii) 2-resilient surveillance and corresponding threshold as 45 time steps (*i.e.*, a point under resilient surveillance must be visited by $k + 1$ (3) different UAVs in 45 units), and (iv) the minimum scores under continuous surveillance and resilient surveillance respectively as 80% and 50%.

The solution to the corresponding formal model provides a *sat* result and provides the trajectory plan, including the refueling schedule. The result shows three criticality levels. According to the trajectory plan, 45 points are under continuous surveillance covering around 81% of criticality, while 28 points are under resilient surveillance covering over 50% of criticality. During the continuous surveillance, all the high (level 3) criticality points (13 points), three-quarters of the medium (level 2) critical points (17 points), and 65% of the low critical points are under coverage. The 2-resilient surveillance covers around 62% of the high critical points (8 points), 48% of the medium critical points (11 points), and 40% of the low critical points. Due to the distribution of the higher critical lines in the topology and a limited number of UAVs, many less critical points are also covered.

If we consider the UAV visits to a particular point, *e.g.*, point 3, the result shows that the point is visited at time step 7 by UAV 1 and so on as follows ([at, by]): [7, 1], [14, 4], [38, 2], [50, 3], [64, 1], and [74, 1]. These visits satisfy not only the continuous surveillance requirement but also the resilient surveillance condition, ensuring the visits of three (k = 2) UAVs followed (and including) from each visit within the threshold time. If we consider point 2, we find that it is only continuously surveilled ([5, 1], [8, 2], [25, 2], [27, 2], [37, 3], [48, 3], [56, 5], [81, 1], and [88, 2]) – the point is visited quite a few times but not by a required number of different UAVs within the threshold time. The output also includes the refueling schedule for the UAVs. According to this schedule, for example, UAV 2 goes for refueling at time 46 from point 9 while UAV 3 goes for refueling at time 16 from point 58.

# CHAPTER 7

# EVALUATION

We have evaluated the proposed surveillance plan synthesis model to analyze surveillance characteristics as well as its scalability. We have presented the results by ploting relevant graphs to justify the evaluation based on different criteria. We analyze the characteristic of surveillance by evaluating the minimum number of UAVs to perform surveillance satisfying the criticality coverage requirement. We consider the coverage requirement for resilient surveillance as half of that for the continuous surveillance. The evaluation is performed for different synthetic transmission infrastructures based on the IEEE test bus systems [10]. We consider the system size as the number of buses. The scalability of executing the proposed synthesis model is evaluated in terms of time by varying the same criticality coverage requirements for different transmission system sizes. We run our experiments on an Intel(R) Core(TM) i7-7700T @3.80 GHz machine with 16 GB memory.

## 7.1 Evaluation Results: Analysis of Characteristic

### 7.1.1 Impact of the Criticality Coverage Requirement on the Number of UAVs for Surveillance

The synthesis of the surveillance plan, *i.e.*, the trajectory of the UAVs and their refueling schedules, depends on the surveillance requirements, *i.e.*, the continuous surveillance

(data freshness) time threshold, resiliency specification and corresponding surveillance time threshold, and the criticality coverage requirements, along with the problem size. The number of UAVs required to perform the surveillance depends on satisfying all these properties. The tighter the constraints, the more UAVs are required to synthesize the surveillance plan.

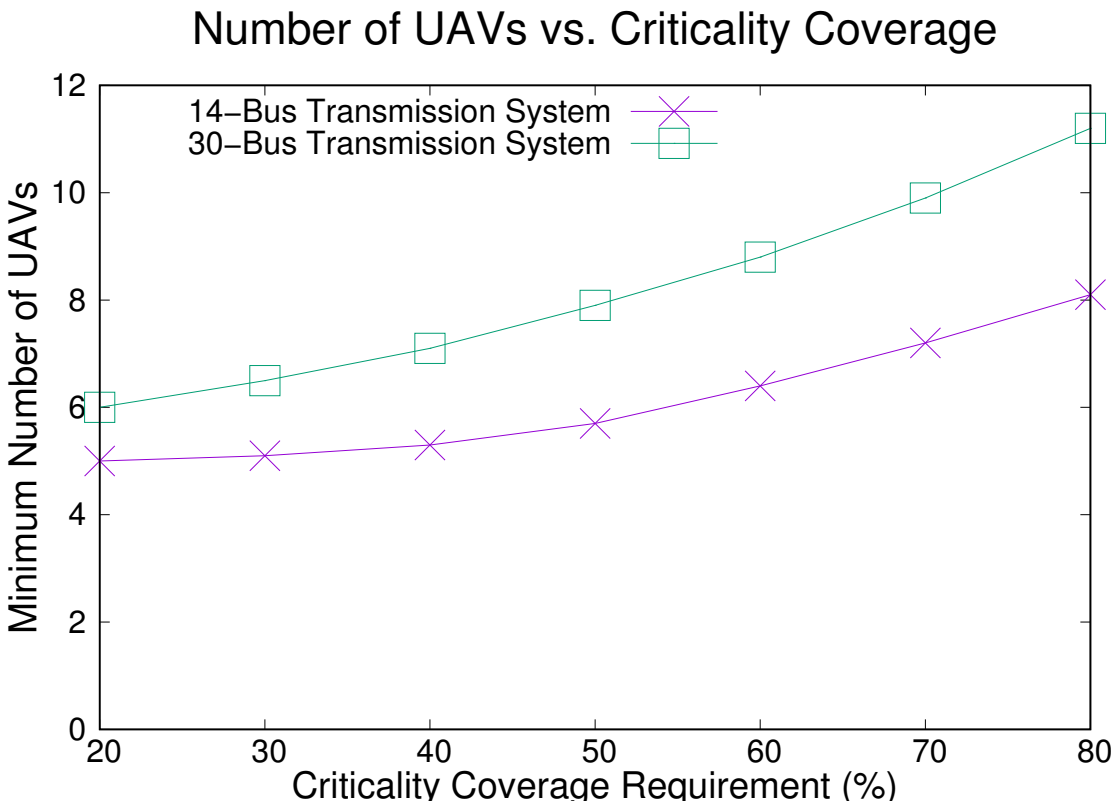


Figure 7.1: Impact of the criticality coverage requirement for continuous surveillance on the minimum number of UAVs to perform the surveillance.

We analyze the impact of the criticality coverage requirement by the surveilling UAVs in this respect. The analysis result is shown in Fig. 7.1. The graph shows that the number of minimally required UAVs increases with the increase in the coverage requirement. This

is because to cover higher criticality, a larger area (*i.e.*, a larger set of points) typically needs to be under surveillance. Therefore, a larger fleet of UAVs is necessary. With the problem size (*e.g.*, the number of buses), this number increases further. As the figure shows, the minimum number of UAVs required for surveillance in the case of the 30-bus system is larger than that of the 14-bus system.

### 7.1.2 Impact of the Grid Size on the Criticality Surveillance

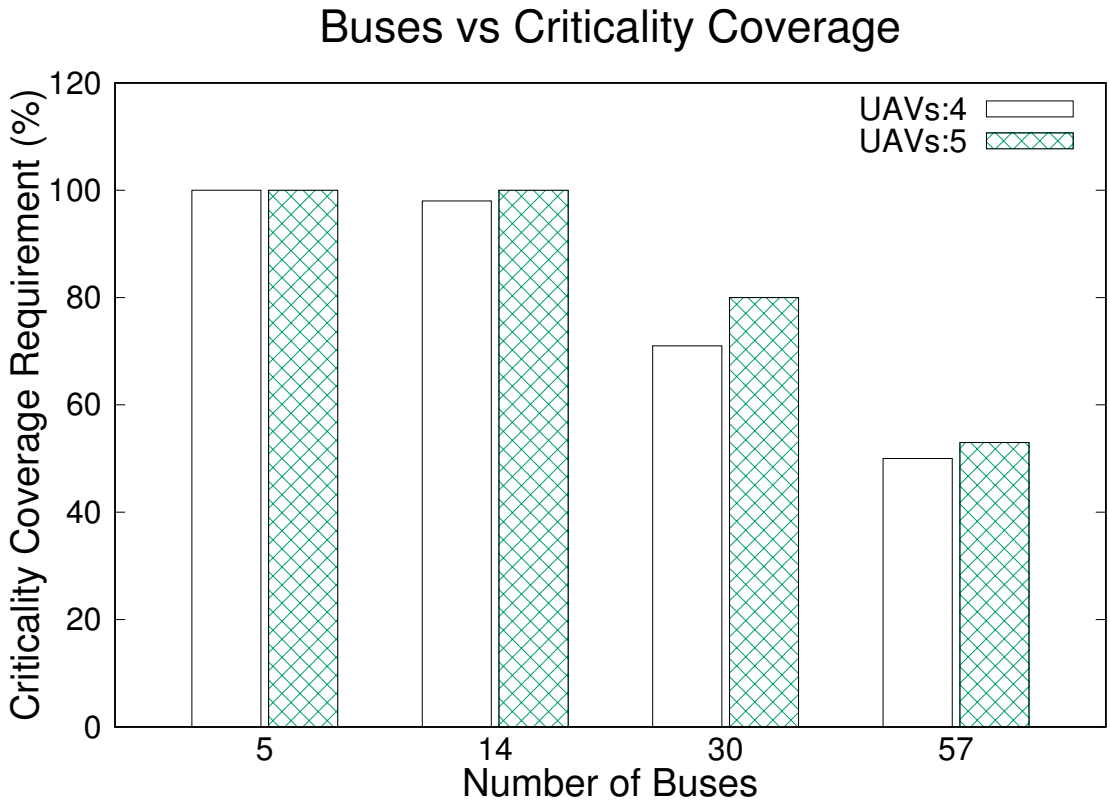


Figure 7.2: Impact of the grid size (*i.e.*, the number of buses) on the criticality surveillance.

We analyze the impact of the grid size (*i.e.*, the number of UAVs) on the maximum

criticality coverage by the surveilling UAVs. The analysis result for two different numbers of UAVs is presented in Fig. 7.2.

The result demonstrate that for a specific number of UAVs, the criticality coverage is limited and with the increased size of the transmission system (*i.e.*, the number of buses) the criticality coverage reduces. The is because a larger system has a larger set of surveillance points, often more critical points, and these points are distributed in a wider infrastructure. Therefore, for a increased system size, a particular set of UAVs cannot but surveilled a reduced part of the system's criticality. With the number of UAVs, the criticality coverage increases. As the figure shows, a set of 5 UAVs can perform surveillance covering a larger criticality than a fleet of 4 UAVs, especially when the problem size is larger.

### 7.1.3 Impact of the Grid Size on the Required Number of UAVs

As we can see in Fig. 7.2 that a larger set of UAVs can provide more criticality coverage, we analyze the impact of the grid size (*i.e.*, the number of UAVs) on the minimum number of UAVs required to cover a particular criticality coverage. Here, we consider 60% and 30% as the criticality score coverage requirements for continuous surveillance and resilient surveillance. The analysis result for two different resilient surveillance intervals is presented in Fig. 7.3. In both cases, the continuous surveillance interval is 25 time steps and k = 1.

The graphs in the figure demonstrates that larger grid (a larger number of buses) requires a higher number of UAVs to cover the required criticality score because, as we

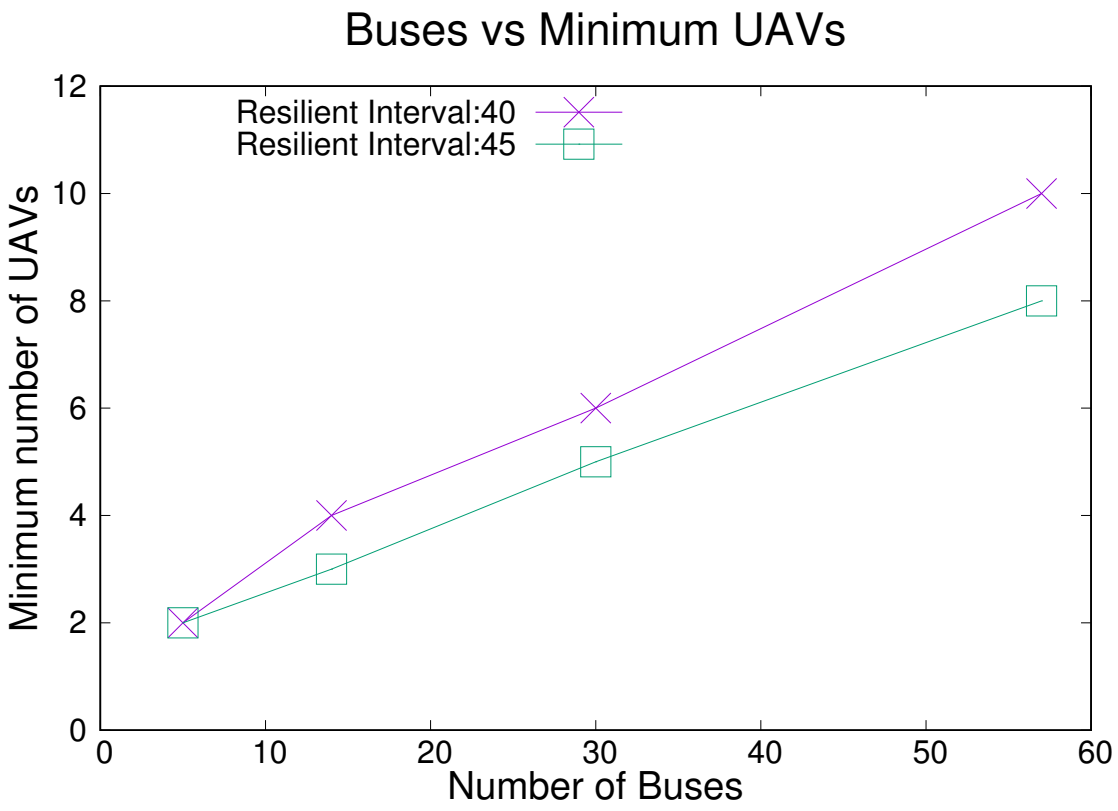


Figure 7.3: Impact of the grid size (*i.e.*, the number of buses) on the resilient surveillance.

discusses earlier, a larger system has a larger set of surveillance points distributed in a wider infrastructure.

We can also observe in the figure the impact of resiliency surveillance interval on the number of UAVs. As the figure shows, a larger interval requirement (*i.e.*, 45 steps) often needs a smaller number of UAVs than a smaller one (*i.e.*, 40 steps). This is because a larger interval allows a longer time frame to visit a point k + 1 (2) times, which often reduce the number of UAVs required to achieve the criticality surveillance score. The impact is also more prominent in a larger grid size.

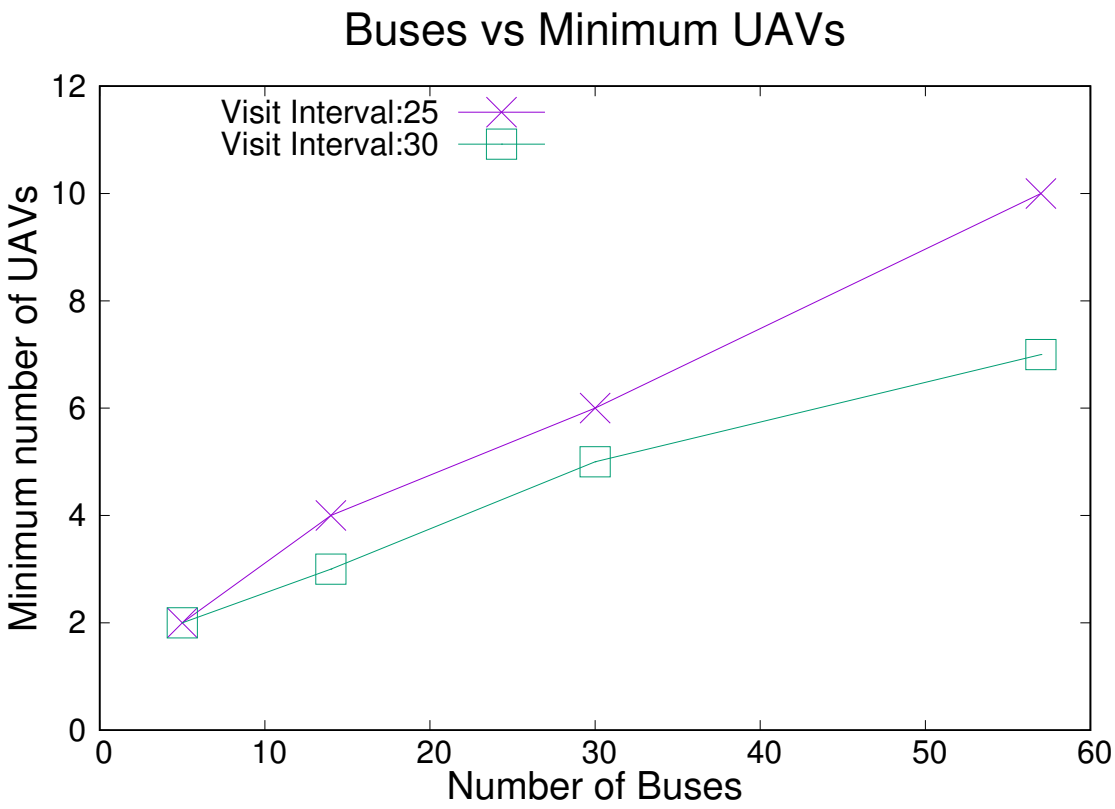


Figure 7.4: Impact of the grid size (*i.e.*, the number of buses) on the surveillance period.

We have also used 60% and 30% as the criticality score coverage requirements for continuous surveillance and resilient surveillance to find a relation with the surveillance interval. Fig. 7.4 shows the impact of increasing problem size (*i.e.*, number of buses) to the number of UAVs required. We have modeled this with two different visit interval requirements as 25 steps and 30 steps. As it is shown, with a less visit interval, number of UAVs are higher because of the continuous surveillance task. For both of the cases, we have fixed the resilient requirement as 40 steps during out evaluation.

## 7.2 Evaluation Results: Analysis of Scalability

### 7.2.1 Impact of the Criticality Coverage Requirement

Fig. 7.5 shows the execution/solving time of the proposed formal model with respect to the criticality coverage requirement for 14-bus and 30-bus transmission systems.

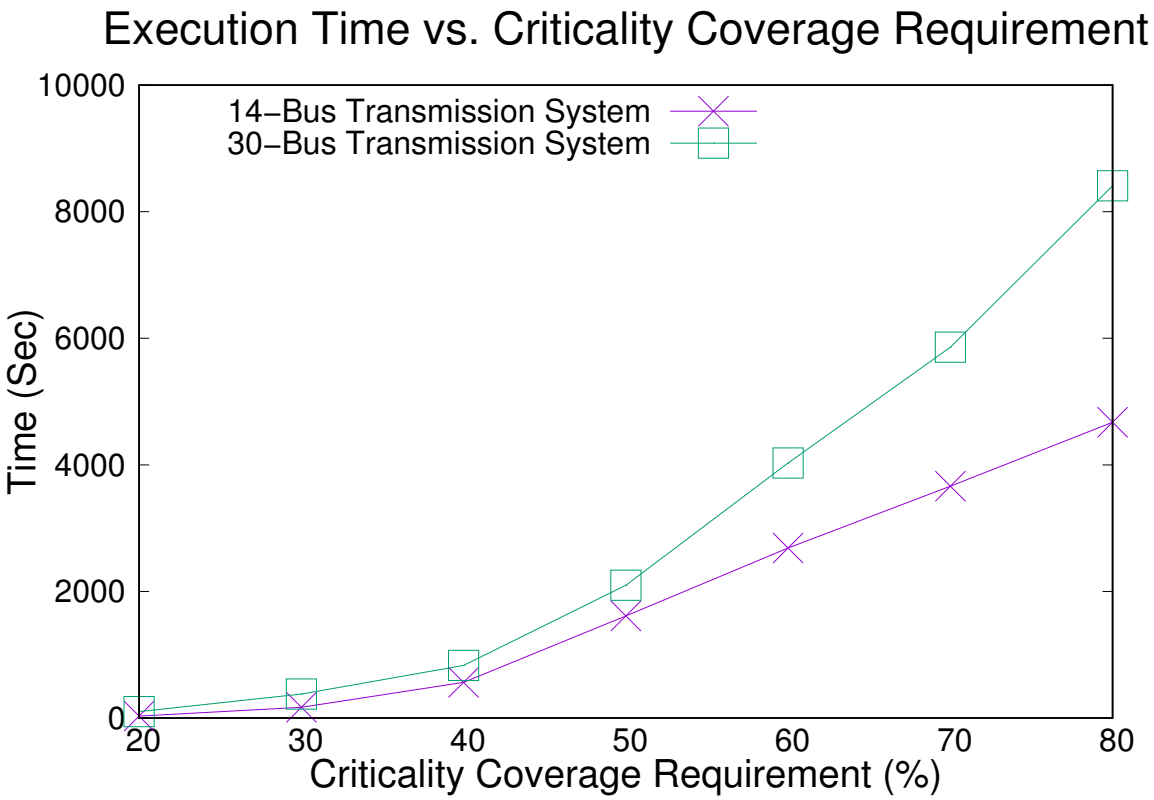


Figure 7.5: Impact of the criticality coverage requirement for continuous surveillance.

As shown by the graphs, with the increase in the coverage requirement, the time to solve the model grows. The tighter the requirement, as discussed earlier, a larger area needs to be covered, which increases the execution time. Moreover, the more the requirement

become close to the maximum possible coverage for a particular set of UAVs, the search space increases rapidly, which increases the execution time. When a larger set of UAVs is required for the surveillance, the size of the model (i.e., the number of variables and assertions/clauses) also expands. A larger model requires an increased, often exponentially high, solving time.

### 7.2.2 Impact of the Number of UAVs

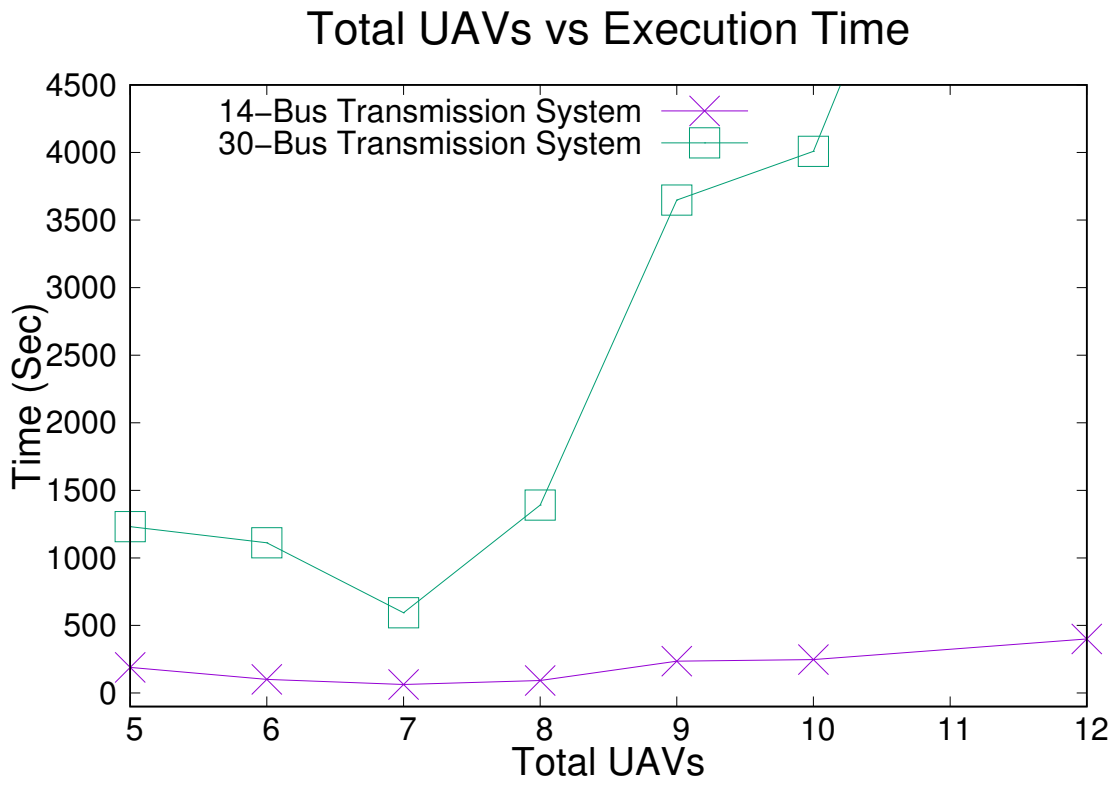


Figure 7.6: Impact of the size of the UAV fleet (*i.e.*, the number of UAVs).

Here, for a particular problem size and a surveillance requirement, we evaluate the

impact of the number of UAVs on the solving time. The evaluation is performed with respect to three problem sizes, 5-bus, 14-bus, and 30-bus systems. Fig. 7.6. presents the result.

We can observe that initially with the increase in the number of UAVs the execution time reduces. However, after a point, the execution time grows rapidly. This is because, initially an increased number of UAVs eases the search for a solution. However, as the number of UAVs grows, the number of clauses in the model increases superlinearly, which in turn increases the execution time after a point.

### 7.2.3 Impact of the Resiliency Requirement ($k$)

We analyze the impact of the $k$, the resilient surveillance requirement on the 14-bus system. We consider two fleets of UAVs on various $k$ values for a particular set of surveillance interval and score requirements. The evaluation result is presented in Fig. 7.7.

As the result shows with the $k$, the model execution time increases. Since a point under resilient surveillance needs to be visited by $k + 1$ UAVs within a specific threshold time, a larger $k$ tightens the solution space, thus increasing the searching time.


Figure 7.7: Impact of the selection k for resilient surveillance on the formal model execution time.

### 7.2.4 Impact of the Surveillance Interval

The solving time of the proposed formal model depends on the surveillance time interval (between two consecutive visits) threshold (maximum). We perform the evaluation with respect to both continuous surveillance and resilient surveillance. The simulation is done on two problem sizes: 14-bus and 30-bus transmission systems, for different surveillance thresholds. The rest of the input, including continuous and resiliency surveillance score requirements, remains the same. The corresponding results are shown in Fig. 7.8 and

Fig. 7.9.

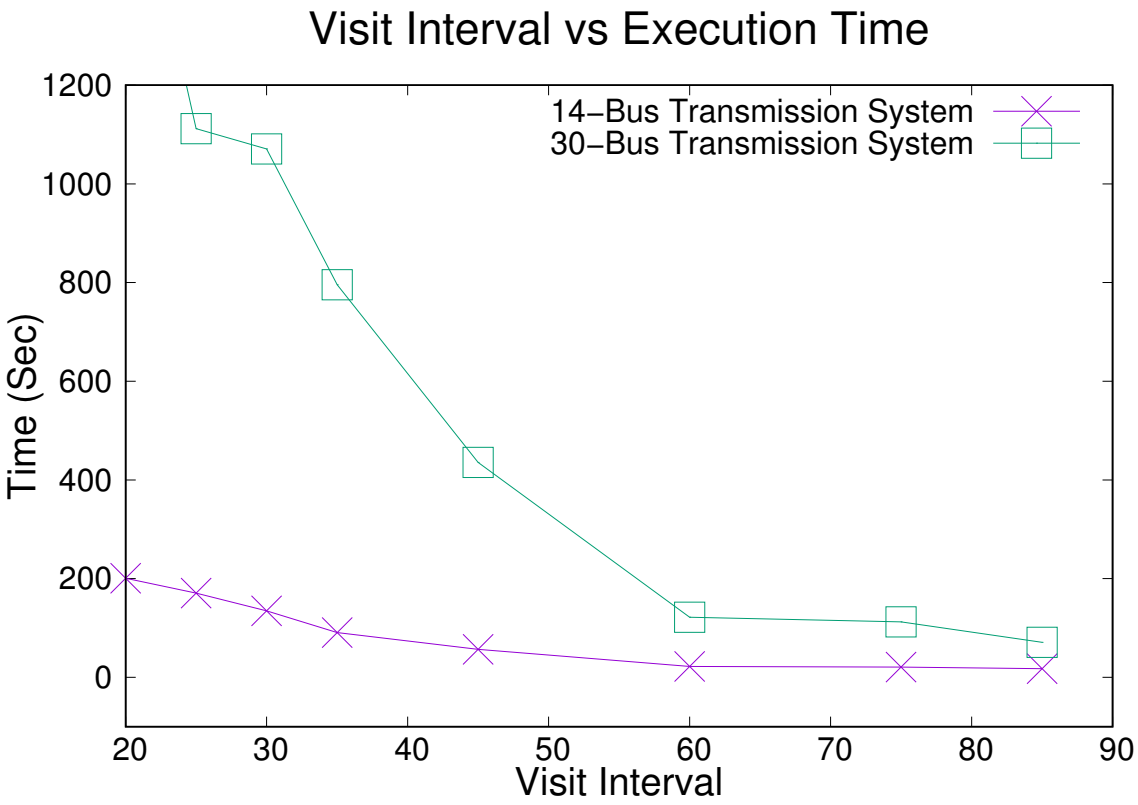


Figure 7.8: Impact of the time interval for the continuous surveillance.

We can observe that the relaxed is the surveillance interval requirement (*i.e.*, with the increase in the threshold), the time to solve the model reduces. As the figure shows, the impact is more prominent for a larger transmission system (*i.e.*, the 30-bus system) than a smaller system (*i.e.*, the 14 bus system).

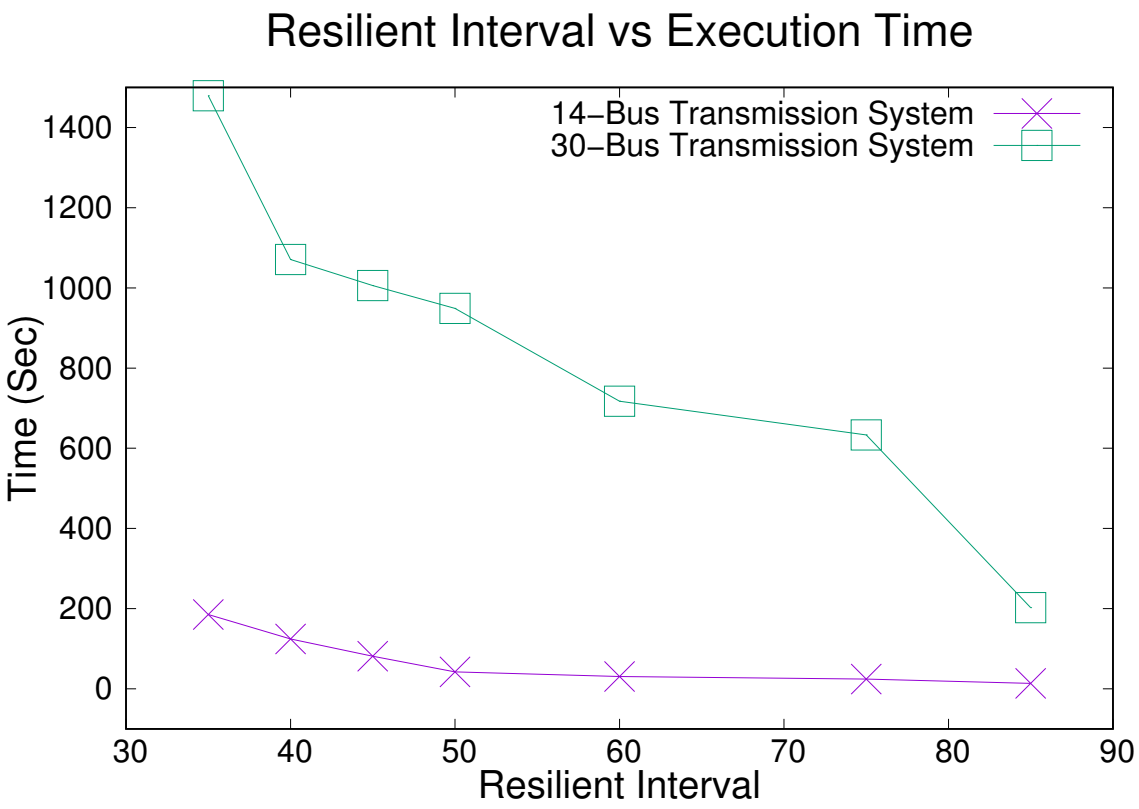


Figure 7.9: Impact of the time interval for the resilient surveillance.

### 7.2.5 Impact of the Surveillance Period

The impact of surveillance period (in time steps/units) on the execution time is presented in Fig. 7.10 and, as the graph shows, the execution time grows rapidly with the period.

Each time step is associated with a number of clauses. As a result, if the surveillance time expands, the number of clauses grows, which ultimately increases the model solving time.

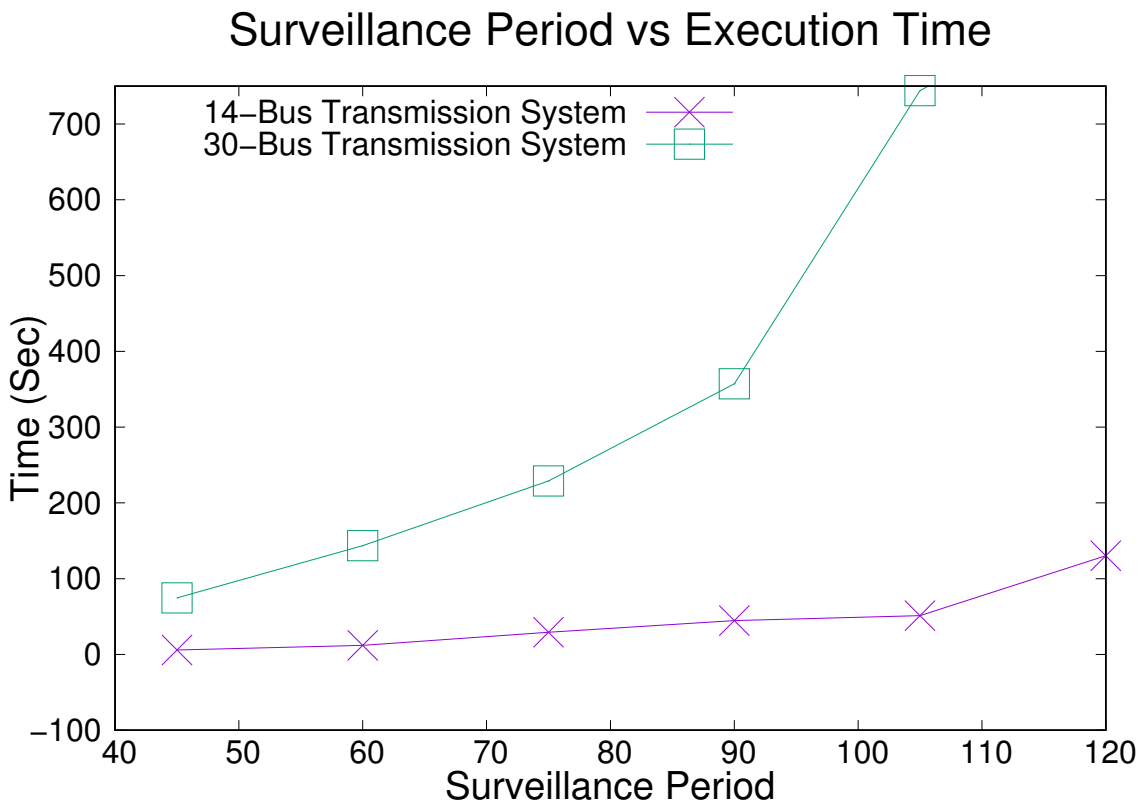


Figure 7.10: Impact of the surveillance period on the formal model execution time.

### 7.2.6 Impact of the Criticality Distribution over Lines

The required number of UAVs to preform the surveillance operation depends on the scenario how the critical weights are distributed over the transmission lines. Fig. 7.11 shows the impact of critical weight distribution over the transmission lines to the number of UAVs.

In this graph, the critical weights are distributed in two cases where top 30% and 50% lines are taken into account for the analysis. Here, top 30% or 50% refers to the portion from the total transmission lines which possesses different distribution of criticality over the lines. If we increase the criticality within each portion (top 30% or 50%), the

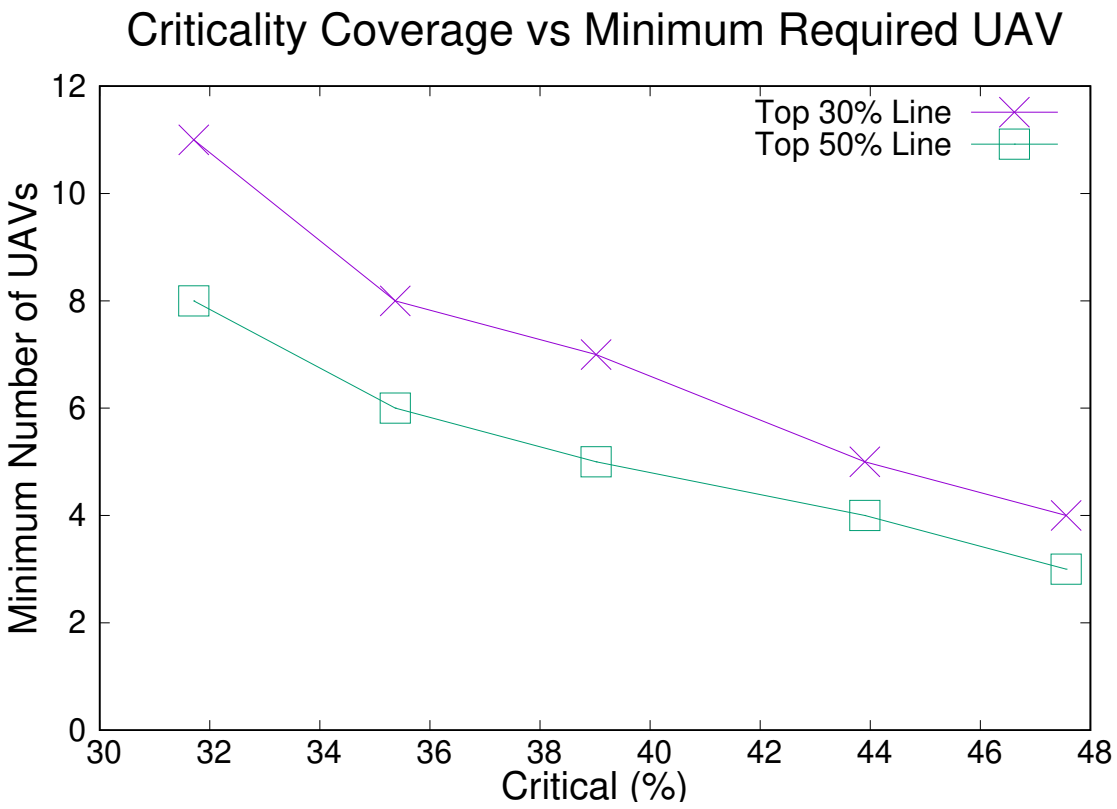


Figure 7.11: Impact of the criticality for line possession on the minimum number of UAVs for surveillance.

number of requires UAVs decrease. It is because, the continuous surveillance clauses are now satisfied to the model requirements based on the higher critical distributions which needs less number of UAVs to fulfill the coverage for the overall power transmission lines.

# CHAPTER 8

# GRAPHICS (UNITY3D) SIMULATION

The continuous surveillance model results in a routing path for the UAVs. The designated trajectories for each of the UAV satisfiy the requirements of the refuleing capacity, critical scoring capability of the transmission lines as modeled, continuos monitoring schema within the threshold time unit as assigned, and movement of the UAVs' over the power line segment by point-to-point manner. In this regard, we have developed a graphical simulation of the overall surveillance task based on the outcome of the formal method/SMT [38] model.

## 8.1 Background

Unity Technologies developed Unity engine as a cross-platform framework released in June 2005 written in $C\#$ as a primary programming language [39]. This framework provides a real-time engine providing the capability to the designer for creating three dimensional along with two-dimensional simulation model for several platforms (*e.g.*, Windows, iOS, Android).

Unity is a widely used game development and simulation design tool now a days. Xu *et al.*, introduces a Fire Dynamic Simulation (FDS) model for high rise buildings developed for Unity3d engine [40]. The work has simulated the smoke and fire model to visualize the

accurate effect of smoke particles to give a real and unbiased view to the user.

Yang *et al.*, describes the mentionable impact of simulated software concept in the case of drivers' training task [41]. The authors have used 3D design module of Unity to develop a system platform introducing the training process of driving through the simulation software. The simulation it can improve the training efficiency of people reducing the cost of venues and vehicles as well. A vehicle simulation platfprm has been developed to verify a virtual reality system using Unity3D in [42]. In this simulation design, virtual structures (*e.g.*, roads, cars, pedestrians, automobile bodies) are generated first. Then different conceptual implementation such as weather condition, blockade, blind spots are created to provide a driving simulation. The model then verifies the effects of environmental factors with the driver's test scenario for the improvement of sensor depth.

F*ü*rst implemented a system module for building control to simulate the actuations and schedules integrating the physical and the virtual world [43]. This simulation has been done using Unity3D with sMAP (Simple Measurement and Actuation Profile). The simulation mode input can be taken from user to define the rules as BIM (Building information modeling) to run the simulation and see the feedback on building's control parameters (*i.e.*, temperature, energy consumption). Meng *et al.*, presents the idea of miniature UAV based navigation system with GPS-denied environmental control by Unity3D [44]. Motivated by the concept of verification of designed algorithm, the simulation work by the authors ensures the identification of any problems. This potential finding from the Unity3D simulation can help for the swift operational verification before the real flight and test the accuracy of the

physical implementation as well. Their proposed approaches have integrated robot operation system (ROS) with Unity engine to provide smooth handling of multi-UAV navigation in real-time manner along with the sensor data processing directly online.

## 8.2 Development Design

To develop the simulation we have used Unity3D [34] design framework to model and map the routing plan of the UAVs on a sample IEEE-14 bus test system scenario. Fig 8.1 illustrates the design of the simulation development. The model maps the solution results to the Unity3D simulation based in the geographical position specified as inputs for the program.

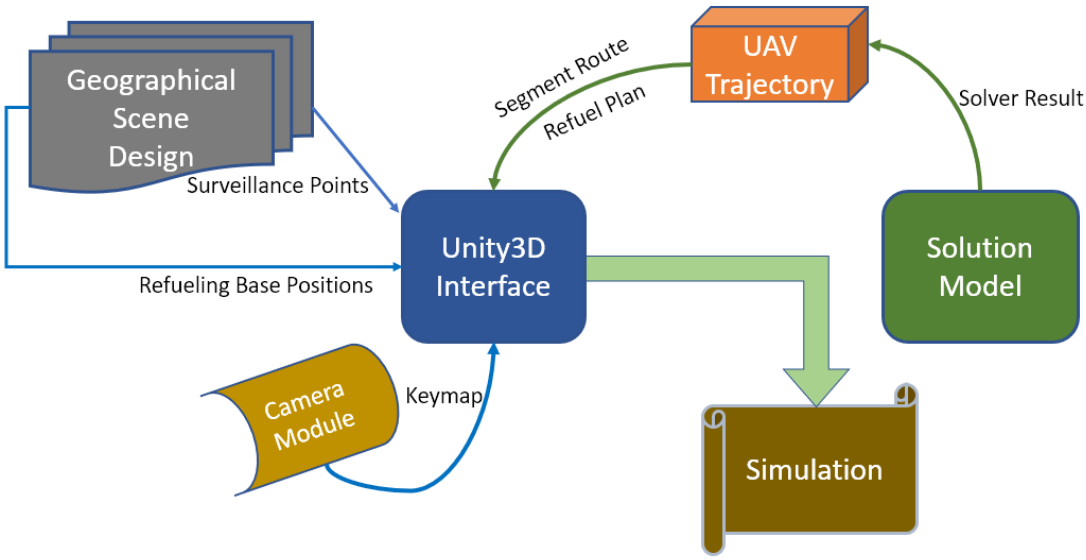


Figure 8.1: Unity3D Simulation design model.

In order to make the simulation scene flexible and convenient to the viewer a **camera**

Table 8.1: Camera Movement and Keymap for the Simulation Scene.

| Keymap | Movement Action |
|---|---|
| Q | Look Up |
| E | Look Down |
| Z | Rotate Left |
| X | Rotate Right |
| ↑ | Go Forward |
| ↓ | Go Backward |
| → | Go Right |
| ← | Go Left |

feature has been implemented along with the default view configuration of the Unity3D library. This *camera* enables the viewer to use movements listed in Table 8.1 with the keymap for the designed UX module. Apart from the keyboard buttons, we have added mouse acitivity feature to the camera as well. If there is no mouse click, taking the cursor to any corner of the screen will let the scene go towards the direction in same height. The mouse 'scroll' button works as a $zoom - in/zoom - out$ view of the scene.

### 8.3 Simulation Example

Fig 8.2 presents an illustration of the graphics scene of an example power grid with a 3D plane-based (coordinate system *i.e.*, $X, Y, Z$). To implement the graphical demonstration, we have used a Nvidia GeForce GT 640M GPU with 2GB of VRAM for the smooth rendering of the UAV movement during the simulation.

To design the power grid scene for an IEEE-14 bus test system, we have loaded the substation (bus) positions based on the real topology (*i.e.*, inter-connection between buses). To map these substations geographically in the scene, we have used cartesian three dimentional coordinates (3-D) plane as for the input segments. This design for the surveillance

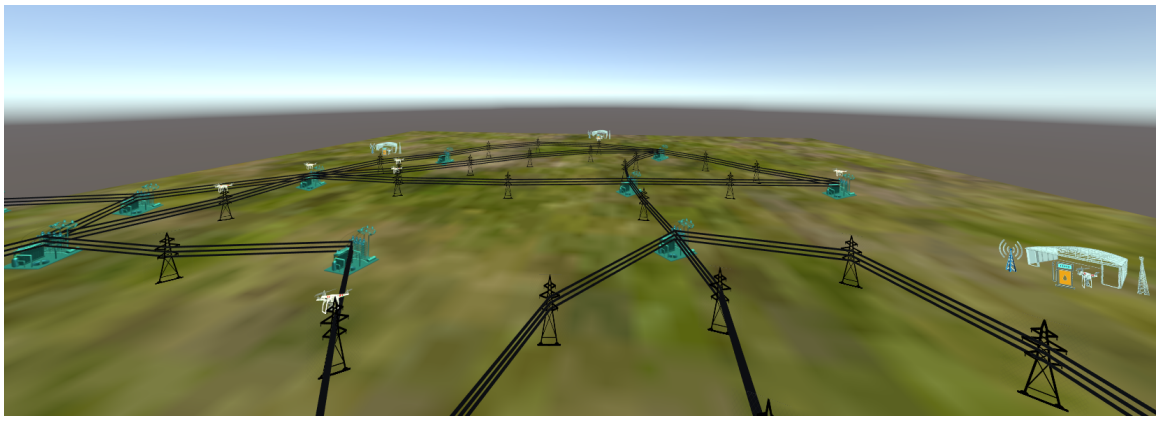

Figure 8.2: Unity3D Simulation snapshot for the illustration example in Fig 6.1.

model test case can be found in Fig 6.1 as an illustration for the segmentation formation. As segments are points on the same transmission lines according to our implementation, in the simulation scene, we have treated these *segments* as *towers* of a power grid system. In addition to that, *buses* in the system are represented as *substations* which may connect multiple *towers* to create the electrical *cables* (transmission lines) for the developed scene in Unity3D.

We have introduced the refueling bases for the UAVs in our design to provide resilient and continuous surveillance task. We assumed that the UAVs have followed standard and conventional technologies to calculate amount of fuel available, communicate with *nearby* bases and pre-mapped the position coordinates for the refuleing bases.

The refueling bases are placed in our simulation module in a way that it is colser to a group of *substations* with an approximately equal cartesian distances from each of them. For example, if a group is formed with some adjacent buses (*e.g.*, bus 1, 5, 6 ,12), one refuleing base will be dedicated for this group as *nearby* based on our scene formation.

The significance of this concept we have used is to provide the UAVs flexibility to go to the closest bases for refueling if those are on any of the buses/points of the group. It actually increases the routing capability of an UAV to cover/go through more of surveillance points (*towers*/*substations*). As it will go to the closest bases assigned for the points it is over, this approach will reduce the time steps to the refuel base. Also, returning to the main surveillance points after refueling becomes more empowered and realistic like an actual, physical UAV in surveillance action.

We have designed our surveillance task so that our assumption holds in a way that any UAV can only go '**over**' the transmission lines. Hence, if there is no *cable* between two *towers* or *substations* in the scene, a UAV will not route towards the path as per our design; which clarifies to the fact that the movement of a UAV must be to follow the *cables*, if it wants to reach any surveillance point in the scene. However, there exists one exception to this statement in case of refueling. While going towards a refueling base from a point, it can choose any cross directional disctance unlike the regular overhead surveillance task.

The developed simulation is released as a standalone application (*e.g.*, a windows executable *.exe*). Once compiled for the IEEE-14 bus test system, a scene is developed based on the coordinate mapping and refuel base station positioning numbered after the last *id* of the point-set. The refuel bases are grouped and taken as an input from a text file for the Unity3D program.

A sample refuel bases are shown in Table 8.2 to form groups of surveillance points for the test case scenario. Here, we have considered four refueling base stations for a case

Table 8.2: Refueling Base Groups for IEEE-14 Bus Test System Scene

```
# Refuel Base No., Associated Points (each line for one base station)
1, 2 3 4 5 6 7 8 9 15 16 17 19
2, 10 11 12 13 14 20 21 22
3, 23 24 25 26 29 30 31 34 35 41 42
4, 1 18 27 28 32 33 36 37 38 39 40 43 44 45 46
```

Table 8.3: UAV Route From Solver Result

```
# UAV Id, Routing Points
1, 1 1 1 4 5.....13 12 2
2, 10 11 10.....13 12 13
3, 5 4 1 6.....5 4 4 4
4, 13 13 12.....11 10 9
```

study shown in Fig 6.1. Each line of the input file from the Table 8.2 holds each refuel base respectively (numbered as 1, 2, 3, 4).

We have another input taken for the simulation program which helps to dynamially model our surveillance plan resulted from the solution model. This is also a text file containing all the routing points for each of the UAVs. This input helps to simulate the routing path for each of the UAVs based on different solution outcome from different constraints for the continuous surveillance model. Table 8.3 represents a sample routing plan with four UAVs.

The power grid surveillance space is usually a large area which contains multiple *substations/towers*. As the geographical position of any power grid is a fixed positioning space in real sceanrio, we have designed the simulation UX accordingly. To summarize, it has the fixed position of the substations and the towers but different UAV movements for each surveillance plan according to the satisfiable solution of the proposed solver design.

# CHAPTER 9

# CONCLUSION AND FUTURE WORKS

Overhead power transmission lines in a smart grid require regular assessment for reliable and uninterrupted operation, especially considering physical (potential) damages due natural calamities, aging factors, technical errors, or physical attacks. The emergence of UAV technology provide the opportunity of keep this critical infrastructure under continuous surveillance. In this chapter, we are concluding our thesis work by summarizing the solution approaches to the continuous surveillance problem. We also discuss the novel take away from the work for future extension and scope for improvement.

## 9.1 Summary

In this thesis work, we have proposed a formal framework that synthesizes the trajectory plan as well as the refueling schedules for a given set of UAVs. This UAV routes satisfying the refueling schedule will perform a continuous surveillance over the power transmission lines. The surveillance will provide the ability to monitor various critical lines specified from the $n$-1 contingency analysis. We have implemented the proposed solution and evaluated the tool to analyze surveillance characteristics as well as its scalability on synthetic data. We have applied the solution model for standard IEEE test bus systems to evaluate the effectiveness of our approaches. Apart from time-coinstrained monitoring, our

design has satisfied the resiliency requirements for the UAVs as well. The resiliency surveillance requirement for a point ensures that if $k$ UAVs fail or become compromised still there is a UAV to collect the data at the point no later than a threshold time. This combined module ensures the continuous and the resilient trajectory plans for the surveillance UAVs. Finally, we have developed a Unity3d based UX simulation tool to present the work in a graphics animation. This simulation maps the successful outcome from the solution model to an animated scene. This scene provides a interface for the user to see, which contains the ultimate movements for the surveillance UAVs over the designated trajectory routes based on the solution ($sat$) result.

## 9.2 Scope For Future Works

As summarized above, the results of the continuous resilient surveillance design open up the door to furture investigations in multiple directions.

**Collaborative UAVs:** Relying on timely information sharing and real-time communication with the base station through the fleet member of a UAV swarm, collaborative UAVs are an emerging technology. We would like to extend this research to consider collaborative UAVs so that it can perform the surveillance for vast area of power-grid infrastructure. In this manner, the on-going technology of $master - follower$ approach may help to build a communication infrastructure-less scenario. With the collaborative UAV concept, we will consider the communication overhead analysis and encryption methodology for secure data delivery to the control station. With real-time data transfer capability, we will try to design

a robust and wide trajectory planning to cover the surveillance for large bus network such as IEEE-118, IEEE-300 test systems.

**Web-based simulation panel:** We have the vision to develop a feasible web-based simulation infrastructure as an extension of our current simualtion tool in Unity3d. This framework will work for a real-time SMT solver for the scenarios provided by users. The 'solver' module of this open source platform will run the solution for the specified input. Then, the UX interface will map the solution with the graphics implementation. This graphical design will run the animation on the web according to the criteria (*i.e.*, number of UAVs, threshold-time, problem size), which are selected as per the solution, showing the UAV trajectories and movements.

# REFERENCES

[1] Khawaja, A., Huang, Q., and Khan, Z. "Monitoring of Overhead Transmission Lines: A Review from the Perspective of Contactless Technologies." volume 18, 12 2017.

[2] Ma, Lili and Chen, YangQuan. "Aerial surveillance system for overhead power line inspection." *Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State Univ., Logan, Tech. Rep. USU-CSOIS-TR-04-08 (September 2000)*, 2004.

[3] Chen, Y. and Xu, Y. "Detection and localization of untwisted strands in transmission lines using cascaded shape filtering and color filtering." In "IEEE Workshop on Signal Processing Systems (SiPS)," pages 1–6, Oct 2015.

[4] Srinivasan, S., Latchman, H., Shea, J., Wong, T., and McNair, J. "Airborne traffic surveillance systems: video surveillance of highway traffic." In "ACM Workshop on Video Surveillance & Sensor Networks," pages 131–135, 2004.

[5] Trasviña-Moreno, Carlos A, Blasco, Rubén, Marco, Álvaro, Casas, Roberto, and Trasviña-Castro, Armando. "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring." *Sensors*, volume 17, no. 3, page 460, 2017.

[6] Li, Linxin. "The UAV intelligent inspection of transmission lines." In "Proceedings of the 2015 International Conference on Advances in Mechanical Engineering and Industrial Informatics," pages 1542–1545, 2015.

[7] Abur, A. and Exposito, A. G. *Power System State Estimation: Theory and Implementation*. New York, NY: CRC Press, 2004.

[8] Wood, Allen J. and Wollenberg, Bruce F. *Power Generation, Operation, and Control, Third Edition*. Wiley, 2013.

[9] de Moura, Leonardo and Bjørner, Nikolaj. "Satisfiability Modulo Theories: An Appetizer." In "Brazilian Symposium on Formal Methods," , 2009.

[10] of Washington, University. "Power Systems Test Case Archive." `http://www.ee.washington.edu/research/pstca/`.

[11] "Economic Benefits of Increasing Electric Grid Resilience to Weather Outages, August 2013."
URL `http://energy.gov/sites/prod/files/2013/08/f2/Grid20Resiliency20Report_FINAL.pdf`

[12] Campbell, Richard J. "Weather-related power outages and electric system resiliency." Congressional Research Service, Library of Congress Washington, DC, 2012.

[13] Vock, Daniel C. "The Pact Changing How Governments Respond to Disaster."
URL `http://www.governing.com/topics/transportation-infrastructure/gov-emergency-management-local-federal-fema-states.html`

[14] Criss, Doug. "Puerto Rico's power outage is now the second-largest blackout on record, April 16, 2018."
URL `https://www.cnn.com/2018/04/16/us/puerto-rico-blackout-second-largest-globally-trnd/index.html`

[15] Latchman, Haniph A, Wong, Tan, Shea, John, McNair, Janise, Fang, Michael, Courage, Ken, Bloomquist, David, and Li, Irene. "Airborne Traffic Surveillance Systems: Proof of Concept Study for the Florida Department of Transportation.", 2005.

[16] Jaimes, A., Kota, S., and Gomez, J. "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAV (s)." In "IEEE International Conference on System of Systems Engineering," pages 1–6, 2008.

[17] Montambault, Serge and Pouliot, Nicolas. "Inspection robot capable of clearing obstacles while operating on a live line." *Retrieved November*, volume 6, page 2008, 2007.

[18] Montambault, S., Pouliot, N., Lepage, M., Michaud, J., Latulippe, P., and Comte, C. "Remote-controlled vehicle designed to be mounted on a support and capable of clearing an obstacle.", June 2009. US Patent 7,552,684.

[19] Li, Z., Liu, Y., Hayward, R., Zhang, J., and Cai, J. "Knowledge-based power line detection for UAV surveillance and inspection systems." In "International Conference Image and Vision Computing New Zealand," pages 1–6, 2008.

[20] Pagnano, Angelo, Höpf, Michael, and Teti, Roberto. "A roadmap for automated power line inspection. Maintenance and repair." *Procedia Cirp*, volume 12, pages 234–239, 2013.

[21] Luque-Vega, Luis F, Castillo-Toledo, Bernardino, Loukianov, Alexander, and González-Jiménez, Luis E. "Power line inspection via an unmanned aerial system based on the quadrotor helicopter.", 2014.

[22] Semsch, E., Jakob, M., Pavlicek, D., and Pechoucek, M. "Autonomous UAV surveillance in complex urban environments." In "IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology," pages 82–85, 2009.

[23] Lim, Gino J, Kim, Seonjin, Cho, Jaeyoung, Gong, Yibin, and Khodaei, Amin. "Multi-UAV pre-positioning and routing for power network damage assessment." *IEEE Transactions on Smart Grid*, 2016.

[24] Deng, Chung, Wang, Shengwei, Huang, Zhi, Tan, Zhongfu, and Liu, Junyong. "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications." *J. Commun*, volume 9, no. 9, pages 687–692, 2014.

[25] Birnbaum, Zachary, Dolgikh, Andrey, Skormin, Victor, O'Brien, Edward, Muller, Daniel, and Stracquodaine, Christina. "Unmanned aerial vehicle security using behavioral profiling." In "Unmanned Aircraft Systems (ICUAS), 2015 International Conference on," pages 1310–1319. IEEE, 2015.

[26] Abbaspour, Alireza, Yen, Kang K, Noei, Shirin, and Sargolzaei, Arman. "Detection of fault data injection attack on uav using adaptive neural network." *Procedia computer science*, volume 95, pages 193–200, 2016.

[27] Li, Z., Liu, Y., Walker, R., Hayward, R., and Zhang, J. "Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform." *Machine Vision and Applications*, volume 21, no. 5, 2010.

[28] Rahman, Mohammad Ashiqur, Duan, Qi, and Al-Shaer, Ehab. "Energy efficient navigation management for hybrid electric vehicles on highways." In "Cyber-Physical Systems (ICCPS), 2013 ACM/IEEE International Conference on," pages 21–30. IEEE, 2013.

[29] Treinen, R. "Shift factors: Methodology and example." *Presentation, CRR educational class*, volume 5, 2005.

[30] "Contingency ranking and selection." nptel.ac.in/courses/108107028/module5/ lecture6/lecture6.pdf. Online: Accessed on June 24, 2018.

[31] Han, Jiawei, Pei, Jian, and Kamber, Micheline. *Data mining: concepts and techniques*. Elsevier, 2011.

[32] Zhang, Yong, Yuan, Xiuxiao, Li, Wenzhuo, and Chen, Shiyu. "Automatic Power Line Inspection Using UAV Images." *Remote Sensing*, volume 9, no. 8, 2017.

[33] "Z3 Theorem Prover." Microsoft Research. `https://github.com/Z3Prover/z3/wiki`.

[34] "Unity 3D, a cross-platform game engine, developed by Unity Technologies,." https://unity3d.com/; https://docs.unity3d.com/Manual/index.html. Online: Accessed on July 20, 2018.

[35] Jameson, Terry. "A Fuel Consumption Algorithm for Unmanned Aircraft Systems." http://www.arl.army.mil/arlreports/2009/ARL-TR4803.pdf, May 2009. Army Research Laboratory (ARL-TR-4803).

[36] "HyDrone 1550 Hydrogen-powered." http://en.mmcuav.com/ProductsSt/196.html;https://www.uavs
long-range-surveillance-drone/. MicroMultiCopter Aero Technology Co. Ltd.

[37] "The Perimeter Drone." https://skyfront.com/perimeter-drone/. Skyfront.

[38] De Moura, Leonardo and Bjørner, Nikolaj. "Z3: An efficient SMT solver." In "International conference on Tools and Algorithms for the Construction and Analysis of Systems," pages 337–340. Springer, 2008.

[39] Riccitiello, John. "John Riccitiello sets out to identify the engine of growth for Unity Technologies (interview)." *VentureBeat. Interview with Dean Takahashi. Retrieved January*, volume 18, page 3, 2015.

[40] Xu, Yongzhe, Kim, Eunju, Lee, Kyunjoo, Ki, Jaesug, and Lee, Byungsoo. "FDS simulation high rise building model for unity 3D game engine." *International Journal of Smart Home*, volume 7, no. 5, pages 263–274, 2013.

[41] Yang, Kuang and Jie, Jiang. "The designing of training simulation system based on unity 3D." In "2011 Fourth International Conference on Intelligent Computation Technology and Automation," volume 1, pages 976–978. IEEE, 2011.

[42] Yang, Chi-Wen, Lee, Tsung-Han, Huang, Chien-Lung, and Hsu, Kuei-Shu. "Unity 3D production and environmental perception vehicle simulation platform." In "2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)," pages 452–455. IEEE, 2016.

[43] Fürst, Jonathan, Fierro, Gabe, Bonnet, Philippe, and Culler, David E. "BUSICO 3D: building simulation and control in unity 3D." In "Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems," pages 326–327. ACM, 2014.

[44] Meng, Wei, Hu, Yuchao, Lin, Jiaxin, Lin, Feng, and Teo, Rodney. "ROS+ unity: An efficient high-fidelity 3D multi-UAV navigation and control simulator in GPS-denied environments." In "IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society," pages 002562–002567. IEEE, 2015.

# APPENDICES

# APPENDIX A

# CODE LISTINGS

Program A.1: Shift Factor (SF) Calculation

```
linedata = linedatas(buses);
lineStarts = linedata(:,1);
lineEnds = linedata(:,2);
lineReactances = linedata(:,4);

B = zeros(buses, buses);
C = zeros(buses, buses);
for i = 1:lines
    s = lineStarts(i);
    e = lineEnds(i);
    C(s,e) = i;
    C(e,s) = i;
end

for i = 1:buses
    for j = 1:buses
        if (i == j)
            continue;
        elseif (C(i,j) > 0)
            B(i,j) = -1/lineReactances(C(i,j));
        end
    end
end

for i = 1:buses
    sum = 0;
    for j = 1:buses
        if (C(i,j) > 0)
            sum = sum + 1/lineReactances(C(i,j));
        end
    end
    B(i,i) = sum;
end

B = B(:, 2:end);
B = B(2:end, :);
Z = inv(B);

D = zeros(lines, lines);
for i = 1:lines
    D(i,i) = 1/lineReactances(i);
end

A = zeros(lines, buses);
for i = 1:lines
    s = lineStarts(i);
```

```
47      e = lineEnds(i);
48      A(i,s) = 1;
49      A(i,e) = -1;
50  end
51
52  H = D*A;
53  H = H(:, 2:end);
54
55  S = H * Z;
```

Program A.2: Line Outage Distribution Factor (LODF) Calculation

```
 1  Ybus=zeros(num_of_bus);
 2  Impedence=zeros(num_of_bus);
 3  line_direction=zeros(num_of_bus);
 4
 5  for i=1:size(line_data,1)
 6      Impedence(line_data(i,2),line_data(i,3))=1/line_data(i,4);
 7      Impedence(line_data(i,3),line_data(i,2))=1/line_data(i,4);
 8      end
 9
10  for i=1:size(line_data,1)
11      Ybus(line_data(i,2),line_data(i,3))=-line_data(i,4);
12      Ybus(line_data(i,3),line_data(i,2))=-line_data(i,4);
13  end
14
15  for i=1:size(Ybus,1)
16      Ybus(i,i)=-sum(Ybus(i,:));
17  end
18
19  Y_eliminated=Ybus(2:end,2:end);
20  M=inv(Y_eliminated);
21  X_sen=M;
22  X_sen=[zeros(1,size(X_sen,2)); X_sen];
23  X_sen=[zeros(size(X_sen,1),1) X_sen];
24
25  LODF=zeros(num_of_line);
26  lodf_row=0;
27
28  for i=1:num_of_bus
29      for j=i+1:num_of_bus
30          if abs(Impedence(i,j))~=0
31              lodf_row=lodf_row+1;
32              lodf_column=0;
33
34              for k=1:num_of_bus
```

```
35                  for l=k+1:num_of_bus
36                      if abs(Impedence(k,l))~=0
37                          lodf_column=lodf_column+1;
38                          if i==k && j==l
39                              LODF(lodf_row,lodf_column)=0;
40                          else
41                              numerator=Impedence(i,j)*(X_sen(i,l)-
                                    X_sen...
42                                  (j,l)-X_sen(i,k)+X_sen(j,k))/
                                        Impedence(k,l);
43                              denominator=Impedence(i,j)-(X_sen(i,i)+
                                    X_sen...
44                                  (j,j)-2*X_sen(i,j));
45                              LODF(lodf_row,lodf_column)=numerator/
                                    denominator;
46                          end
47                      end
48                  end
49              end
50          end
51      end
52  end
```

# APPENDIX B

# ALGORITHMS

**Algorithm 1** Criticality-Ranking-Algorithm

Input: $\mathbb{X} = \{x_1, \ldots, x_i, \ldots, x_N\}$ is the set of data points (PIs), where $N = |\mathbb{L}|$.
Input: $D$ is the maximum distance (between a data point and the cluster center) allowed within a cluster.
Input: $K$ is the initial number of ranks (clusters).
Output: Let $\mathbb{R} = \{r_1, \ldots, r_i, \ldots, r_N\}$ is the set of criticality weights of the transmission lines.
Let $\mathbb{C}_1, \ldots, \mathbb{C}_j, \ldots, \mathbb{C}_K$ be the set of clusters to which the data points are to be assigned. Each cluster $\mathbb{C}_j$ has a cluster center (or mean) $c_j$. Let $\mathbb{C} = \{c_1, \ldots, c_j, \ldots, c_K\}$.
Let parameter *Status* denote if there is a valid clustering.
Initialize: *Status* := FALSE
**while** TRUE **do**
    Clustering-Algorithm($\mathbb{X}$, $K + 1$)
    $K := K + 1$
    Calculate the distance ($d_{i,j}$) between each data point $x_i$ and each cluster center $c_j$.
    **if** The maximum of $d_{i,j}$s $> D$ **then**
        **if** Status = TRUE **then**
            Consider the last cluster
            **for** each $\mathbb{C}_j$ **do**
                **for** each $x_i \in \mathbb{C}_j$ **do**
                    $r_i := c_j$
                **end for**
            **end for**
            **return** $\mathbb{R}$
        **else**
            Clustering-Algorithm($\mathbb{X}$, $K + 1$)
        **end if**
    **else**
        Status := TRUE
        Clustering-Algorithm($\mathbb{X}$, $K - 1$)
    **end if**
**end while**

**Algorithm 2** Clustering-Algorithm

Input: $\mathbb{X} = \{x_1, \ldots, x_i, \ldots, x_N\}$.
Input: $K$, the number of clusters.
Output: Clusters $(\mathbb{C}_1, \ldots, \mathbb{C}_j, \ldots, \mathbb{C}_K)$ and corresponding cluster centers or means $(\mathbb{C})$.
Initialize: $\mathbb{C}_j$s are initially empty.
Initialize: $c_j$s are arbitrarily chosen from the data points $x_i$s.
**while** TRUE **do**
    Calculate the distance $(d_{i,j})$ between each data point $x_i$ and each cluster center $c_j$.
    Assign $x_i$ to the cluster $(\mathbb{C}_j)$ with center $c_j$ if $d_{i,j}$ is the minimum.
    **if** No change in the members of $\mathbb{C}_j$s **then**
        **return** $\mathbb{C}$ and $\mathbb{C}$s;
    **end if**
    **for** $c_j \in \mathbb{C}$ **do**
        Update cluster center as $c_j = \sum_{x_i \in \mathbb{C}_j} x_i / |\mathbb{C}_j|$
    **end for**
**end while**

# VITA

Rahat Masum was born in Dhaka, Bangladesh, on December 10, 1993. He attended elementary schools in Mirpur and graduated from Monipur High School in 2008. He completed his two years of Higher Secondary School from SOS Herman Gmeiner College in July 2010. The following February, he entered Bangladesh University of Engineering and Technology in 2011 and received the degree of Bachelor of Science in Computer Science and Engineering in March 2016. He entered Tennessee Tech University in August 2017 and received a Master of Science degree in Computer Science with a concentration in Cybersecurity in May 2019.