Detection of False Data Injection Attacks in Multi-Microgrid

Master of Science Thesis
University of Turku
Department of Computing
Information and Communication Technology
2021
Juho Kaasalainen

Supervisors:
Ethiopia Nigussie Adj. Prof.
Antti Hakkala D.Sc.(Tech)

UNIVERSITY OF TURKU
Department of Computing

Juho Kaasalainen: Detection of False Data Injection Attacks in Multi-Microgrid

Master of Science Thesis, 50 p.
Information and Communication Technology
December 2021

In this thesis an Intrusion Detection System was developed to fight False Data Injection Attacks in Multi-Microgrids. Multi-Microgrids are a part of future power systems and they form the core part of critical infrastructure where resiliency and availability are exceedingly important. Severe consequences in the main power grid can happen if security is not taken into account. The Energy Management System has to be protected against cyber-attacks and one of the dire threats is a False Data Injection Attack. False Data Injections in Energy Management Systems are among the critical threats that need to be taken seriously as they can cause a major harm.

In this thesis, the impact of a False Data Injection Attack on Multi-Microgrids and Energy Management Systems has been explored. It has also been researched how to detect these attacks by designing and developing a Multi-Microgrid model in MATLAB/Simulink for emulating the operation of Multi-Microgrid. The MATLAB/Simulink model simulates a Multi-Microgrid environment over the course of 24 hours. To detect False Data Injection Attacks from the data created in this simulation a Kalman Filter based Intrusion Detection System was developed. The Kalman Filter based Intrusion Detection System analyzes simulation data for possible False Data Injection Attacks. Further analysis was done based on the results of the Kalman Filter based Intrusion Detection System implementation. The implementation was tested with a set of attack simulations. The results analysis revealed that developed Kalman Filter based Intrusion Detection System is suitable for detecting simple attacks but it has low accuracy for complex intrusion attacks. With taking into account only the types of attacks the implementation was initially planned to detect the detection rate averaged to 87 %. The detection accuracy could be improved in future work by considering complex attack types early on in the implementation of the detection system.

Securing power systems against malicious actors from causing harm or gaining financial benefits is a far-reaching research topic with plenty of future paths to explore. Kalman Filter based methods are one of the potential methods for detecting False Data Injection Attacks in Energy Management Systems. More research on Kalman Filter based protections is part of the ongoing race in protecting ourselves from cyber-attacks against critical infrastructure.

Keywords: Multi-microgrid, FDI, FDIA, IDS, ICT

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
APT	Advanced Persistent Threat
CC	Cloud Computing
CIA	Confidentiality, integrity and availability
CPS	Cyber-physical System
DDoS	Distributed Denial of Service
DER	Distributed Energy Resource
DG	Distributed Generation
DIA	Data Injection Attack
DoS	Denial of Service
DS	Distribution System
DSE	Distributed State Estimation
EMS	Energy Management System
FDIA	False Data Injection Attack

FDI	False Data Injection
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Prevention System
MG	Microgrid
MMG	Multi-Microgrid
MQTT	Message Queue Telemetry Transport
PV	Photovoltaic
QOS	Quality of Service
RSC	Recursive Systematic Convolution
SCADA	Supervisory Control and Data Acquisition
SG	Smart Grid
SSE	Static State Estimator
WAN	Wide Area Network
WAPMC	Wide-area protection, monitoring and control
WLS	Weighted Least Squares

1 Introduction

Microgrids (MGs) and their networked Multi-Microgrid applications (MMGs) can be thought as energy systems used for handling different kinds of electricity needs for complex and varied environments. MGs can provide a good infrastructure for environmental friendly and cost-effective electricity by best utilizing the distributed energy resources (DERs) available in the system. When these MGs are connected into a cluster of MGs as MMG the complexity of the system increases as the networks have to discuss with each other and an additional communication layer is required. Improving the reliability and resiliency of electric network are one of the key areas for utilizing the microgrids. Understanding how these systems work and securing them is important as the technology gets a bigger foothold in critical systems of our surrounding infrastructure and our everyday lives. [1] [2]

1.1 Motivation

Microgrids are part of the future daily lives of multiple people around the globe running different kinds of environments with different kinds of electricity needs. There will be 28.5 billion networked devices by 2022 and IoT means to connect all different kinds of these devices with IP-based solutions. The MGs can be considered to be one of the most significant applications of these IoT technologies. The MGs can be connected to the main power grid to send surplus power or receive power when in need and such it is not a matter of only working in a closed off system but within larger interconnected network which is part of the major infrastructure. [2] [3]

The security of these systems rises in importance as they come to be more common. Important infrastructure like the Iranian nuclear power plants (Stuxnet) or Ukraine's power systems (Disakil) have already been reported to be under attack from cyber-attacks. Researching the security of future systems is a good subject to secure the future infrastructure. This thesis focuses on creating and evaluating a tool for detecting one type of attack that can happen in MG environment: False Data Injection Attack (FDIA) that can cause havoc in critical systems. [4]

Power outages can cause catastrophical damage to critical infrastructure like military services, emergency services, hospitals or water treatment plants with major financial consequences. Doing research on this topic is vital as power outages for example in the United States have been estimated to have cost 25-70 billion dollars annually. MMGs being a Cyber Physical System (CPS) are very vulnerable to cyber-attacks because of their critical nature and distributed architecture. Any methods to improve the situational awareness of the system against cyber-attacks are important. One of the more severe cyber-attacks against MMGs are FDIs. Working on a IDS to cover against these FDIs and improving the cyber security is one of the key goals of this thesis. [5]

1.2 Objective of the Thesis

The primary objective of this thesis is to develop an Intrusion Detection System (IDS) for detection of FDIAs. The subobjective of this thesis is also to learn and gather information about how the MGs work, how the underlying ICT networks function and learn about the cyber security side of these systems. Learning about FDIAs and how to detect false data in pragmatic and real world infrastructure. The third subobjective is to create a reference point for future research into the topic for any further insight into how the MG security could be implemented.

1.3 Problem Statement

Multiple self-governing MMGs are emerging to be one of the best forms for fulfilling the future power needs as they are a resilient and reliable source of energy. Securing these future systems is vital and this thesis focuses on designing and developing a system for detecting cyber based threats against these critical pieces of infrastructure. The current state-of-the-art research is still working on developing different protections against the cyber-attacks against these networks. We go over multiple different papers that touch on the subject and focus on the specific cyber-attack of FDIA by designing and developing a system against these attacks. This FDI is one of the prominent threats and in this thesis we focus on developing a defence for this. [1]

1.4 Research Questions

Thesis will answer three research questions:

- RQ1: What kind of IDS system would be suitable?
- RQ2: Distributed vs centralized?
- RQ3: Which Implementation option?

1.4.1 RQ1: What Kind of IDS System Would Be Suitable?

There are multiple different ways to implement an IDS. What would be the best possible to use in MMG context? If distributed alongside the network, The system should be lightweight enough to handle itself in low-powered computing systems without massive stacks of computing power. The machines that run the IDS may or may not be situated in the same logical are as the rest of the power system: If the IDS is running in a CC environment, then there are more computing power, but aggregating information from all the nodes rises up as a potentially troubling question. Should the system be signature-based, specification-based or anomaly-based? [3]

1.4.2 RQ2: Distributed vs Centralized?

Which system will host the IDS? the researchers propose to split the communication on physical / application layers and handle stuff in the cloud. => More computing power in centralized cloud?

Centralized cloud offers the system more computing power. It increases the relience of being able to communicate with the cloud based servers. In a MMG where network connections can be seen as flaky it is reasonable to assume that access to cloud networking resources are not always available. On the other hand distributed systems might also feel a little more complex as handling asynchronous updates with different nodes is seen as an additional step over implementing the base synchronous model. Centralization brings forth a single point of failure, but it is arguably harder to handle a complex distributed system. [6] [7]

1.4.3 RQ3: Which Implementation Option?

What kind of implementation? The implementation can be done in a simulated environment or additionally as a real world test suite implementaion with limited hardware like Raspberry Pis and photovoltaic (PV) panels. MATLAB/Simulink, MATLAB/Python are some of the possible simulation environments. MQTT a possible protocol candidate. [2] [8]

1.5 Thesis Structure

This thesis focuses on going over revelant articles and analyzing the key components of what are multi-microgrids, their core features and how they are best secured against cyber-attacks. Of these cyber-attacks this thesis will specifically focus on how to detect FDI attacks in the system.

The Introduction chapter 1 will go over the motivation of why this is important and give a preliminary setting for the work. In the chapter 2 of the thesis we will do a survey of relevant research papers in the field giving us a suitable overview of what are multi-microgrids, what kinds of ICT networks they work on and what kinds of cyber-security questions arise when talking about MGs and how to protect our systems from them. In the chapter 3 we go over the design of the network and security solution we came up with. We will talk about what kind of solution we came up with. The fourth chapter 4 will discuss about the implementation of our network simulation, tools that were used in the making of the system and how the system is configured. The fifth chapter 5 focuses on future research and what to improve in our system.

2 Research Literature Surveys

Idea is to do in-depth study of existing literature. We start from defining multi-microgrids and their configurations. The first article introduces a hybrid communication platform and architecture for multi-microgrids. The second article of the MG focused set talks about the future of what MGs could be like.

We then follow by surveying an article about the ICT network layer of the system and how the communication in SGs is done in existing systems and what the future beholds: What standards are in the works and what research is done about the future implementations. The researchers state their proposed model of separated physical and application layers for SGs to work on.

The third section touches on the topic of cyber security in MGs, the importance of MGs in the critical infrastructures, cyber threats and the ways to counter them.

2.1 Communication Technologies in Smart Grids

Information infrastructure is vital to operate smart grids (SGs). In the paper the researchers go over the major topics of architecture, key technologies, the requirements of SG communication infrastructure, standardization, applications and future issues with the additional touch of discussing cloud computing and Internet of Things (IoT) on SG level. [6]

The smart grids communication infrastructure can be divided into three levels on the architectural point of view: short-range, medium-range and long-range networks.

The article gives a detailed overview of SG communication infrastructure and states the following contributions:

- Detailed review on technical features of communication infrastructure

- Survey of standardization works of the communication infrastructure

- Overview of Cloud Computing (CC) and IoT with potential applications in SG context

- Open issues, challenges and future research directions

The intelligent features of SG are healing, awareness, self-configuration and organized operations with received data in mind. The SG should also analyze and learn from its surroundings and act within reasonable time. Security is critical for the generation of energy and the customer side in the power system. The researchers propose a model for SGs to separate physical concepts from the applications. This method assumes that the SG typically consists of two layers: physical and application layer [6]

2.1.1 Physical Layer

Consists of power, information and communication infrastructure. The SG provides bidirectional flow when compared to more traditional power grid.

The three physical layer infrastructures:

- Power infrastructure: The generation, transmission and consumption of power

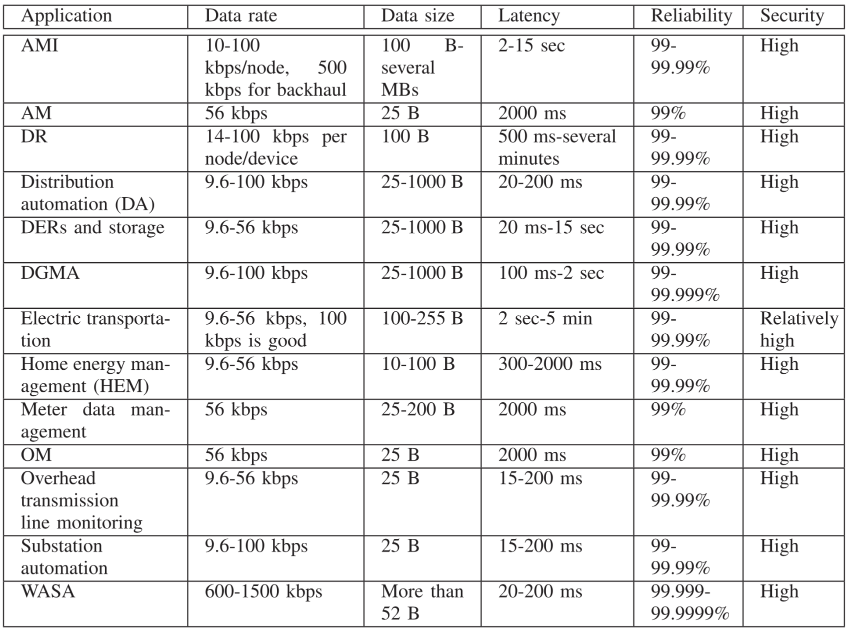
- Information infrastructure: Metering, monitoring, controlling and management

- Communication infrastructure: Information exchange between the system and applications [6]

2.1.2 Application Layer

The Application layer handles management and security.

The key points:



Application	Data rate	Data size	Latency	Reliability	Security
AMI	10-100 kbps/node, 500 kbps for backhaul	100 B-several MBs	2-15 sec	99-99.99%	High
AM	56 kbps	25 B	2000 ms	99%	High
DR	14-100 kbps per node/device	100 B	500 ms-several minutes	99-99.99%	High
Distribution automation (DA)	9.6-100 kbps	25-1000 B	20-200 ms	99-99.99%	High
DERs and storage	9.6-56 kbps	25-1000 B	20 ms-15 sec	99-99.99%	High
DGMA	9.6-100 kbps	25-1000 B	100 ms-2 sec	99-99.999%	High
Electric transportation	9.6-56 kbps, 100 kbps is good	100-255 B	2 sec-5 min	99-99.99%	Relatively high
Home energy management (HEM)	9.6-56 kbps	10-100 B	300-2000 ms	99-99.99%	High
Meter data management	56 kbps	25-200 B	2000 ms	99%	High
OM	56 kbps	25 B	2000 ms	99%	High
Overhead transmission line monitoring	9.6-56 kbps	25 B	15-200 ms	99-99.99%	High
Substation automation	9.6-100 kbps	25 B	15-200 ms	99-99.99%	High
WASA	600-1500 kbps	More than 52 B	20-200 ms	99.999-99.9999%	High

Table 2.1: SG Application Requirements. [6]

- Management infrastructure: monitoring, control and management services

- Protection infrastructure: protection for SG against physical attacks, cyber-attacks and faults. The major area of interest of this thesis would be to work on this level. [6]

The smart grid communication network should also provide certain level of performance to fulfill its requirements. Some critical communication requirements according to the researchers are: Bandwidth, coverage area, data rate, latency, reliability and security. The communication infrastructure of the SG requires minimum features from the communication network. The researchers provide the table 2.1 to illustrate these requirements.

2.2 Multi-Microgrids

MGs are an essential part of future power grids. MGs can contain lots of different equipment connected to a network and they MGs can communicate inside the

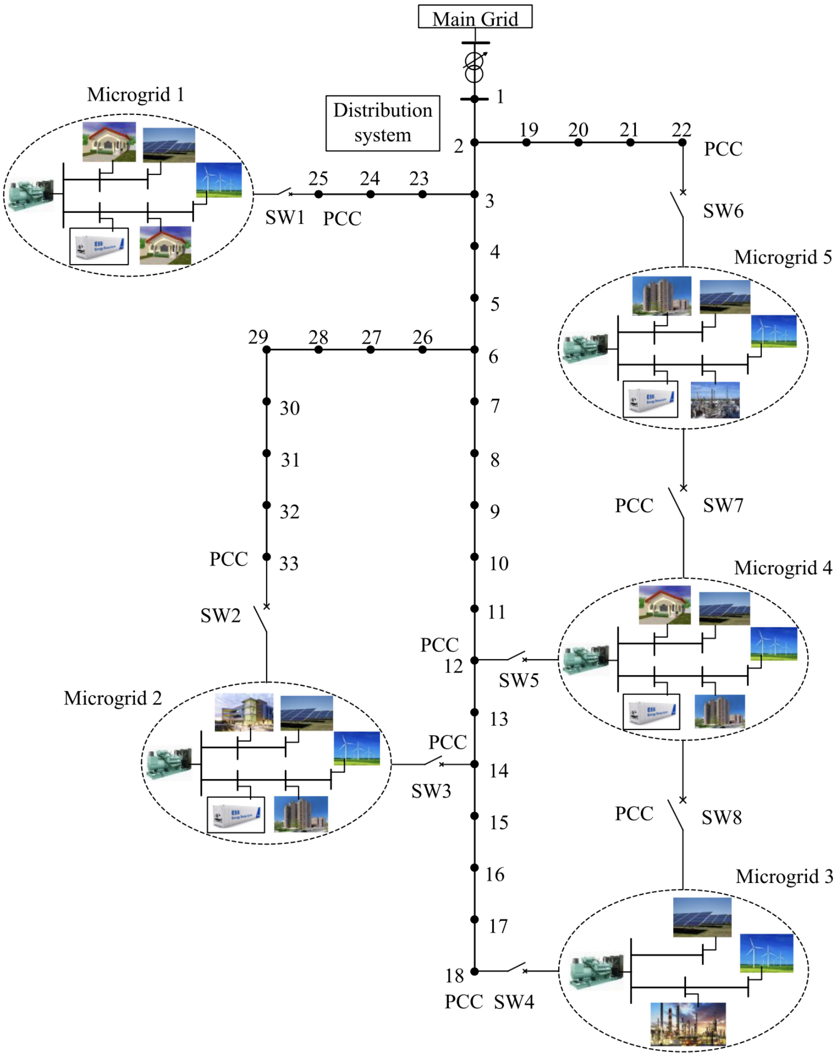


Figure 2.1: Multi-Microgrid Distribution System. [1]

grid system between the devices and with the local controller. When talking about Multi-Microgrids (MMGs) another layer of communication rises above the communication inside the specific MG: The interconnected communication and operation of different MGs working together to create a bigger MMG system. [2] These MMGs can improve the resiliency and reliability of the power system networks in a cost-effective and potentially environmental-friendly way by utilizing the DERs available. Figure 2.1 demonstrates a type of MMG distribution system setup. [1]

2.2.1 A Hybrid Communication Platform for Energy Management System

The researchers go over one possible architecture for multi-microgrid network. They specifically focus on how the Energy Management System (EMS) is best implemented. This proposition suggests using multiple different kinds of communication protocols in different settings to get a best possible communication protocol for each individual use-case. Multiple microgrids are designed to communicate with each other so it is vital to have the best possible way of communication between the grids. The figure 2.2 displays a system with EMS nodes working together. [2]

Microgrid control systems can be made to be centralized or distributed: In the centralized system the information is aggregated into cloud where different kinds of algorithms can be ran over the collected data running optimization procedures as needed. The down side is that centralized decision making takes more time, is more complex and needs reliable communication within the network. The distributed mode of operation gives weaker and less reliable solutions, but is cheaper and consumes less time. The local controllers collect what data they can from their intelligent electronic devices and meters and take into the account the state of the other microgrids. Then they run their optimization algorithms for their own grids without any input from the cloud environment. The researchers suggest an architecture that is a hybrid solution between the centralized and distributed solutions. [2]

Modbus communication is utilized to communicate withing the grid and the central controller. Message Queue Telemetry Transport (MQTT) is used to talk with central contoller and with the could server. Hypertext Transfer Protocol (HTTP) requests handle the communication with cloud channels. Virtual Wide Area Network (WAN) emulator is used to add latency to the system emulation. [2]

Possible protocols to use for the implementation:

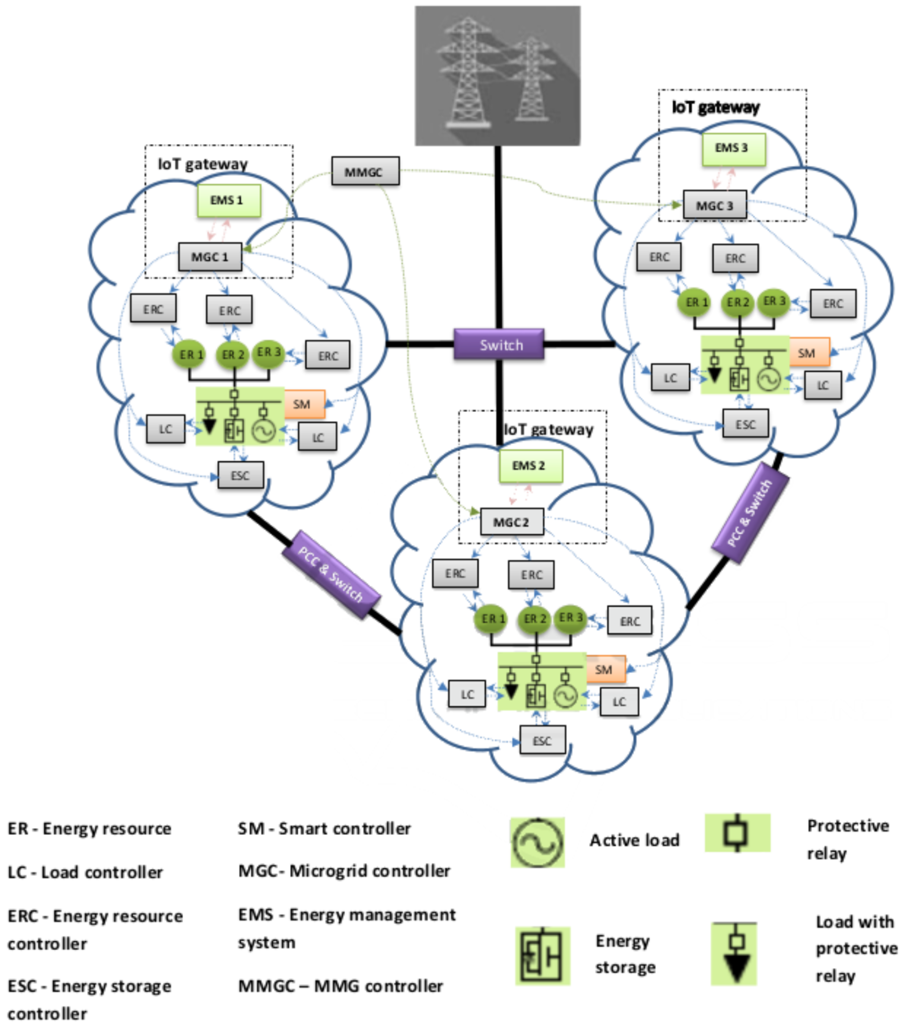


Figure 2.2: EMS and MMG Network. [9]

- Modbus: Within the grid.

- MQTT: Between central controller and a server.

- HTTP: More costly option.

What is the current state of the microgrids? What does the future of the microgrids behold? The research team states that self-governed microgrids are one of the best possibilities for improving the power system network: Both the resiliency and reliability of the system. Multi-microgrid can generate cost-effective and environmental friendly energy from Distributed Energy Resources (DERs). The microgrids are not necesserily working only with renewable energy sources like solar and wind power, but can also contain diesel engines, micro-turbines, fuel-cells and power plants. [1]

The different communication styles presented in the paper are: Master-slave communication and publish-subscribe communication. The biggest difference between these two is how the communication side is handled. [1]

The researchers divide the paper in these major topics:

- Detailed discussion of the architecture.

- State-of-the-art review of various control strategies, communication technologies and energy management techniques.

- The most important works in networked MG systems

- Benefits and challenges of networked MGs and areas of research

- Future perspectives of networked MGs [1]

Networked MGs are a connection of two or more MGs that can connect to exchange power and the Distribution Systems (DSs) as needed by the system. Operating clusterms of MGs as Networked MGs is according to the researchers the best way to utilize DERs and in critical situations a way to extend the supply's duration and black start nearby systems when needed. [1]

MGs can be either AC or DC based depending on their equipment and use cases. AC MGs are currently dominating although DC grids are gaining momentum by solar panels and popular dc loads like led lights, mobile phones and other dc systems.

2.3 Cyber Security

Smart grids can be considered one of the most important implementation of IoT systems. The IoT based smart critics are absolutely critical infrastructure which contain systems that can lead to national security deficits, disruption of public order, large scale economic damage or loss of life according to the researchers. The main security rules are confidentiality, integrity and availability. Also known as the CIA triad. The identification of vulnerabilities and different kinds of cyber security threats are important for going through different counter measures and counter cyber-attacks. Cyber-security issues are slowing down the development of traditional electricity grid into smart grids. Steady improvements are made to improve the situation. [3]

The major themes the journal goes over are as follows:

- Present the background for IoT-based SGs

- Describe SG cyber-security objectives

- Use the CIA triad and network layer knowledge to evaluate existing cyber-attacks

- Propose solutions and analytic frameworks to help plan defences and analyze the SG security

- State open research issues and trends when working with SGs [3]

The CIA triad is at the core of securing the system:

Confidentiality is the protection of the system from unauthorized entry and leaks. Only the authorized people and systems should be able to read and write the data in the system. Smart grid network can contain home bound devices with visibility into the lives of the people living inside: The privacy of the users is core part of the confidentiality of the system. The data inside the system must remain coherent and untampered.

The integrity part of the CIA triad is at the core of this thesis as we focus on false data injection attacks. The integrity of the system means that only authorized parties can modify or destroy the data flowing through.

The availability of the system is the third part of the CIA triad and it means that the system should be available at all times as it is expected to be. The information system of the SG should also be accessible and usable when needed without any unreasanoble delays. Attacks against availability can corrupt, block or delay the transmission of information. Preventing Denial of Service (DoS) attacks is a core part of the availability and should be taken into account when designing the FDI protection system. [3]

By taking the CIA triad into the account the False Data Injection (FDI) protection system should utilize these further cyber-security requirements stated by the researchers:

- Authentication
- Authenticity
- Authorization
- Accountability
- Privacy
- Dependability
- Survivability
- Safety Critical [3]

According to the paper an IDS or Intrusion Prevention System (IPS) can have three detection modes: Anomaly-based, signature-based or specification-based. Which one should be used for this thesis? [3]

2.3.1 Cyber-Security of MMGs

The researchers have gathered all kinds of material about the recent studies in the field of MGs and their security for reference for future studies in different fields. They also state some groundwork about the relevant terms in revelant literature about CPS and MGs. [4]

SGs are subjected to Advanced Persistent Threats (APTs), Distributed Denial of Service (DDoS), botnets and zero-day attacks. The researchers state Stuxnet, Duqu, Red October and Black energy as examples where industrial security is breached. The researchers state a paradox that the systems should be designed to be intuitive to understand the operation of the system while simultaniously keeping the system safe. The IDS system of this thesis should be as simple as possibly while doing what its required. [4]

Popular method for detecting bad data in power transmission systems is to use a Static State Estimator (SSE) using weighted least squared (WSL) solution but the researchers state that it is not immune to the attacks itself. This paper refers to the paper refered in this thesis in chapter 2.3.2 so more about SSE approach in that chapter. [4]

The paper states that one referred research proposes to increase the security of the system by coding the signals of the states measurement by using an error-correcting code Recursive Systematic Convolutional (RSC) code with Kalman filter estimator. [4]

Distributed State Estimation (DSE) enables MMGs to be aware of the state of the whole grid system. The researchers present a secure DSE system for MMGs for fighting FDIAs in the nodes of the system using a trust-based algorithm. This algorithm excludes bad nodes from the network by utilizing an adaptive combination policy. The paper demonstrates the researcher's DSE algorithm and some case studies as results. The negative side of their algorithm is that it trusts that only one node is compromised. The research problem of bad data turns into a question of can the nodes be trusted. [7]

2.3.2 Types of Attacks

The researchers present the state estimation approach in networked MGs to detect FDIs in the control network. Each of the microgrids handles the monitoring of data coming from the neighbouring grids using a distributed state estimator. The researchers present a distributed state estimation system, which detects FDIs but also handles load disturbances and they demonstrate their built system with a 12kV networked MG in MATLAB/Simulink. [5]

Power outages in critical infrastructure has significant financial consequences. MGs can provide resilient energy during these power outages to these critical systems. Most power outages can be caused by extreme weather events, but the worry about cyber-attack induced power outages are on the rise. MGs can utilize DERs to power local load devices and with such ease the problem. The MGs, being a critical distributed CPS, are a good candidate for cyber-attacks. [5]

With CPSs different kinds of intrusions can be classified as:

- Bias injection attack
- Zero dynamics attack
- DoS attack
- Eavesdropping attack
- Replay attack
- Stealthy attack
- Covert attack
- Dynamic FDIAs

These attacks focus on different aspects of the CIA-triad in different ways. [5]

FDIA in an MG is a severe cyber-attack where attacker can corrupt the measurement data and possibly also the control data. With power transmission systems a popular way to detect these attacks is a static state estimator (SSE) with its basing on weighted least squares (WLS). The downside is that the SSE can be tricked if the network architecture is known. The researchers state that SSE approaches are not that good for MGs and follow with their own novel approach. [5]

2.3.3 The Importance of Securing EMS

The goal of the networked MGs is to achieve overall economic dispatch or in other words meeting the required loads with optimal output. To achieve the optimal communication each of the EMS systems at each MG should function efficiently at the MG distribution level and when communicating between different MGs. The functions of EMS at the distribution level include the power quality of the entire system and additionally the overall economic operation. At the MG level the functions of EMS include voltage frequency monitoring, load and generation balancing and energy storage management at each MG. The figure 2.3 indicates a typical setup diagram for networked MGs and their EMS with Distributed Generations (DGs) and power quality monitoring. The networked MGs hold multiple EMS for each node and each EMS maintains power supply inside the MG. The extra power generated is stored in a storage system or supplied to DS or neighboring MG with the help of coordination of different EMS. For any power deficits in the MG the system is supplied with power from the distribution grid or other MGs with the help of distribution level EMS. The EMS try to maintain the continuity of the whole system in an economic manner. As the EMS are key utilities for the whole system there exists a threat of cyber-attacks against them. There are various kinds of communication technologies and different kinds of information involved. There exists a possibilty of cyber-attacks hampering the whole system. [1]

In centralized control scheme a central utility can collect information from all

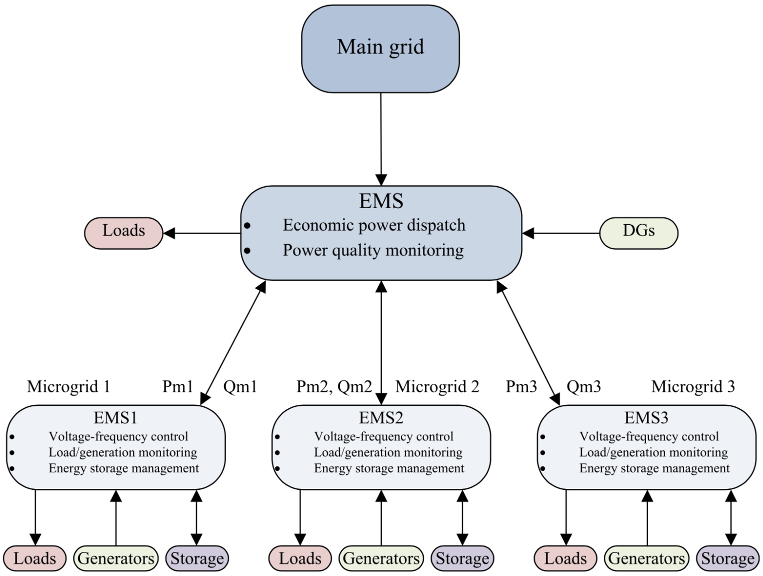

Main grid
EMS
Economic power dispatch
Power quality monitoring
Loads
DGs
Microgrid 1
Pm1
Qm1
Pm2, Qm2
Microgrid 2
Pm3
Qm3
Microgrid 3
EMS1
Voltage-frequency control
Load/generation monitoring
Energy storage management
EMS2
Voltage-frequency control
Load/generation monitoring
Energy storage management
EMS3
Voltage-frequency control
Load/generation monitoring
Energy storage management
Loads
Generators
Storage
Loads
Generators
Storage
Loads
Generators
Storage

Figure 2.3: Networked MGs and their EMS with Distributed Generations (DGs) and power quality monitoring. [1]

nodes and process the information for abnormalities. The central node and also broadcast control commands to all other nodes enabling global situational awareness against cyber-attacks. In distributed control schemes the local controllers are vulnerable to local attacks since they access local data and partial neighbour data. Malicious nodes could corrupt the data exchanged and according to the researchers is considered more vulnerable to cyber-attacks. [1]

The economic power dispatch is done by EMS regularly. During basic conditions the MGs can be grid-connected, islanded or a mix of these two conditions depending on the needs of the system. If there is an emergency condition where the exists no power supply from the main grid the MMG is optimized and only critical operations exist in each MG. The surplus energy is stored in the DS. Any extra energy can be sent to the main grid depending on the needs of the system. If the power generated is not enough for the whole system then only critical loads are supplied and networked MGs can provide some extra hours of power supply for critical loads until the main grid is restored. Because of this networked nature the privacy of MGs is mentioned to be a key challenge for MGs. Based on the ownership of the system, an MG can be one of three types: Utility MG, Community MG or Private MG. Keeping Community and Private MGs private is a concern as all the grids do not have similar interests. These types of grids might want to share as little information as possible to maintain privacy. [1]

EMS is a subsystem related to SCADA which handles real-time monitoring and controlling of the electricity distribution network. Tampering of intercepting the data damages the grid and to combat against that different technologies can be used: VPNs, IPsec, firewalls, IDS, user and device authentication. As the EMS system can be emulated by Advanced Metering Infrastructure (AMI) the Quality of Service (QOS) requirements like performance, energy efficiency and security can be well handled with modern set of services. The systems can respond to power failures, load shedding, real-time pricing and software updates. [3]

As high computation, communication and power based controllers are currently vulnerable to cyber-attacks, the sychronism of inverters can be disrupted by corrupt data leading to unstability and FDIs in power system networks have been already been reported, severe cyber-attacks may be seen in the future. Arrangements must be done to handle these issues in an economic manner. [10]

2.4 FDIA and Detection Schemes

According to the researchers the data injection attack (DIA) can be defined as an attack that means to manipulate data. Some examples in the article of data are feedback signals, sensor readings and energy consume signals. The targets of these attacks are usually state estimators, smart meters and wide-area protection, monitoring and control (WAPMC). The motives behind these attacks can be seen to be either financial benefit or system damage. The false data injection attack according to the paper happens if bad data is used to inject into a smart meter or neighborhood area network measurement attacking the SG infrastructure. The attack tries to damage the integrity of the measurements and monitoring sub-systems with the goal of manipulating phasor and meter measurements. The Supervisory Control and Data Acquisition (SCADA) system's state estimation is impacted and the attackers can potentially circumvent any data integrity checks applied during the state estimation process. [3]

2.4.1 Detection of False Data Injection Cyber-Attacks

As cyber-physical systems (CPS) are more and more utilized in different kinds of practical applications such as renewable energy plants, energy distribution and transportation, it is important to detect any possible cyber attacks in micro grids. [8]

The researchers propose a Artificial Intelligence (AI) method for detecting cyber-attacks in the MG using an FDIA as an example. This proposal works using time series analysis and a special kind of neural network for estimating voltages and currents in the system. The system is first run normally to gather training data for the AI model without any FDIAs. Then the system is being actively exploited while running and based on the error of estimation a cyber-attack is catched. [8]

One way to conduct an FDIA is to record the sensors of MG system for a specific time and start to repeat these readings to the system deceive the system. The researchers also state some previous works in identifying FDIAs in SGs:

- FDIAs on voltage measurements can be detected by using a cooperative vulnerability factor and monitor the output of secondary sub-layer to see any changes and see units under attack.

- Discordant element and concensus theory. Identifying any nodes where FDIA is under way by current measurements.

- Kalman filter bases method has been developed to identify FDIAs using mathematical model of the system.

- FDIA detection as matrix separation problem, nuclear norm minimization and low-ranking matrix factorizations.

- Kullback-Leibler distance based method that calculates healthy and attacked system's data differences. Can detect most of any possible attacks but has difficulties with some state variables.

- Another Kalman filter based approach by measuring any deviations between estimates and real values. They implemented chi-square detector and did some cosine similarity matching to identify attacks. [8]

Simulations are run in a test MATLAB/Simulink environment and the results are verified with OPAL-RT digital simulator.

3 Intrusion Detection Methods for FDIA

On this chapter we go through different design options and evaluate the taken steps for the system. What decisions would help us build the best possible system for detecting FDIAs in MMGs. We go through the overall high level architecture of the framework, the MG simulation model and the detection algorithm chosen for the FDIA. We go over how these were implemented in the chapter 4. The high level abstraction of how we could answer our research questions is discussed on this chapter.

3.1 System Overview

The architecture of the whole system should be able to properly deliver us a basic framework where we can simulate the MG and apply the IDS detection algorithm on the available data. The RQ2 of distributed vs centralized can be evaluated and the centralized option be chosen as the distributed system would be more complex while simulatneously offering no real benefits in the main scope of implementing an IDS system. To keep the scope of work in check a centralized system should be used. The overall system could function in the following manner and act as a base for any further enhancements:

- Create the MATLAB/Simulink simulation model and build the binary with

Simulink Coder.

- Run the MATLAB/Simulink's simulation binary with MG model to generate the simulation data.

- Output this data in a proper exchange format. The data is written into a .mat file.

- Handle the simulation data and plot intermediate graphs for easy previewing.

- Check for FDIA. Different possibilities for checking the FDIA was valuated and more discussed in the chapter: 3.3.

- Once the data has been processed and the IDS system picks up a candidate for attack. Give a notice for the end user about further actions.

3.2 The Role of EMS

The EMS is integral part of MG ecosystem. As discussed on the chapter 2.3.3 the core functionality EMs is to manage the energy loads on MG level and handle communication about the energy needs between different MG nodes forming a MMG with communication between the network of nodes referred in figure 2.3. As the EMS is at the core of how MMGs operate and it is vital for the fluid working of the system the IDS is vital: Securing the operation and the data integrity of the energy management is a key research topic and including a robust IDS framework with an EMS is important for any real world applications. EMS is a key part of MMGs and it has to operate correctly and securely. [1]

The figure 2.1 visualizes the interconnected whole formed by different nodes forming a collective of multiple EMS that work toward a collective optimization of the system with different kinds of DERs, static and active loads like EVs. The job of the EMS is to work with these loads and DERs and form a cohesive whole unit. The nodes can work as either islanded or grid-connected nodes in the system. Islanded system has to be responsible for a lot more of its own funtions but it can manage to work as its own entity in a case of losing connection to neighbors. Grid-connected mode brings MG a lot more stability as it can request more power from the main network as needed or offer back any extra power generated upstream. The networked MG is different from traditional power distribution network as the flow of power can be bi-directional and any changes in the network topology are frequent in networked MGs. [1]

Some of key challenges held in MMGs are the stability of the system, protection coordination, privacy of MGs and threats of cyber-attacks. DERs with uncertainties can lead to major stability issues in the networked MGs system. Photovoltaic and wind generated power sources have major stability issues and can cause frequent imbalances with loads and the energy generation. A strong EMS system with stability schemes is important. The protection coordination with changes in the network topology gives out challenges as different parts of the system might have different fault currents and safety limits. The protective devices have different optimal settings for islanded and grid-connected mode. The privacy of MGs is another challenge where security and collaboration are working against each other as not all node operators might want to share their data. The fourth challenge is the threat of the cyber-attack. As MMGs are highly computational and complex systems with many different kinds of communication technologies involved there exists a threat of cyber-attacks against the system. Involunarily corrupt data can also cause havoc in the system leading to instability. The centralized control scheme with central control entity gives the system a global situational awareness into any possible cyber-attacks. In distributed control scheme the access to local and neighbour data makes the system as a whole vulnerable to malicious attacks. A malicious attacker could corrupt a node and cause a distributed cyber-attack. All in all the MMGs contain lots of high computation controllers with many different communication systems and that makes them exposed to cyber-attacks. Developing a proper IDS system can be one of the tools to combat this threat. [11] [1]

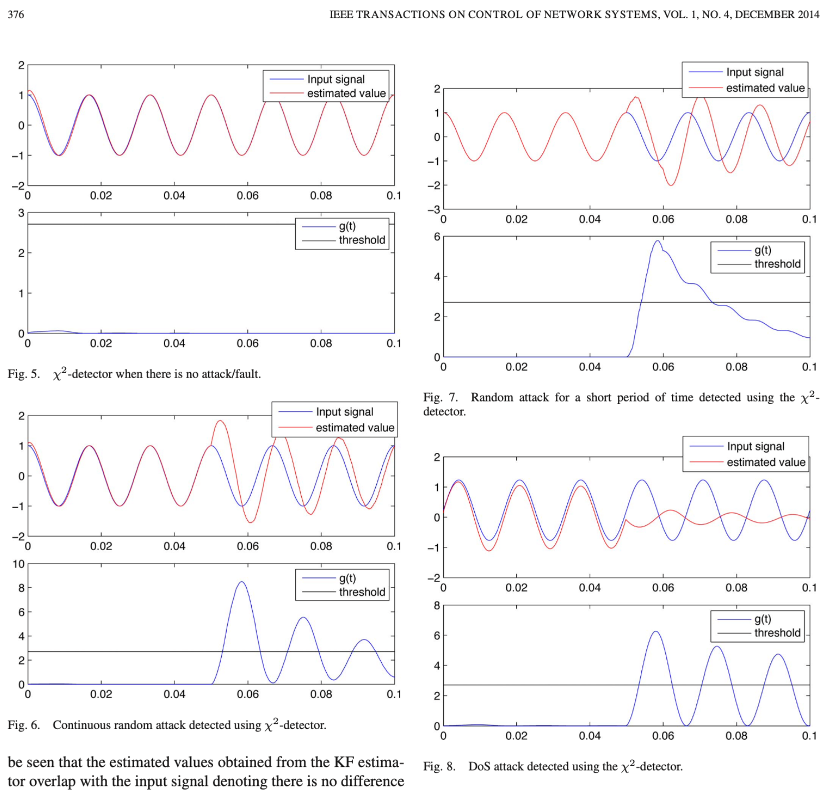
3.3 Kalman Filter based FDIA Intrusion Detection Method

On Detection of False Data Injection Cyber-Attacks in DC Microgrids based on Recurrent Neural Networks [8] the researchers used an AI to teach a model to detect anomalies in voltages and currencts in the system. One FDIA attack is to record the sensors of MG system for a specific time and start to repeat these readings to the system deceive the system. For our IDS this anomaly based detection algorithm was deemed to be out of scope of the thesis and a simpler algorithm was decided to be used.

For the RQ1 of what kind of IDS system would be suitable for this use case of IDS for MMGs the different algorithms were evaluated and Kalman Filter was deemed a proper fit for the problem as it is previously researched and deemed efficient for this use case. On the Detection of False Data Injection Cyber-Attacks in DC Microgrids based on Recurrent Neural Networks paper [8] a future reference was found with a suitable implementation of Kalman filter for finding out unpredictable data in the system. The paper went over thoroughly the use case and implementation of the algorithm and it seemed just perfect base implementation for our system. [12]

On figure 3.1 the network is operating normally. However on figure 3.2 an FDIA is introduced into the system and a treshold is breached causing the FDIA to be detected by the system.

To learn more how to implement the algorithm in question a Creative Commons Attribution 4.0 International and MIT licensed ebook Kalman and Bayesian Filters in Python [13] provided lots of help. The loop of predicting and updating was clearly taught and easy to follow. Basing the implementation on this detection algorithm



376 IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, VOL. 1, NO. 4, DECEMBER 2014

Fig. 5. χ^2-detector when there is no attack/fault.

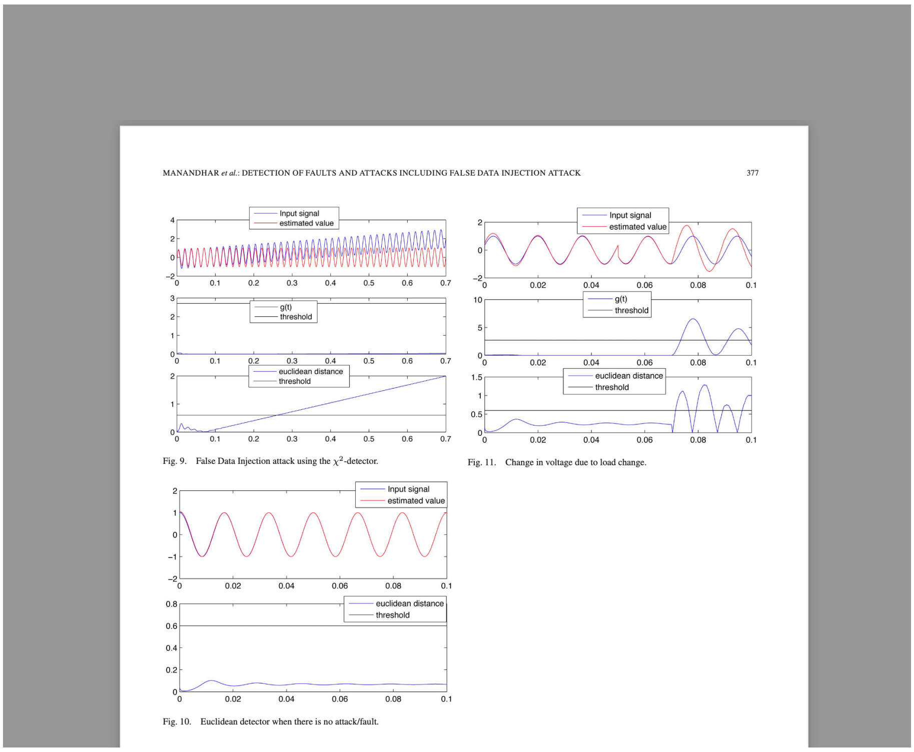
Fig. 6. Continuous random attack detected using χ^2-detector.

be seen that the estimated values obtained from the KF estimator overlap with the input signal denoting there is no difference

Fig. 7. Random attack for a short period of time detected using the χ^2-detector.

Fig. 8. DoS attack detected using the χ^2-detector.

Figure 3.1: Estimation Without an Attack. [12]



MANANDHAR *et al.*: DETECTION OF FAULTS AND ATTACKS INCLUDING FALSE DATA INJECTION ATTACK 377

Fig. 9. False Data Injection attack using the χ^2-detector.

Fig. 11. Change in voltage due to load change.

Fig. 10. Euclidean detector when there is no attack/fault.

Figure 3.2: Estimation With an Ongoing FDIA. [12]

and implmenetation guide was a clear choice.

How does the algorithm work exactly? To give a relatively easily digestable form of the algorithm the set of steps can be described as pseudocode. Leaving out all the practical implementation phases of for instance a Python script and by focusing on the important business code steps we can form an easily transmittable set of principles to state the intention of the algorithm. The pseudocode description of the Kalman Filter algorithm can be summarized with the following steps by R. Labbe: [13]

```
Initialization
1. Initialize the state of the filter
2. Initialize our belief in the state


Prediction
```

```
1. Use system behavior to predict state at the next time step
2. Adjust belief to account for the uncertainty in prediction


Adaptation
1. Get a measurement and associated belief about its accuracy
2. Compute residual between estimated state and measurement
3. Compute scaling factor based on whether the measurement
or prediction is more accurate
4. set state between the prediction and measurement based
on scaling factor
5. update belief in the state based on how certain we are
in the measurement
```

The initialization phase is used to setup the baseline of the filter and to state the proper context for the filter to work in. The predict phase uses the state propagation equations of the Kalman filter to predict the next state. After the predict phase the algorithm moves to the update phase where a new measurement is added to the Kalman filter and the state is updated. The residual between what was predicted in the prediction phase and what is actually measured is taken into the account and the final estimate is made. To visualize what is happening in a form of a diagram the figure 3.3 can be utilized. The prior is predicted from the posterior and after getting a new actual measurement we can form a new estimate. It can be crudely summarized that the Kalman Filter smooths out the changes in the system based on previous assumptions.

The core loop of how the Kalman Filter Algorithm works can be seen in the block diagram figure 3.4 made on diagrams.net with the first step would be to setup the initialization parameters. This gives the Kalman Filter proper context on what level to work with. Next we enter the core loop where we iterate over each datapoint, do

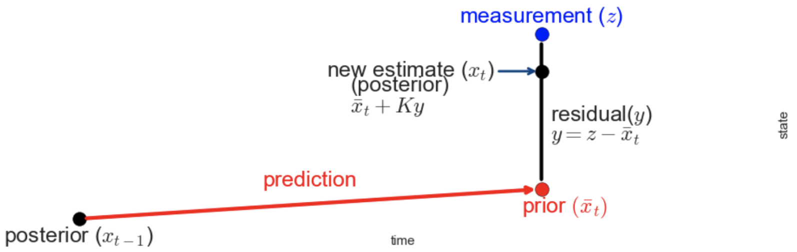


Figure 3.3: Diagram of Kalman Filter Algorithm. [13]

a prediction step on the data, perform an update step and see that do we continue with more data or was that it. [14]

The variable names for the predict and update functions can be considered awful from a software engineering point of view, but they are established in the literature of Kalman filters as idiomatic way to display this information. R. Labbe gives some background for the letters for the predict and update functions:

- x is the state estimate
- R is measurement noise
- Q is for the process noise
- P is for the state variance
- z is the measurement
- u is the control vector [13] [15]

The initialization assumptions also mentioned on figure 3.4 we are making are based on these variables.

- The x is the state estimate of the step we are currently evaluating.

- The R is the measurement noise and is being used in the update function to take into consideration for any spikes in the data. It can also be called the measurement variance as real world measurements tend to be very noisy.

- Q is the process noise in the predict function and it is used to model the unknown changes in the system. The process noise represents the idea that the system state keeps changing but we do not know the specifics of how and when the

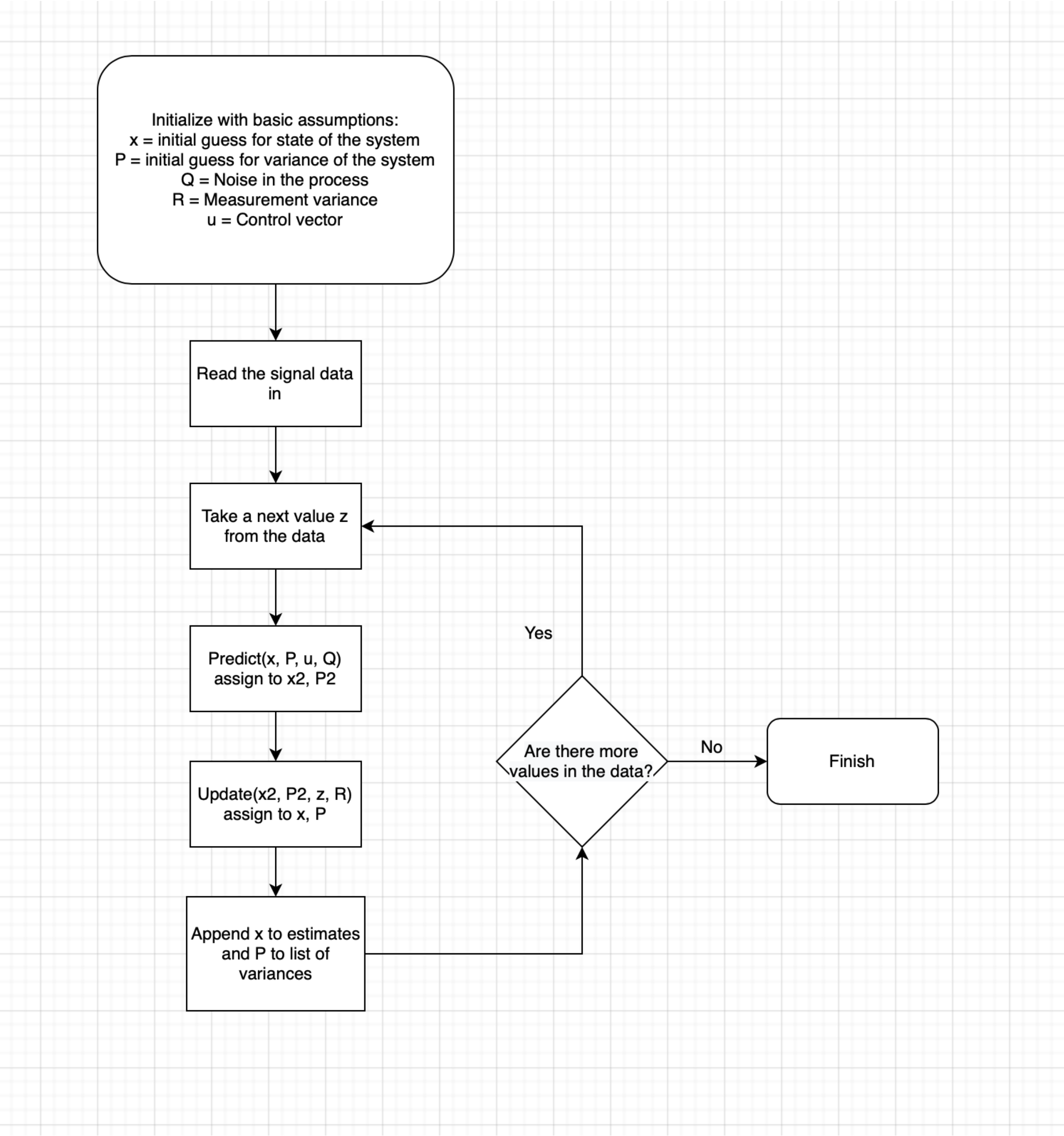

Initialize with basic assumptions:
x = initial guess for state of the system
P = initial guess for variance of the system
Q = Noise in the process
R = Measurement variance
u = Control vector
Read the signal data in
Take a next value z from the data
Predict(x, P, u, Q) assign to x2, P2
Update(x2, P2, z, R) assign to x, P
Append x to estimates and P to list of variances
Are there more values in the data?
Yes
No
Finish

Figure 3.4: Process of Kalman Filter based Prediction and Adaptation. [13]

changes happen.

- The P is the state variance of the system and it is used to take account the variation in the data.

- The variable z is for the measurement and it is used in the update function for the current update step.

- u can be used for the control vector in the predict function, but as the system in question is pretty simple it can be omitted. [16] [13] [15]

4 Implementation

The implementation phase takes the outcomes of the design chapter and implements them in a pragmatic matter. On the first section we go over how the MATLAB/Simulink simulation is implemented and how we get the results for further analysis. Following the powergrid simulation we go over the implementation phase of the Kalman Filter and how it works. In the third chapter we evaluate how the Kalman Filter was used to point out any anomalies in the data and how the IDS system scans for exceptional breach of predicted variances in the data. In the result analysis phase we evaluate our results and see if the goals were reached.

4.1 Simulink Model of a Microgrid

The simulator environments research for implementation were NS3 and MATLAB/Simulink. The MATLAB/Simulink had a good setup of preinstalled examples for Micro-grid simulations with premade assets to act as a foundation for the simulation. One key example for picking and learning MATLAB/Simulink was the Simplified Model of a Small Scale Micro-Grid [17] example project as it was perfect illustration of a simple microgrid network. Using this information as a base for learning how the MATLAB/Simulink system works and building on the shoulders of giants its a lot easier to come up with a customized implementation suitable for the needs of this IDS system. Building a Microgrid with needed systems should be perfect with this information.

The RQ3 of what kind of implementation could be suitable is summarized with going forward with the MATLAB/Simulink implmenetation for modeling the MG power system and an external python program to process the data. The results of the simulation can be refined for analysis with the chosen detection algorithm. The user can be notified about how the detection algorithm has fared and give them a heads up of an anomaly.

The MATLAB based Simulink handles the microgrid's power network simulation. In figure 4.1 we have a graphical overview of parts of the system and how they function. The scenario data uses the load and solar data of the [17] as a baseline as it is more interesting than a static sine wave and generating random data is not engaging part of this thesis. This control data is fed into the power system via AC Controlled Current Source block where it is controlling the currents provided by the static AC Voltage Source block with frequency of 60 Hz and peak amplitude of 100 Volts.

The simulation uses a fixed-step discrete solver with a fixed-step size of 0.1 that acts as a fundamental sample time. With a stop time of 24*60*60 = 86400 units of simulation time we get a nice 24 hour simulation with one hour time steps. On the figure 4.2 we can view the loads of the system over 24 hour time period inside a Scope Block.

The C Code generation settings are set to build to a MATLAB's Generic Real-Time Target or grt.tlc file format. The building is done with Clang and gmake to target a 64-bit Mac system. Once the simulation code has been generated, the model built and the program finally compiled into a binary we can run the simulation on a source machine and generate a dataset to feed into the IDS system. The intermediate data is generated into a .mat file format and can be easily viewed inside Matlab with tools for further processing or in our use case loaded into an external program. The IDS system processes this .mat file with SciPy's [18] Input and Output tools. The

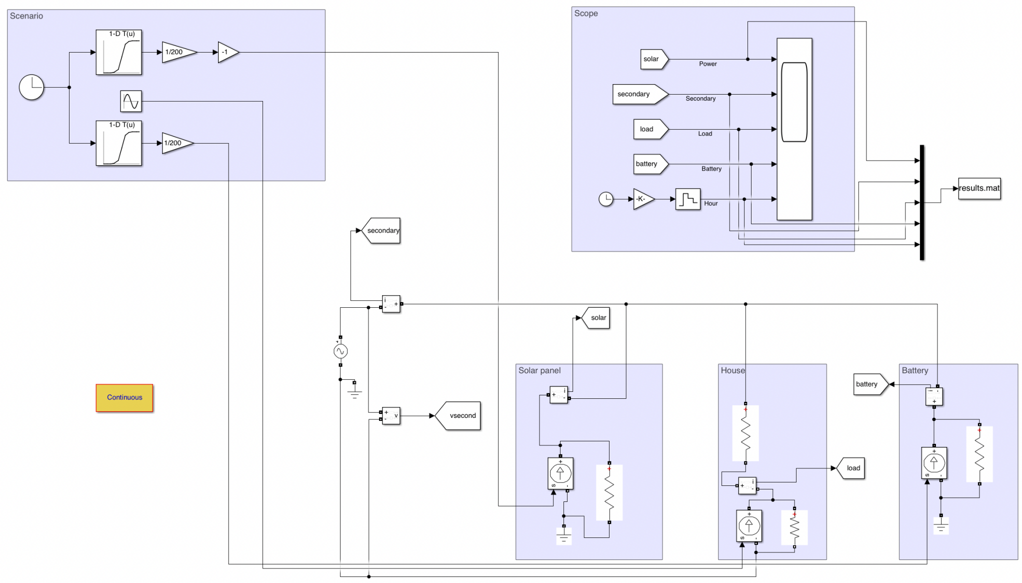


Figure 4.1: Simulink Microgrid Model.

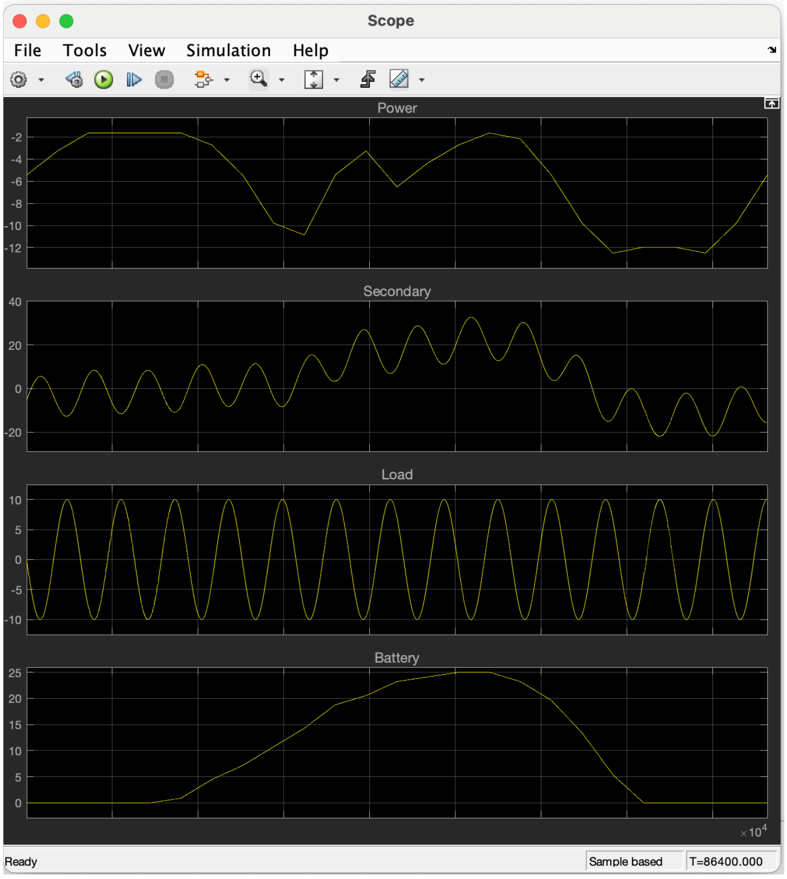


Figure 4.2: 24h Simulink Simulation Results Scope with simulated values for Solar Power, The Secondary Power or The System Power, Load of a House and Battery Load.

loadmat handles the loading of MATLAB files and enables further processing while simultaneously enabling us to graph the intermediate data points for viewing. The loading of the data is implemented as follows:

```
import scipy.io

data = scipy.io.loadmat('results.mat')['values']
```

4.2 Implementation of FDIA IDS

The Kalman Filter uses a series of data points including any possible noise and turns it into an estimate of what is the general consensus or the main point of the data. [13] The IDS system creates a prediction of the variance of the data and checks for any deviations from the expected values. One further point of improvement could be to only flag mutliple transgressions, but currently the system flags the data as faulty from even the slightes deviations from expected variance. The IDS implementation uses MIT licensed FilterPy python library for the predicting and updating the steps based on the Kalman Filter's algorithm.

The initialization variables mentioned on chapter 3.3 are assigned and guessed as follows:

- x the state estimate: This state estimate was initialized to be 25 as the load values in the figure 4.1 were in the range of -10 and 10 and the reference implementation of R. Labbe was in the same ballpark. The guess for the initial values should be close enough for our use cases. [13]

- R the measurement noise: The guess for measurement noise was set to be quite high at 10000 as the data was pretty specific and can be considered to be very noisy. The simulation data had lots of datapoints.

- Q the process noise: The process noise is how much error there is in the process model and the data we are working on can be very noisy but it should be very expectable in how it behaves. The loads vary a lot during the day but in the time ranges we are operating there shouldn't be any big spikes unless under an FDIA. The process noise was guessed to be 1. [13]

- P the state variance: The variance is chosen to be 1000 as the initial variance should be wide enough to cover the data cases but not be too far out.

- z the measurements are collected from the datapoints iteratively.

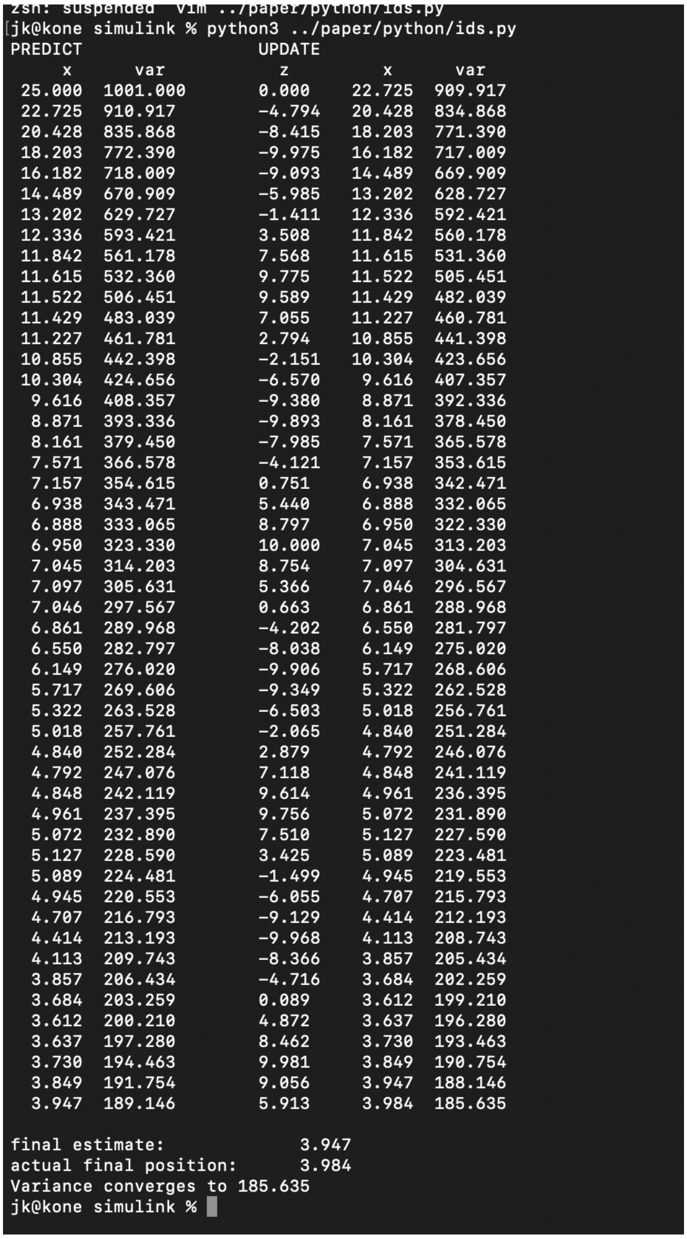
- u the control vector was deemed to be outside the scope of this implementation in the 3.3 chapter as the implementation is relatively simple in nature.

In the figure 4.3 we print out the calculation steps that go over a subset of the simulation data and create an estimation based on the data available. The left side column holds the prediction steps and the right side column holds the update steps. Based on this data we calculate a final estimate and variance range. The core loop of the calculation is as follows:

```
voltage_std = 100
process_var = 1
x = 25.
P = 1000.
ps = []
estimates = []

print('PREDICT\t\t\tUPDATE')
print('     x      var\t\t  z\t    x      var')

for z in data:
    x2, P2 = kf.predict(x=x, P=P, u=0., Q=process_var)
    x, P = kf.update(x=x2, P=P2, z=z, R=voltage_std**2)
```



```
zsh: suspended  vim ../paper/python/ids.py
[jk@kone simulink % python3 ../paper/python/ids.py
PREDICT                 UPDATE
     x      var           z         x      var
 25.000  1001.000       0.000    22.725  909.917
 22.725  910.917        -4.794   20.428  834.868
 20.428  835.868        -8.415   18.203  771.390
 18.203  772.390        -9.975   16.182  717.009
 16.182  718.009        -9.093   14.489  669.909
 14.489  670.909        -5.985   13.202  628.727
 13.202  629.727        -1.411   12.336  592.421
 12.336  593.421        3.508    11.842  560.178
 11.842  561.178        7.568    11.615  531.360
 11.615  532.360        9.775    11.522  505.451
 11.522  506.451        9.589    11.429  482.039
 11.429  483.039        7.055    11.227  460.781
 11.227  461.781        2.794    10.855  441.398
 10.855  442.398        -2.151   10.304  423.656
 10.304  424.656        -6.570    9.616  407.357
  9.616  408.357        -9.380    8.871  392.336
  8.871  393.336        -9.893    8.161  378.450
  8.161  379.450        -7.985    7.571  365.578
  7.571  366.578        -4.121    7.157  353.615
  7.157  354.615        0.751     6.938  342.471
  6.938  343.471        5.440     6.888  332.065
  6.888  333.065        8.797     6.950  322.330
  6.950  323.330        10.000    7.045  313.203
  7.045  314.203        8.754     7.097  304.631
  7.097  305.631        5.366     7.046  296.567
  7.046  297.567        0.663     6.861  288.968
  6.861  289.968        -4.202    6.550  281.797
  6.550  282.797        -8.038    6.149  275.020
  6.149  276.020        -9.906    5.717  268.606
  5.717  269.606        -9.349    5.322  262.528
  5.322  263.528        -6.503    5.018  256.761
  5.018  257.761        -2.065    4.840  251.284
  4.840  252.284        2.879     4.792  246.076
  4.792  247.076        7.118     4.848  241.119
  4.848  242.119        9.614     4.961  236.395
  4.961  237.395        9.756     5.072  231.890
  5.072  232.890        7.510     5.127  227.590
  5.127  228.590        3.425     5.089  223.481
  5.089  224.481        -1.499    4.945  219.553
  4.945  220.553        -6.055    4.707  215.793
  4.707  216.793        -9.129    4.414  212.193
  4.414  213.193        -9.968    4.113  208.743
  4.113  209.743        -8.366    3.857  205.434
  3.857  206.434        -4.716    3.684  202.259
  3.684  203.259        0.089     3.612  199.210
  3.612  200.210        4.872     3.637  196.280
  3.637  197.280        8.462     3.730  193.463
  3.730  194.463        9.981     3.849  190.754
  3.849  191.754        9.056     3.947  188.146
  3.947  189.146        5.913     3.984  185.635

final estimate:             3.947
actual final position:      3.984
Variance converges to 185.635
jk@kone simulink % █
```

Figure 4.3: Estimation and Variance.

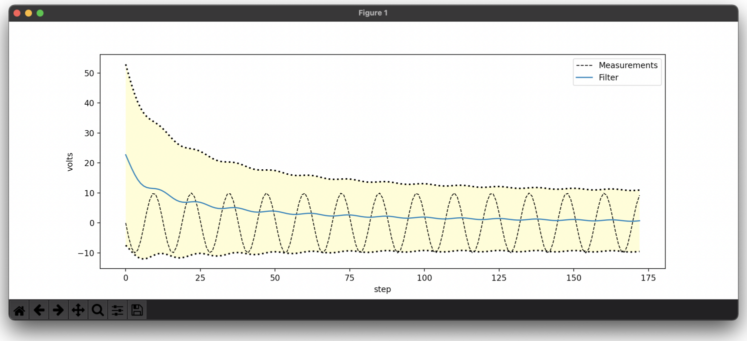


Figure 4.4: Normal Data.

```
    print_gh(x2, P2, x, P, z)
    estimates.append(x)
    ps.append(P)


print()
print('final estimate:        {:10.3f}'.format(x2))
print('actual final position: {:10.3f}'.format(x))
```

Once we include a larger subset of datapoints and plot this out open we get a nice graph of how the filter is at first undefined in the context of data but is slowly being adjusted to the data context. The figure 4.4 displays how an imput signal without any FDIAs is adjusting the intial values of the Kalman filter and the variance with the yellow background. The blue filter line is smoothing out the input signal curve and gives us an approximation of the averaged out input of the signal.

The implementation simulation has these attack types to test out the implementation:

- No attack. How the system handles the normal situation without any attack. Looking for false positives.

- Continuous FDIA. The data signal is artifically inflated with a constant trend upwards. The researchers refer to this as just an FDIA. The figure 3.2 with "False

Data Injection attack using the X2-detector" can be used as a visual reference. [12]

- Continuous FDIA with double the values. Similar to the previous attack but with more bolsterous attack.

- Negative FDIA. Attack with negative values. Trying out how the system handles unexpected results.

- Random Attack. Inflate the data with random values from a range of -10 to 10. Checking out for false positives and negatives.

- Big Random Attack. Inflate the data with random values from a range of -100 to 100. Checking out for false positives and negatives.

Once we want to commence a simulated FDIA with tampering of the data we can run the program with –fdia handle to generate an attack where the data is manipulated for personal gain or just causing havoc in the system. The figure 4.5 shows us a case where signal is artificially inflated. The IDS system detects this by comparing the calculated variance and the prediction of what the data should be with the actual values that are flowing in the system. Once this comparison detects an anomaly by detecting values outside the range of possible variance the system notifies the user of the program about a possible candidate for an ongoing FDIA.

In the FDIA figure 4.5 we can observe that the inflated measurements rise while the Kalman Filter has difficulties to keep up with the abnormal measurements. The yellow variance area and the blue filter estimation try to keep up with the rising signal, but the change is too great. The yellow variance area also diminishes towards the estimate of the filter as the variance is not being updated to catch up. We can simulate the continuous FDIA with the handle –fdia in the program.

With the continuous attack with double the attack values we get a graph like in figure 4.6 with even larger offset. This can be invoked with the handle –fdiax2

Negative FDIA range going outside the scope. The IDS didn't catch this one as the values were only looking for positive abnormalities. The negative attack can be

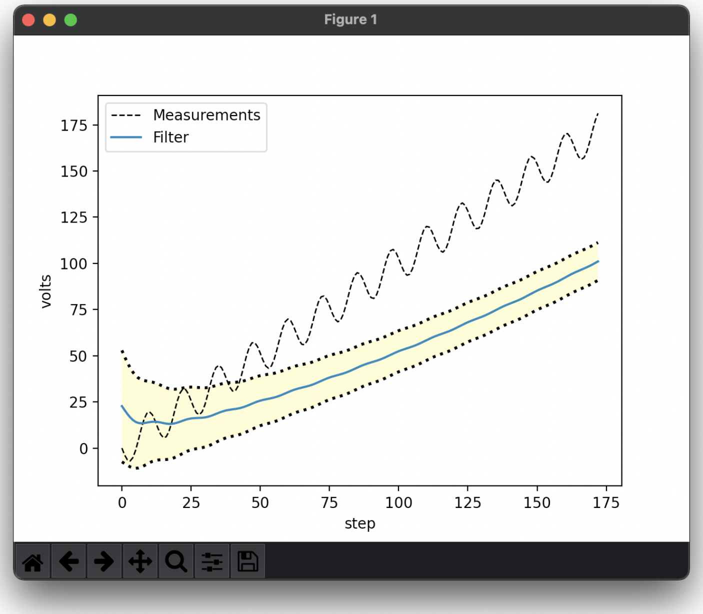


Figure 4.5: Data with a Continuous FDIA outside the scope of the variance.

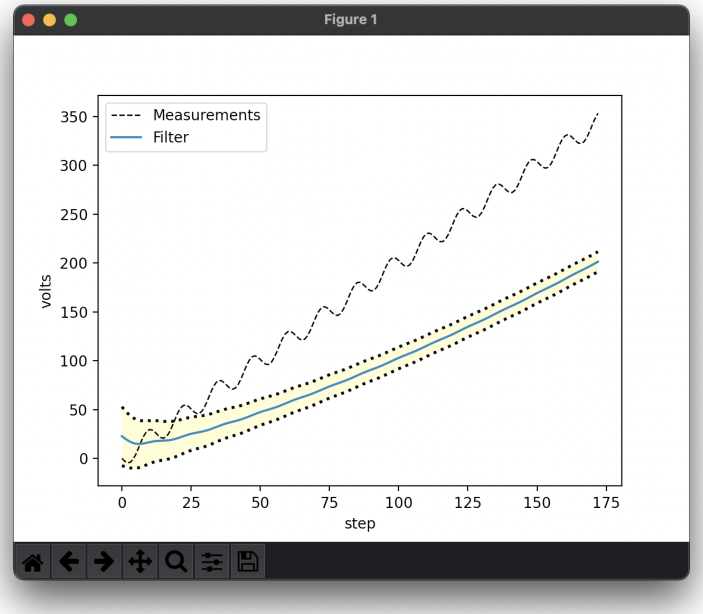


Figure 4.6: Data with a Continuous FDIA with double the values going outside the variance range.

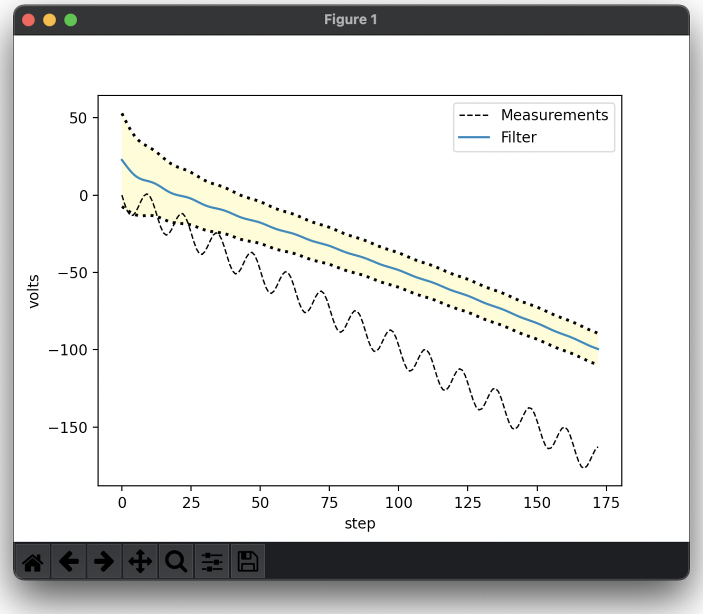


Figure 4.7: Data with a negative FDIA outside the scope of the variance.

simulated with –fdianegative.

Random FDIA had us an interesting graph at figure 4.8. The measurements momentarily exit the variance range. As the difference to the range is so small no fdia is detected. With higher random values the attack was more clear to the system and it resulted in a detection. This can be simulated with –fdiarandom.

With bigger random values affecting the data the attack is easier to detect. We can simulate the big random FDIa with –fdiarandomx10.

4.3 Results Analysis

The goal of the implementation was to develop an IDS to detect FDIAs in MMG EMS environments. The IDS was developed as planned in the 3 chapter and some data about the detection accuracy of the implementation could be collected. The implmenetation should also work in environments with limited resources as some

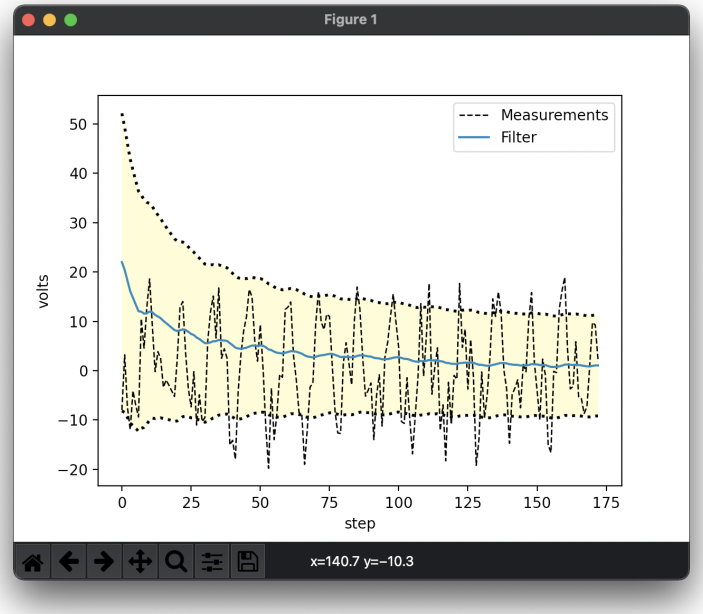


Figure 4.8: Data with a random FDIA momentarily outside the variance range.

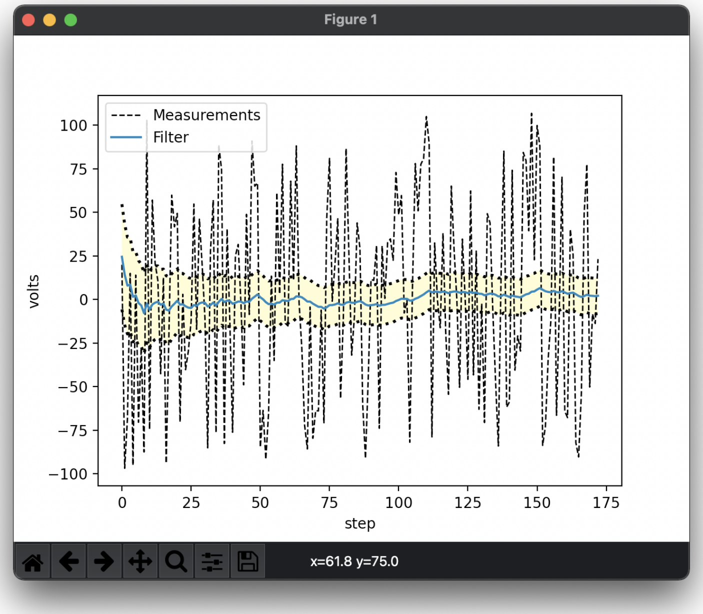


Figure 4.9: Data with a big random FDIA momentarily outside the variance range.

MMGs might have resource constraints. Especially when the MMG operates in island mode with access to main grid cut off. The developed IDS system has reached the goal of detecting an FDIA in MMG power grid although the implementation still is not perfect and it has its blind spots. The research questions laid out in the chapter 1.4 could be answered with the experiences learned in the implementation phase and by the data available.

To get some datapoints about the efficiency of the IDS the system has five different attack type simulations as mentioned in the 4.2 chapter: Continuous FDIA, Continuous FDIA with double the values, Negative FDIA, Random FDIA and Big Random FDIA. The simulations proved that the IDS system developed is still not suitable for detecting attacks that are not really obvious. In the table 4.1 we can see that the really explicit Continuous FDIA and Double Continuous FDIA had on average 100 % detection rate against the simulated attacks. In the figures 4.5 and 4.6 the measurements are clearly going outside of what the system expected and the measurements never converged back to what was expected. In the more subtle Negative FDIA and Random FDIA the detection rate of the system flatlined: The coded detection treshold in 4.7 only took into consideration the values that would be higher than expected while ignoring the lower values. In the random FDIA 4.8 the changes were so subtle that the treshold was not reached. The adaptive nature of the Kalman Filter based IDS expects some changes to the data that were not anticipated. The parameters of the Kalman Filter affects how aggressive the adaptation is. With the current parameters the Kalman Filter's adaptation was not aggressive enough for these small scale abnormalities and the values stayed within expectations. The Big Random FDIA 4.9 had large enough changes that with ten iterations a good chunk of 70 % of attacks were over the treshold although the detection rate fell with more iterations nearing 50 %. By averaging out these simple and complex attacks we get a 60 % detection rate against all attack at 1 iteration,

Table 4.1: Detection Accuracy

Attack Type	10 Iterations	20 Iterations	100 Iterations
Continuous FDIA Detection Accuracy	100 %	100 %	100 %
Double Continuous FDIA Detection Accuracy	100 %	100 %	100 %
Negative FDIA Detection Accuracy	0 %	0 %	0 %
Random Attack Detection Accuracy	0 %	0 %	0 %
Big Random Attack Detection Accuracy	70 %	50 %	61 %

54 % detection rate against all attacks at 10 iterations and 50 % detection rate for 20 iterations. By excluding the simulation cases where the implementation does not take into account the type of attack we can see a detection rate ranging from 83 % to 90 % depending on the number of iterations. Averaging out these we could see a detection rate of 87 %. With small amount of further work against the simulated attacks we could see a detection rate in similar numbers.

4.3.1 Answering RQ1: What Kind Of IDS System Would Be Suitable?

The IDS system implemented is working with the data provided by the simulation and as such it is portable to different kind of environments. The Kalman Filter approach is lightweight for modern hardware as the core operation is mostly calculating with primitive arithmetic and providing an estimation based on those calculations. The amount of datapoints is relatively large, but as the base loop of predicting and updating an estimate and coming up with a variance is mathematically very easy and computers are optimized to work on primitive arithmetic, the implementation is very fast and could work well on low powered hardware.

Running the software with the BSD general command "time" we can get a grasp of the execution speed on a Intel Core i7-9750H CPU @ 2.60GHz:

```
time python3 ./ids.py

python3 ./ids.py  0.63s user 0.15s system 95% cpu 0.819 total
```

The timing is done without showing the plot results of the script as they require user intervention to close as a default. Nevertheless the combined CPU time of the user and the system being 0.819 seconds for the dataset provided should provide us with a clear perspective of how relatively light the computing process is. Especially considering Python being notorious for its slow startup time.[19] Running the software on a headless embedded device like a Raspberry Pi should not prove to be too unefficient. As the detection of FDIA does not need to be real time, the calculation could also be offset to high powered time of the day on any networks with power constraints and requirements for rationing of computing power.

4.3.2 Answering RQ2: Distributed vs Centralized?

The implemented IDS system can be hosted by any part of the system capable of accessing the signal data it is supposed to analyze. The EMS would be a suitable host for a lot of applications as it is handling the energy levels of the grid and has access to all the energy levels. It is also in charge of handling the data so it should have the suitable computing power to perform additional tasks like the IDS system on the side. The distributed option would not have offered any additional benefits as the question of should the data be trusted would have also been accompanied by the question of can we trust the decentralized nodes [7]

Going with the centralized system and running the IDS system on the same level as the EMS system would seem to be the optimal solution for this level of implementation where the availability of different MMG islands is assumed to be good. If the simulated data would have holes in the data caused by different nodes being dropped, then distributing the analysis of the signal data would be beneficial.

4.3.3 Answering RQ3: Which Implementation Option?

The simulated environment proved to be more practical as using physical hardware would not really bring any new data on the table: The IDS system would still have a certain data set to process and work on. Implementing the power network with physical hardware could be seen as busywork as the main goal of the thesis is in developing an IDS as mentioned in the Objective of the Thesis chapter 1.2. The MATLAB/Simulink proved to be a good tool to simulate power networks but lacked in tools available for developing the IDS system inside the MATLAB/Simulink environemnt. Taking the data from the simulation and using existing data processing tools with Python proved to be more portable and pragmatic alternative. MATLAB/Simulink environment can be seen as a walled garden and it is in their interest to keep all the strings in their court by keeping the processing inside their system. Taking just the model simulation data out and doing the processing with Python gave a good decoupling from the MATLAB/Simulink products if the IDS system would be developed further.

5 Conclusions

In this thesis an IDS for the FDIA in EMS has been developed. It is really critical to secure critical infrastructure like the power system from any potential harm. The Multi-Microgrid power environment can be one of the potential future implementations for electricity systems for many homes, companies and settlements. At the heart of the MMG being the EMS that handles the loads and currents in the system, it is key to secure its operations. From the implementation done in the thesis in can be observed that securing a complex system does not have a one simple solution that would suit all the situations.

The suitable IDS system for MMGs for detecting FDIAs should be light weight and capable of handling different kinds of attacks. The Kalman Filter based approach is one of the potential proposals as the mathematical arithmetic is relatively simple for computers to calculate. The simplicity saves energy in situations where it is limited and the system should be relatively easy to reason about. The Centralization of the system with the EMS offers a good view into the data of the grid and lets the IDS system to operate with enough resources and visibility at the same time avoiding the complexities that come up with decentralization. The implementation of the system with simulation tools provides us with a pragmatic and portable view into how the IDS system is working and it frees us from the bounds of setting up the physical system for possible replication of the designed grid. The MATLAB/Simulink toolset offered us a good set of tools for simulating a power network's activities over a period of 24 hours in couple of seconds for signal data. Implementing the IDS system with Python and FilterPy proved to be the most versatile and deployable in platform independent manner. The detection rate for the developed IDS with the initially planned attacks averaged to around 87 %. The detection rate for additional attack types like a subtle random attack was far worse as the treshold for a detected attack was not reached due to the adaptative nature of the Kalman Filter. With far heavier adaptation curves the subtle random attack could have potentially been detected but the system could have been more prone for false alerts due to any noise.

The system still needs further improvement against additional simulated attacks and any possible testing for the simulations should be done with high amounts of iterations to get precise enough result. To further improve any potential detection rates and development of the system, a testing framework that automatically runs the simulations in the background with large enough of iteration count could be organized. The testing framework could also contain different kinds of attacks than the simulation currently contains. It could be a potential subject for a new thesis to come up with a conclusive set of attacks in the testing framework the test the IDS system against. The sensitivity of the IDS to subtle attacks could also be improved by optimizing the threshold level of the IDS. The attacks that are seemingly small can have surprisingly big effects as the MMG system can be in the works for many years. Taking into account months and years long attacks can also be one good point of improvement for the IDS.

For future work the system could work with additional types of cyber attacks and more rare FDIAs. The detection accuracy for the existing simulations could be improved by taking more subtle and complex attacks into account in the implementation. The power system could support islanding and the IDS system could offer dyanmic actions of what to do with the information that an attack is ongoing. The

EMS system could also be more dynamic and respond to different kinds of load variations over several days. Kalman Filter based IDS can also be supplemented with other types of detection schemes running in parallel as the Kalman Filter based system is light weight and does not require lots of computing power.

References

[1] M. N. Alam, S. Chakrabarti, and A. Ghosh, “Networked microgrids state-of-the-art and future perspectives”, 2018.

[2] M. Moghimi, P. Jamborsalamati, J. Hossain, S. Stegen, and J. Lu, “A hybrid communication platform for multi-microgrid energy management system optimization”, 2018.

[3] M. Z. Gunduza and R. Das, “Cyber-security on smart grid: Threats and potential solutions”, 2020.

[4] B. Canaan, B. Colicchio, and D. O. Abdeslam, “Microgrid cyber-security review and challenges toward resilience”, 2020.

[5] T. Vu, B. Nguyen, T. Ngo, M. Steurer, K. Schoder, and R. Hovsapian, “Distributed optimal dynamic state estimation for cyber intrusion detection in networked dc microgrids”, 2019.

[6] M. Ghorbanian, S. H. Dolatabadi, M. Masjedi, and P. Siano, “Communication in smart grids: A comprehensive review on the existing and future communication and information infrastructures”, 2019.

[7] M. H. Cintuglu and D. Ishchenko, “Secure distributed state estimation”, 2019.

[8] M. R. Habibi, H. R. Baghaee, T. Dragicevic, and F. Blaabjerg, “Detection of false data injection cyber-attacks in dc microgrids based on recurrent neural networks”, 2019.

[9] S. Boudko, H. Abie, E. Nigussie, and R. Savola, “Towards federated learning-based collaborative adaptive cybersecurity for multi-microgrids”, 2021.

[10] S. Abhinav, H. Modares, F. L. Lewis, F. Ferrese, and A. Davoudi, “Synchrony in networked microgrids under attacks”, 2018.

[11] Y. Li, P. Zhang, and P. B. Luh, “Formal analysis of networked microgrids dynamics”, 2018.

[12] K. Manandhar, X. Cao, F. Hu, and Y. Liu, “Detection of faults and attacks including false data injection attack in smart grid using kalman filter”, 2014.

[13] R. R. Labbe. “Kalman and bayesian filters in python”, Roger R. Labbe. (2021), [Online]. Available: `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python` (visited on 07/18/2021).

[14] JGraph. “Diagrams.net”, JGraph. (2021), [Online]. Available: `https://app.diagrams.net` (visited on 09/23/2021).

[15] R. R. Labbe. “Filterpy”, Roger R. Labbe. (2021), [Online]. Available: `https://filterpy.readthedocs.io/en/latest/index.html` (visited on 09/23/2021).

[16] Dave. “Explain process noise terminology in kalman filter”, Stackoverflow. (2013), [Online]. Available: `https://stackoverflow.com/questions/19537884/explain-process-noise-terminology-in-kalman-filter` (visited on 10/19/2021).

[17] H. Mita and MathWorks. “Simplified model of a small scale micro-grid”, MathWorks. (2021), [Online]. Available: `https://www.mathworks.com/help/physmod/sps/ug/simplified-model-of-a-small-scale-micro-grid.html` (visited on 06/16/2021).

[18] S. developers. “Scipy, scientific computing tools for python”, SciPy developers. (2021), [Online]. Available: `https://scipy.org` (visited on 07/18/2021).

[19] V. Stinner. “Python startup time”, Victor Stinner. (2018), [Online]. Available: `https://pythondev.readthedocs.io/startup_time.html` (visited on 09/08/2021).