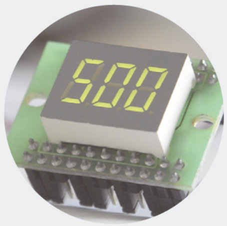
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA ELECTROTÉCNICA - SISTEMAS ELÉCTRICOS DE ENERGIA






Simulation and Analysis of the Operation and Reconfiguration of a Medium Voltage Distribution Network in a Smart Grid Context in MATLAB Simulink

ALEXANDER VAN WAEYENBERGE

Junho de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Electrotécnica
Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto

Simulation and Analysis of the Operation and Reconfiguration of a Medium Voltage Distribution Network in a Smart Grid Context in MATLAB Simulink

Mestrado em Engenharia Eletrotécnica - Sistemas Elétricos de Energia

Alexander Van Waeyenberge

Supervisors:
Dr. Bruno Miguel Da Rocha Canizes
Dr. João André Pinto Soares
Dr. Simon Ravyts

Ano Letivo: 2020-2021

Abstract

This work will present a Medium Voltage Distribution Network that is operated as a Smart Grid. As the distribution infrastructure for electric energy ages and the share of EV's and renewables increases, changes will have to be made to support the increasing power flows in the network. A more long-term solution than reinforcing the network with heavier cables is constructing an intelligent network that reacts to changing power flows inside the network and adapts accordingly to guarantee optimal functionality. External data from an optimisation algorithm is used to determine the switching behaviour. The network was modeled and analysed using a simulation software. If done correctly, a simulation can offer a lot of insight for only a fraction of the cost of constructing and testing the network in reality. MATLAB Simulink was used for the virtual modeling and analysis of the network. The main objective is to construct a model of the MVDN and use it to generate and analyse the power flows in the network to determine the plausibility of exploiting a similar network in an existing city. The models for each of the network components were developed and picked to combine them into a functioning network model based on the smart city's topology. Simulating a smart grid is in essence not novel, but has not been done in Simulink before in this context. The hardest obstacle to overcome during the construction of the network model was finding a way to achieve the making and breaking of network connections in a way that Simulink could compute the network parameters correctly and in a timely manner. A whole section is dedicated to resolving these development issues. Following this, the results of the simulation regarding power flows and losses in the network are discussed. When it comes to the renewable generation implementation, the results showed promising results, even on days with low wind velocity the renewables aided in reducing the power demanded from the substation. The total generated power is compared with the total consumed power in the loads to find the grid losses. It became apparent very quickly that the grid losses were very high, up to 9.7%, which is a lot higher than the expected 2-6%. Overall the model showed promising results, as well as serving as a baseline for future works to improve upon.

Contents

List of Figures

List of Tables

Glossary

DER Distributed Energy Resources. 4

DLMP Distributed Locational Marginal Pricing. 2

DN Distribution Network. 14, 31

DPF Displacement Power Factor. 14, 15

GECAD Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development. 3

HVDN High Voltage Distribution Network. 8, 30

ICT Information and Communication Technologies. 5

LVDN Low Voltage Distribution Network. 32

MVDN Medium Voltage Distribution Network. i, ii, 1, 2, 4, 5, 7–9, 13, 16, 17, 19, 20, 22, 24–28, 30, 31, 33, 35, 42, 43, 52, 57, 58, 63–65

ODE Ordinary Differential Equation. v, 20–25, 40, 42

PMU Phasor Measurement Units. 5

PV Photo Voltaics. 1

RES Renewable Energy Sources. 5, 6, 16

SC Smart City. 2, 4

SG Smart Grid. 1, 2, 4, 6, 22, 27, 28, 35

SLG Switching, Load and Generation. 40

Acknowledgements

I would like to start off by thanking KULeuven and ISEP for giving me the opportunity to go on an international student exchange. I want to thank Bruno Canizes, João Soares and Simon Ravyts for guiding me through the process of writing a master thesis. Besides learning about simulations and medium voltage distribution networks in a smart grid context, I learned a lot about myself as a person here. This experience was unforgettable and I will look back on my time in Porto with a big smile on my face.

Verder zou ik ook graag mijn ouders bedanken, die altijd in mij geloofden. Zij hebben mij de kans gegeven om te doen wat ik het liefste doe, op school vlak en daarbuiten, en hebben daarbij mede bijgedragen tot de persoon die ik nu ben. Mijn broer Nicolas en zus Sarah zou ik ook graag willen noemen. Ik ben heel trots op hen en ook zij zijn elk op hun eigen manier een grote steun geweest. Als we het dan over Porto hebben, zou ik graag La Colove en in het bijzonder Ismael Zighed willen bedanken. De ontelbare uren die ik in goed gezelschap daar heb doorgebracht, werkend aan deze thesis, zullen mij zeker bijblijven. Verder gaat mijn hart ook uit naar The Fellowship, stuk voor stuk topkerels die bij hebben gedragen tot een onvergetelijke ervaring hier. Alessandro, Philipp, Kevin, Marco en Ismael, ik zie jullie snel weer. Op deze manier zou ik ook de andere mensen willen bedanken die mijn tijd hier onvergetelijk hebben gemaakt: Breyner House, en in het bijzonder Lotta en Sarah, Joy house en Villa Moreira, mijn huisgenoten. Jullie weten wie jullie zijn. Ik kan mijn dankbaarheid naar iedereen genoemd hierboven maar moeilijk in woorden uitdrukken.
Thank you for everything, see you soon!

Chapter 1

Introduction

In the following chapter a brief explanation will be given about the goals and objectives of this thesis, the main problem this work is trying to contribute to solving and the motivation for chipping in to the solution. Requirements for the understanding and recreation of the thesis and the overall structure of the dissertation can also be found in this chapter.

1.1 Problem

In a rapidly changing globalised world where combustion engines and fossil fuels are slowly phased out [1] in favor of more ecological and durable energy solutions, the current Medium Voltage Distribution Networks (MVDN) will have to adapt accordingly. As one might imagine, if everyone comes home from work in the evening and plugs in their electric vehicle (EV) at the same time, there will be a surge in the electric energy demand. This impact on the grid can not be underestimated [2]. Combining this with the fact that the overall penetration of photo-voltaic (PV) and wind energy has been increasing [3], the distribution network should also be able to handle the rapidly changing power flow in the grid due to the fluctuating character of renewables [4], for networks with a radial topology, this is however not obvious [5]. This implies quite a few challenges along the way as the current MVDN is not ready yet to handle this more complicated approach to energy distribution. Making the necessary changes to adapt the network to its new workings and exploiting the MVDN as a smart grid (SG) looks like the way forward as it offers a lot of advantages over the more traditional approach of having central production units supply the downstream consumers with electrical energy [6].

1.2 Motivation

The evolution of people moving to big cities in favor of the countryside in recent years has a big impact on the energy needs and quality of the air in these cities. Cars and fossil fuel energy plants contribute for a big part to the air pollution present. EV's and renewable energy sources make for an excellent solution to this problem. It is in everyone's interest to make the city life appealing and healthy and uphold and improve the overall quality of life. With this work, I hope to make a small contribution to a healthier and more sustainable way of living in Smart Cities (SC).

1.3 Objectives

The goal of the following work is to give a deeper understanding in the solid state behaviour of a SG using a modeling software, in this case MATLAB simulink. Similar works have been made in the past, but then with different software packages [7] [8]. This inexpensive method can give insight in the behaviour of the network in a lot of configurations without the need to build and test a real MVDN. Moreover, the model can give us a good idea on how to optimally implement certain parts of a similar network in an actual city.
The first part to achieving this lies in the development of and/or use of models for each of the different components of the SG. Famous in the computer science and information theory is the saying: "Garbage in, Garbage out". The same holds true for models. If a model of a component does not behave closely to the real object, any input in this model will result in an output that doesn't come close to the behaviour of the object that it was modelled after. This implies that we have to take extra care when developing/picking a model for each of the components. Simulink includes a lot of toolboxes which contain many tried and true models for a lot of the components used in the MVDN.
Once the models for the different components have been decided on, the model is assembled as a whole and collected data from an optimal distribution grid operation algorithm using DLMP-based pricing [9] from 19/03/2017 will be fed to the MVDN. The output data from the model will then give us a good idea on how the SG behaves at different times of the day.
Along the way, a few challenges had to be overcome to extract useful information from the MVDN simulation. One element in particular, responsible for connecting and disconnecting certain busses from each other, proved to be quite a challenge to implement. The complications this component created, along with the possible causes and the solutions proposed will be presented in this work. Afterwards, the results of the simulation will be shown and discussed. As a final objective, a conclusion will be drawn from the obtained results and suggestions on how to expand on this work will be given. It is important to remark that

the definition of the objectives of this work has benefited from the interaction with national and international RD projects coordinated by or having the participation of the Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), where this thesis has been developed. The breakthrough nature of these projects has enabled this thesis to consider innovative perspectives that have helped to enrich this work. The considered project is: CENERGETIC – Coordinated energy resource management under uncertainty considering electric vehicles and demand flexibility in distribution networks, reference PTDC/EEI-EEE/29893/2017.

1.4 Requirements

As the main goal of this thesis is to obtain usable results regarding the operation and reconfiguration in a smart grid context in simulink, a MATLAB/simulink license of version 2020b or above is required to run the model and simulations discussed in the following chapters. Besides this, an understanding of the basic laws of electricity, electrical components and the different parts of an electrical energy grid are required to thoroughly understand the following work.

1.5 Structure of the Thesis

The following thesis will contain 5 chapters. The first one being the introduction, where the motivation for this work, the objectives that have been set and the requirement for this dissertation were listed.
Following this chapter, we get to the 2^{nd} chapter, where information about MVDN in a SG context is listed, compiled down to a summarized and digestible format, to give the reader more insight into the matter before continuing with the rest of the work. The discussed works in this chapter should give some deeper insight on the subject and serve as a reference for the work that will be presented in the following chapters.
The 3^{rd} chapter will start off with the reasons MATLAB Simulink was chosen as the software environment to run the MVDN simulation on. This is followed by an explanation on how simulations in Simulink are evaluated by a computer. The different types of solvers used to solve the inherent differential equations of the model are compared against each other until a suitable solver is decided on. The development and implementation of every model used inside simulink will be discussed after. A deeper look will be taken at every component chosen and how it will be used to reinforce the model. This will explain how we go from the BISITE's (https://bisite.usal.es/en) mock-up SC with 13-bus MVDN and high penetration of Distributed Energy Resources (DER) to modelling the entire network inside the Simulink environment.
The 4^{rd} chapter starts off with an explanation of the challenges faced on the way when coming up with a working MVDN model, the problems encountered and the possible solutions. The remaining part of this chapter will consist of the discussion of the results obtained by running the model. This chapter will contain the network data condensed down into graphs. To finish the structure of the thesis we will end with a general conclusion about the dissertation and suggestions on how to expand this work in the future.

Chapter 2

Medium Voltage Distribution Network in a Smart Grid Context

Chapter 2 will shortly go over some literature regarding smart MVDNs. A deeper look will be taken into each of the MVDN components and how they make up a distribution grid as a whole.

Before going over MVDNs and the elements out of which they consist, a look will be taken at the *smart* part of a smart MVDN. *Verbong G et al.* published a paper about smart grid experiments in the Netherlands where they state the increasing introduction of renewable energy sources and the popularisation of 's and heat pumps prove to be a challenge for the aging energy distribution infrastructure [10]. The simple solution suggested is the reinforcement of the network by replacing the old cables with heavier ones. This might prove to be only a short term solution as the share of renewables, EV's etc. in the network will keep increasing at an unknown pace. The more long term approach is to upgrade to a more intelligent distribution networks, called smart grids. [11] [12] [13] These utilise Information and Communication Technologies (ICT) combined with advanced measuring devices such as Phasor Measurement Units (PMU) equipped with GPS receivers that allow for synchronisation of data from varying distances [14]. This way the grid is made more 'intelligent' as changes to the network connections can be made on the fly, based on real-time info across the entire network. Certain network components are even specifically designed and tailored to be used within a smart grid network [15] [16]. All these combined makes the network more robust and sustainable as well as improving the overall efficiency by reducing the network losses [17].

According to *Lund et al* [18] there are three phases in the implementation of renewable energy sources (RES). Most European countries are still in the *introduction phase* where there is no or only a small share of renewable energy in the electrical energy system. SGs will offer a big advantage over traditional grids when we get to the second phase, *The large-scale integration phase*, where a large part of the energy in the system has a renewable character. Due to the higher penetration of these RES and their dependence on wind, solar radiation etc., the direction and magnitude of the power flow can change drastically within a very short time span. This implies that the stability of frequency and voltage will have to be monitored closely as the network frequency is linked to active power balance in the grid and the voltage supplied is linked to the imaginary power balance [19]. The third phase is the *100% renewable energy phase* where all demand of electric energy is met by renewable energy generation. This is definitely something for the far future as there are a few concerns right now regarding 100% renewable energy. First of all the scale at which renewable energy is used would have to go up immensely to compensate for the disappearance of traditional and non-renewable resources. It will take a lot of time and resources and as of now there is not much economical incentive to make this transition. The second concern is the irregular character of most renewable energy. Solar and wind for example are highly dependant on solar irradiation and wind velocity. Factors that lie out of human control. This implies a challenge to meet the growing demand of electrical energy at times where RES alone are not producing sufficient power to meet this demand. The use of energy buffers might provide a solution here.

2.1 Distribution lines

To be able to speak of a distribution network, Electrical energy needs to be distributed over medium to long distances. The focus in this distribution lies on keeping the distribution losses as low as possible. Apparent power is transported through the lines. Take the following equation into account:

$$\overline{S} = \overline{V} \cdot \overline{I}^* \tag{2.1}$$

Where S stands for apparent power, V for voltage and I for current, with the bar signifying complex notation and the asterisk standing for complex inverse notation. Most of the losses in a distribution line have a thermal nature. Since these losses are linked to the square of the current flowing through the line $P_{th} = R \cdot I^2$, we can reduce the losses by transforming the voltage to a higher one before transport (see substations later). By increasing the voltage level, the current lowers for transport of the same amount of power. These thermal losses will definitely have to be taken in account when developing a distribution line model. The second type of distribution losses that impact the efficiency of the lines are parasitic capacitance losses. When two conductors with a high difference in potential electrical energy come close to each other, charges of opposite polarity will arise in each of the conductors due to the electric field between them. This basically means they work as a capacitor. This parasitic capacity will draw current to nullify the built up charge in the conductors as described in the following equation:

$$i = C\frac{dv}{dt} \tag{2.2}$$

Where

$$C = \frac{Q}{V} \tag{2.3}$$

As MVDN usually go up to 35kV and can be quite lengthy, the leakage currents associated with these parasitic capacitances can not always be neglected.

2.1.1 Transmission line theory

To model a transmission line, we should start from the physics equations that describe the behaviour of the line. In 1876, Oliver Heaviside published the first work from his *Transmission Line Model*. This model shows that electromagnetic waves are reflected along a transmission line where the sent signal and each of its reflections can be described using the *Telegrapher's equations*. These are two partial differential equations that describe the relationship between voltage and current along the transmission line, as a function of time t and place z [20].

$$\frac{\partial}{\partial z}v(z,t) = -R \cdot i(z,t) - L \cdot \frac{\partial}{\partial t}i(z,t) \tag{2.4}$$

$$\frac{\partial}{\partial z}i(z,t) = -G \cdot v(z,t) - C \cdot \frac{\partial}{\partial t}v(z,t) \tag{2.5}$$

where L, C, R and G are given per unit of distance. Both equations can then be combined to write an equation that only depends on one variable:
For the voltage:

$$\frac{\partial^2}{\partial z^2}v(z,t) - LC \cdot \frac{\partial^2}{\partial t^2}v(z,t) = (RC + GL) \cdot \frac{\partial}{\partial t}v(z,t) + GR \cdot v(z,t) \tag{2.6}$$

For the current:

$$\frac{\partial^2}{\partial z^2}i(z,t) - LC \cdot \frac{\partial^2}{\partial t^2}i(z,t) = (RC + GL) \cdot \frac{\partial}{\partial t}i(z,t) + GR \cdot i(z,t) \tag{2.7}$$

where $LC = \mu\epsilon$ with μ signifying the absolute magnetic permeability and ϵ signifying the absolute electric permittivity.

It can be noted that in High Voltage Distribution Networks (HVDN), the inductive reactance per km and capacitive reactance per km can be much larger than the resistance per km and conductance per km, respectively, of the line. In this case, the resistance can be neglected ($R = 0$) relative to the inductive reactance and the conductive reactance neglected ($G = 0$) relative to the capacity. This means the right side of both equations 2.6 and 2.7.
We are however talking about MVDNs where this is rarely the case as the values of resistance and conductivity are often too close in magnitude to the reactances.

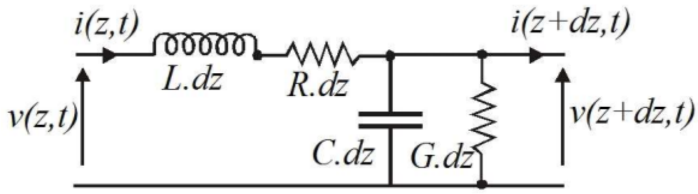


Figure 2.1: Transmission Theory Model
[21]

In figure 2.1 you can find the Model used to visualize the transmission line theory. We find the voltage and current at the beginning of the line in function of the distance z and time t. The end of this piece of transmission line is written as $z + dz$, where dz is an arbitrary distance increment. The difference in state of the voltage and/or current wave at point z and point $z + dz$ is dependant on the distance increment dz and time increment which means that a longer power line will result in voltage and current in beginning and end being more out of sync. It is interesting to think about at which increment dz voltage/current are most out of sync. Europe, as well as most other regions in the world exploit their power grids with a fixed frequency of 50Hz. The relation between frequency and wavelength is connected with the speed of light:

$$c = f \cdot \lambda \tag{2.8}$$

at a frequency of 50Hz and the speed of light being 300,000 $\frac{km}{s}$, we find that the wavelength for a 50Hz electrical signal is 6000km. At half the wavelength and at a certain time t, the difference in position of the voltage wave at the beginning of the line and at the end will be exactly 2 times the amplitude.

2.2 Substations

As discussed in the section about distribution lines, it became apparent that electrical substations play a very important role in the workings of the electrical grid. To go from one distribution network to another, it is often necessary to go from one voltage level to another or shift the phase angle θ to ensure a proper connection between the two networks. When we talk about high power applications, this transformation of voltage level or shift in phase angle is almost always done with a transformer. The MVDN that is discussed in this work has a line to line voltage of 35kV. Substations will be needed to transform the high voltage feeding the MVDN down to 35kV and to transform the 35kV medium voltage down to a usable voltage on the low voltage distribution grid.

2.2.1 Transformer construction

Transformers have a primary and secondary coil, both with a number of turns N_1 and N_2 [22]. Each coil is wrapped around a side of an iron core. If a voltage is applied over the first coil, a voltage over the second coil will be induced. This secondary voltage can be higher or lower, depending on the configuration of the transformer.

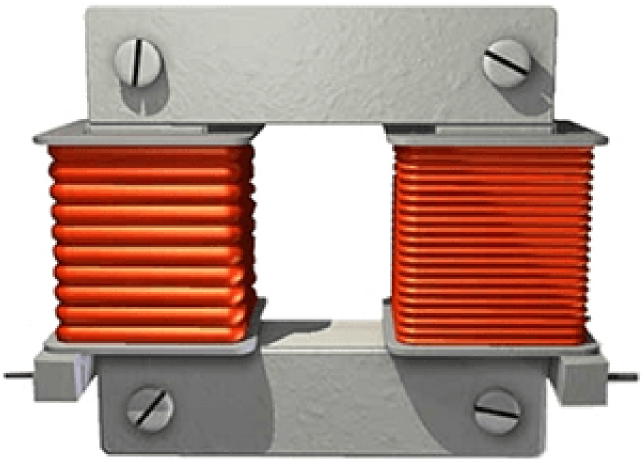


Figure 2.2: Transformer
[23]

The physical principle on which the workings of the transformer rely is Faraday's law:

$$e = -\frac{\mathrm{d}\phi}{\mathrm{d}t} \tag{2.9}$$

If an alternating voltage is applied to the first coil, Faraday's law tells us that a magnetic flux is being imposed in the core. This flux, that lags 90° behind on the voltage, will then travel trough the material of the core until it reaches the second winding along its path. Here the flux will induce a voltage in the windings of the second coil. The ratio of voltage on the first coil to voltage on the second coil is determined by the number of turns on each of the coils according to the following equation:

$$\frac{e_1}{N_1} = \frac{e_2}{N_2} \tag{2.10}$$

This relation is however only true for an ideal transformer. An ideal transformer has what we call a ”perfect coupling” between the primary and secondary coil. This implies that there is no flux leakage from the iron core and all the magnetic energy induced in the core by the primary coil is converted back to electric energy in the secondary coil, this basically mean the reluctance of the magnetic core is equal to 0. The second characteristic of an ideal transformer is that there are no core- and copper losses present.

2.2.2 No-load losses

The first type of transformer losses, called the no-load losses, are associated with the core of the transformer. It is a misconception that these losses only occur when the transformer is not connected to a load. The name just implies that the losses always happen, no matter the power drawn from the secondary side of the transformer. Core losses occur when an alternating flux passes through the iron core of the transformer. They are twofold in nature. On one hand there is the hysteresis losses[22] [24], they are caused by the residual magnetism in the core each time the polarity of the flux changes. It takes work to undo this residual magnetism, which expresses itself in generation of heat. The hysteresis losses have a thermal nature and can proportionally be expressed as active power lost, by the empirical law of Steinmetz:

$$P_{L,hys} = \ f \cdot (\hat{B})^{1,6} \tag{2.11}$$

It has to be noted that the losses are expressed in $[\frac{W}{kg}]$ where $\hat{B}$ stands for the peak magnetic field.
On the other hand there are the losses due to Eddy currents in the core. Because of the Joule effect, these currents will cause the core to heat up and contribute to the overall core losses. The Eddy current losses are proportional to the following identity:

$$P_{L,eddy} = \ \frac{(f \cdot d \cdot \hat{B})^2}{\rho_{Fe}} \tag{2.12}$$

Where d stands for the thickness of the laminations of the core, and ρ_{Fe} indicates the resistivity of the core material.
The hysteresis and Eddy current losses together make up the total core losses. If we take the mass of the core into account:

$$P_{L,core} = m_{core} \cdot P_{L,hys} + m_{core} \cdot P_{L,eddy} \tag{2.13}$$

Hysteresis losses can be reduced by constructing the core out of an easily magnetisable (and de-magnetisable) material, this way the area under the hysteresis loop, which is proportional to the work required to remove residual magnetism, is smallest. Electrical steel is often used because of its favourable magnetic properties. The losses due to Eddy currents can also be heavily reduced by constructing the magnetic core out of thin (0,3-0,5mm) metal sheets, made out a material with a high resistivity, that are also electrically isolated from one another. Paper or a varnish is often used for this purpose. The characteristic buzzing sound that transformers make is also attributed to the core losses. This is caused by magnetostriction, the phenomenon whereby magnetic materials change shape while being magnetised making the laminations vibrate against each other, producing the characteristic sound.

2.2.3 Load losses

The second type of losses that occur in a transformer is when a load is connected to it. These are called the *load losses*. These are caused by the equivalent impedance Z_e of the transformer. This impedance can be split up in an equivalent resistance R_e and an equivalent reactance X_e. The losses caused by the resistance are called the copper losses and can be defined as follows:

$$P_{cl} = R_e \cdot {I_1}^2 \tag{2.14}$$

The equivalent reactance can then be easily found:

$$X_e = \sqrt{{Z_e}^2 - {R_e}^2} \tag{2.15}$$

In analog fashion to equation 2.14, the losses due to the equivalent reactance and impedance as a whole can be found.

2.2.4 Equivalent circuits

Besides the ability to transform voltages up or down, a transformer possesses another very useful property. The primary and secondary side are in complete galvanic isolation to each other. This makes it so they can isolate an entire network from ground. They are commonly used as safety transformers for this reason as they lower the risk of electrocution significantly due to the leak reactance limiting the fault current.

This same property makes it hard to make calculations with transformer in their galvanic isolation form. Figure 2.3 shows an equivalent circuit of a single-phase transformer. Here you can recognise the windings on primary and secondary side with R_0, R_1 and R_2 making up the resistances that cause the core losses, copper losses on primary side and copper losses on secondary side respectively. X_0 symbolises the reactance where the magnetisation current I_m passes through. This current is responsible for maintaining the flow of magnetic flux in the transformers core. The losses associated with this current are also counted towards the no-load losses as they remain constant regardless of the load of the transformer. X_1 and X_2 signify the equivalent reactances of the transformer related to the load-losses.

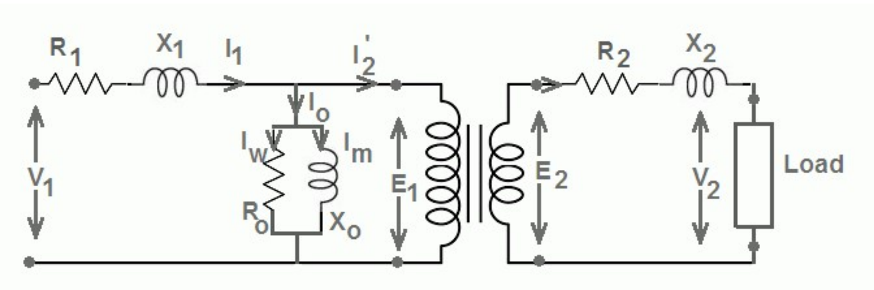


Figure 2.3: Equivalent circuit of a transformer
[25]

To circumvent the problem of a transformer model in this form being hard to work with, the equivalent T-circuit model was invented. The idea is here that in this model, the galvanic isolation is taken away and all the voltages, currents and impedances are referred to one side, generally the primary side.
Since the galvanic coupling is gone now, the model parameters are now dictated by the transformation ratio k. The relations between parameters when transferred from secondary to primary is as follows: For the voltage:

$$E_1 = k \cdot E_2' \tag{2.16}$$

For the current

$$I_1 = \frac{I_2'}{k} \tag{2.17}$$

Combining both to find the impedance. Resistance and reactance can be found in a similar way.

$$Z_1 = \frac{E_1}{I_1} = \frac{k \cdot E_2'}{\frac{I_2'}{k}} = k^2 \cdot Z_2' \tag{2.18}$$

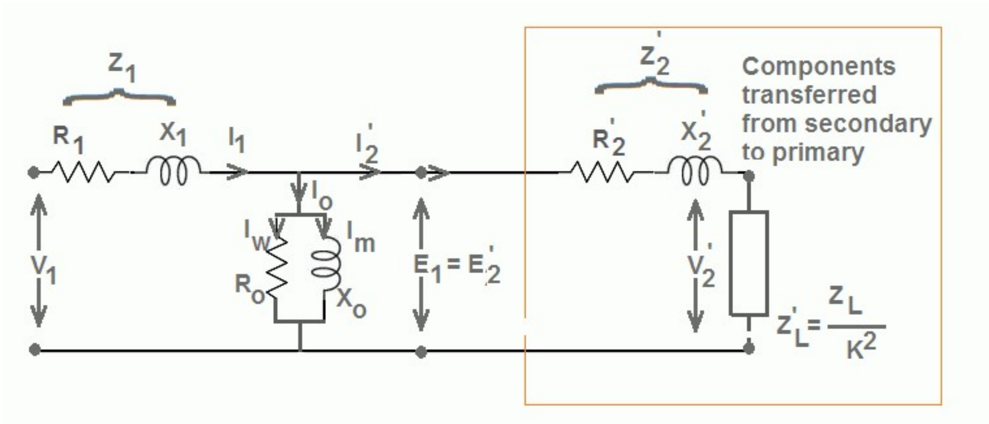


Figure 2.4: Equivalent T-circuit of a transformer referred to primary [25]

2.3 Switches

The next vital component in a MVDN are the power switches. To make a functioning smart grid, the different busses should be able to disconnect from one another fairly easily. To achieve this, a snubber resistance and/or capacitance can be added in parallel. The goal is to limit transient voltage peaks when a switch happens. In a MVDN, the load almost always has an inductive nature. If the current is suddenly cut, a transient voltage peak will occur. This transient voltage is even capable of re-igniting the arc caused by opening the switch [21]. The magnitude of this voltage is dependant on the rate at which the current changes with respect to time and the inductance of the load that is being switched:

$$V_{tr} = L\frac{di}{dt} \tag{2.19}$$

The resistance (in combination with a capacitor) will dampen this transient by providing a path for the current to flow when the switch opens, lowering the rate of current change and reducing the peak voltage over the load. This is beneficial as arcs are undesirable and make it harder to separate the two sides from each other. The increased voltage over the load can also damage certain network elements.

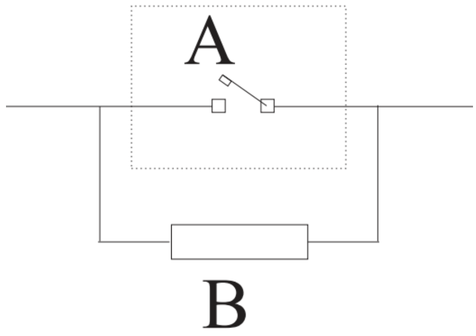


Figure 2.5: Resistive snubber
[21]

2.4 Capacitor banks

A big reason for the use of distribution lines over high voltages to distribute electrical energy is efficiency. Great effort is put into reducing the loss of energy and raising the efficiency of an energy system, especially if the energy is in a noble and highly usable form like electrical energy. Besides distribution losses, harmonic distortion and the displacement power factor play a big role when it comes to loss of electrical efficiency. How to reduce distribution losses is discussed in the section in this chapter about distribution lines. Harmonic distortion also has to be taken into account since harmonic currents with a frequency above that of the base frequency do not contribute to the delivery of active power. The total Harmonic Distortion for the current (THD_I) can be written as:

$$THD_I^2 = \frac{\sum_{i=2}^{n} I_i^2}{I_1} \tag{2.20}$$

These currents will take up capacity in the lines and possibly congest and in some cases overload the DN. To reduce the presence of these higher harmonic currents, low pass filters can used. The working of these low-pass filters depend on the frequency-sensitive impedance of capacitors.

$$Z_c = \frac{1}{2\pi \cdot f \cdot C} \tag{2.21}$$

As you can see, 2.11 shows that the higher the frequency of the current passing through a capacitor, the lower the impedance of the capacitor will be. This means that for high frequency currents, a capacitor between line and ground is basically a short circuit and these currents will flow to ground, resulting in a lower amount of harmonic currents, which means the THD will drop and the efficiency will go up. The third factor that can severely impact the efficiency of an electrical system is the displacement power factor (DPF).

$$DPF = \frac{P}{V \cdot I_1} \tag{2.22}$$

where I_f1 stands for the base frequency current.
As the formula suggests, the DPF (commonly called the $cos\phi$) indicates how much the current lags/leads the voltage and in turn causes a portion of the apparent power to be reactive. This will once again lower the efficiency of the electrical system as the reactive currents will congest and possible overload the network. Most of the electrical loads fed by a distribution network have either a resistive or inductive nature. This means in most cases the current will lag behind the voltage and the reactive power will be positive. To balance out this positive reactive power, we use what is called a "capacitor bank". As the name suggests, these are devices made up out of capacitors. For three-phase grids, three capacitors are connected in Y- or Δ - configuration to one another and connected to the power lines in parallel. The capacitor banks will provide the negative reactive power necessary to compensate for the positive reactive power on the grid, drawn by the inductive loads present. In case the reactive power on the grid is constant, we can easily calculate the necessary capacity for each of the capacitors to compensate the reactive power present and raise the DPF. We start with a certain amount of reactive energy Q that needs to be compensated for. this means the capacitor banks have to provide the reactive power $-Q$. We have to go from necessary amount of reactive power to the impedance value and thus capacity of the capacitors. Each capacitor bank has three capacitors, one per phase. Each capacitor can consist out of multiple actual individual capacitors in series or parallel, for simplicity reasons we use an equivalent capacity in the formula. Now the equivalent capacity value for each capacitor can be calculated. The only difference between a $Y-$ and Δ-configuration, is the voltage over each equivalent capacitor. This will have an influence on the necessary capacity as you can see below.
for a Y-configuration:

$$Q = \frac{3 \cdot (V_{LL}/\sqrt{3})^2}{X_c} = V_{LL}^2 \cdot 2\pi \cdot f \cdot C_{eq,Y} \tag{2.23}$$

for a Δ-configuration:

$$Q = \frac{3 \cdot V_{LL}^2}{X_c} = 3 \cdot V_{LL}^2 \cdot 2\pi \cdot f \cdot C_{eq,\Delta} \tag{2.24}$$

For both equations, the capacity can easily be found:

$$C_{eq,Y} = \frac{Q}{V_{LL}^2 \cdot 2\pi \cdot f} \tag{2.25}$$

$$C_{eq,\Delta} = \frac{Q}{3 \cdot V_{LL}^2 \cdot 2\pi \cdot f} \tag{2.26}$$

2.5 Conclusions

In this chapter we took a look at what the literature had to say about smart grids and MVDN in general. We started off with the advantages of making the transition to this kind of network and what this means for future expansion of RES penetration in networks like these.
Next a look was taken at the different components out of which a MVDN exists. The components were described and their use in the network explained. We dove deeper into the distribution lines taking care of electrical transport in the MVDN through the transmission line theory and use this model (although in a permitted simplified form) to set a base for the modeling of these same distribution lines. The electrical substations where next. The transformers used in these substations where explained both in working as well as in the types of losses associated with them. For an accurate simulation, these will all be factors that have to be taken into account. Equivalent transformer circuits where taken into consideration to make the modeling easier without much loss of accuracy. Switches and capacitor banks, particularly how to dimension them properly was briefly explained in this chapter.

Chapter 3

Modeling an MVDN Inside MATLAB Simulink

In this chapter we will first take a closer look at the methodology of the work. After, we continue with comparing some of the software environments that enable the simulation of a MVDN, this will allow us to find the best tool for the job. To finalise this chapter, we will go into detail on the models of each of the primary components of the MVDN. These will be combined into a usable MVDN model, alongside the MATLAB scripts that aid in the flexibility and automation of the collection of data from the network.

3.1 Methodology

To simulate and analyse the operation and reconfiguration of a MVDN, first it is primordial to construct a high-quality model of the network. For every component inside this network, a model is constructed. Using these individual models and the topology of the network that has to be modeled, an entire network model can be constructed. The data on switches, loads and generation in the network for every period is fed to the model, which will then produce the network results in the form of currents, voltages and powers for this period. The periodical data is then combined into a format spanning the entire simulation time. This concatenated data can then be used to analyse the operation and performance of the network.

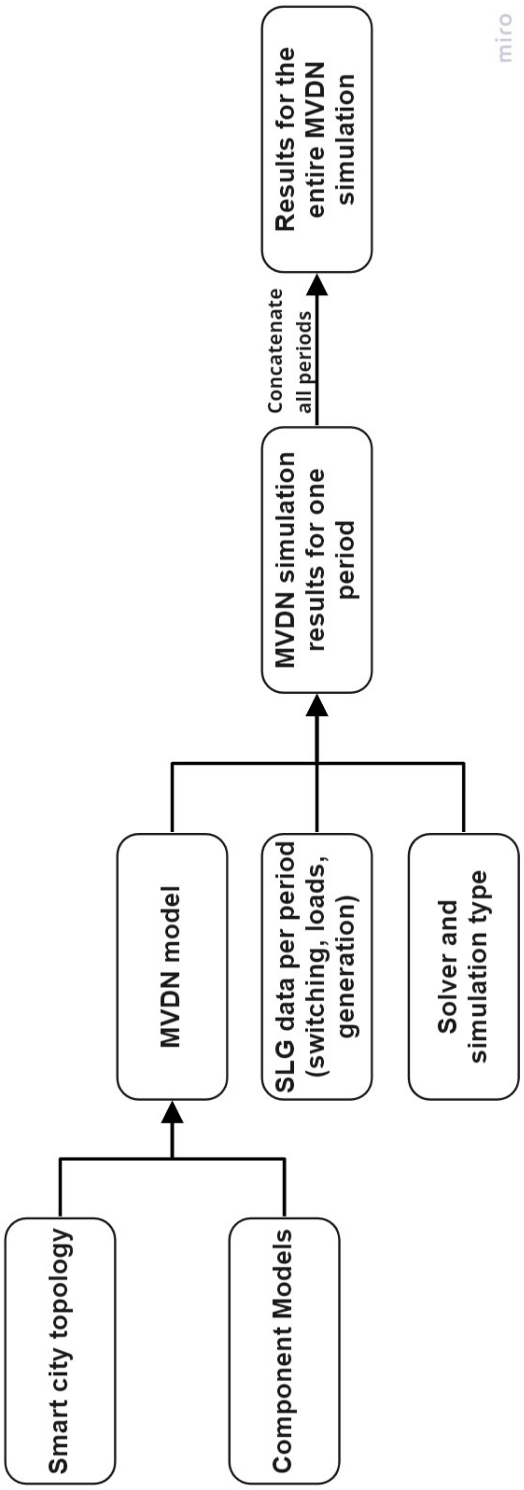


Figure 3.1: Methodology

3.2 Software environment

In the following section, a quick overview of the most widely used numeric computing environments are displayed in table 3.1. All the software packets are usable on all the main operating systems. The most important thing that will have to be taken into account when it comes to an operation and reconfiguration analysis of a MVDN is the modeling environment of the software. MATLAB, GNU Octave and Scilab all have their respective modeling environments.
The main advantage of MATLAB over the other two competitors is the extensive library of models provided. The wheel doesn't need to be reinvented so the libraries will be used extensively to model the BISITE Smart City. The biggest downside to MATLAB seems to be the high cost compared to the other two free software packages. An annual license will set you back $800. You can also opt for a perpetual license, this one goes for $ 2000. It is worth noticing that GNU Octave was built with MATLAB users in mind. They are very similar in usage and functions. Solely because of the size of the available models and the support associated with MATLAB simulink, this software package will be used for the modeling of the MVDN.

Table 3.1: MATLAB vs GNU Octave vs Scilab Comparison[1]

Factors	**MATLAB**	**GNU octave**	**Scilab**
Open source	No	Yes	Yes
Libraries/toolboxes	Lots	Limited	Limited
Platform	All	All	All
Modeling package	Simulink	built in	Xcos
Forums, Help, Manuals	All	Manual, Forums	Manual
Cost	High	Free	Free

[1]https://www.mathworks.com/products/simulink.html
https://www.gnu.org/software/octave/index
https://www.scilab.org/software/xcos

3.3 Simulation in general

Translating a MVDN into a usable computer model is a very interesting challenge. On one hand the goal is to come up with a Simulink model that behaves as close to the real network is possible. Only if the behaviour is adequately similar, useful data can be gathered from the network data and valuable conclusions can be drawn from the analysis this data [26]. Computing power and time are finite and compromises will have to made along the way to ensure that the simulation can produce results in a timely manner, as the time span over which you want the behaviour of an MVDN to be known can be days, weeks or even years.

The model developed for this dissertation was most commonly used to simulate the MVDN behaviour based on 15 minute intervals of data during a full day (24h). It is however capable of simulations with different time intervals at which the data is presented to the model and a different duration of the total time over which the behaviour of the MVDN is wished to be known.

3.3.1 Simulation Settings

To achieve accurate results from a simulation, not only modeling each of the components as close to real behaviour is important, but also the simulation settings play a huge part. The model can be perfect and still give sub-optimal results due to the simulation settings not being very compatible with the to be simulated system. This is why we will dive a little deeper into what makes up these simulation settings.

3.3.2 Solver

To simulate a dynamic system, the different states of the system are calculated at consecutive time intervals over a determined total time span. If all the states of the system are successfully calculated, the system is *solved*. A *solver*, meaning the algorithm that 'solves' the simulation, or more precisely calculates the states of the simulation whenever necessary, is actually solving an *ordinary differential equation* (ODE) [27] that corresponds to the model. In the case of Simulink these ODE's contain one or more derivatives of a dependent variable, in this case y, with respect to the independent time variable t. The highest order derivative that appears in the equation is called the order of the ODE. An example of a second order ODE can be:

$$\frac{d^2y}{dt^2} = 2\frac{dy}{dt} - 6y \tag{3.1}$$

starting from an initial condition y_0 at time t_0 the ODE can then be solved iteratively by applying a particular algorithm to the initial condition and after that the results of the previous step, over a total time of t_n. This goes on until the current time t_x equals t_n. The solver will then return an array with the timestamps of every iteration $t = [t_0, t_1, t_2, ..., t_n]$ and their respective results $y = [y_0, y_1, y_2, ..., y_n]$.

The solvers in MATLAB are only able to solve first order ODE's, these can be in different forms:

- ODE's in explicit form $\frac{dy}{dt} = f(y, t)$
- ODE's in linear implicit form $M(t, y)\frac{dy}{dt} = f(t, y)$ with $M(t, y)$ being a non-singular mass matrix. This matrix can be state- or time-dependent, as well as a constant matrix. Encoded in the matrix are the linear combinations of the first derivative of y involved in linear explicit ODE's. they can be transformed into explicit form $\frac{dy}{dt} = M^{-1}(t, y)f(t, y)$
- ODE's in fully implicit form $f(t, y)\frac{dy}{dt} = 0$. These can not be rewritten in explicit form. There are however some solvers that are specifically designed to solve this type of ODE.

Any amount of coupled ODE's can be solved, as long as there is sufficient computing power and memory.

For n coupled ODE's, with the notation of $\frac{dy}{dt}$ being replaced by y':

$$\begin{pmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_n \end{pmatrix} = \begin{pmatrix} f_1(t, y_1, y_2, ..., y_n) \\ f_2(t, y_1, y_2, ..., y_n) \\ \vdots \\ f_1(t, y_1, y_2, ..., y_n) \end{pmatrix}$$

Since the solvers can only solve first order ODE's, substitutions are used to transform the higher-order equations into a corresponding system of the first order.

$$\begin{aligned} y_1 &= y \\ y_2 &= \frac{dy}{dt} \\ &\vdots \\ y_n &= \frac{d^{n-1}y}{dt^{n-1}} \end{aligned}$$

Using these substitutions, the higher order ODE is solvable by the algorithms.

There are multiple algorithms provided by the Simulink software to solve a model. all of them have their advantages and disadvantages [28]. The first choice one has to make regarding the solver used for the Simulink simulation, is whether it has to be a fixed-step or variable-step solver. The nature of the simulation you want to run definitely has to be considered when deciding on a variable- or fixed-step solver. To understand why, the difference between both will first be explained. As stated above, when solving a system, the different states of the system are determined at sequential time intervals. What these consecutive time intervals look is different in both types of solver.

3.3.2.1 Fixed-step solver:

As the name implies, the time intervals or steps are fixed, meaning the time between each computation is the same for the whole simulation [29]. As this type of solver does not have an error control functionality built into it, getting a more accurate simulation result directly depends on lowering the step size, which will go at the cost of higher computing times. A fixed-step solver is best used when the magnitude of time constants in the system to be simulated is about the same for all the components in the network. If there is a subsystem with a very low time constant (e.g. electrical) combined with a system that possesses a very high time constant (e.g. mechanical inertia of a concrete-breaker starting up), the step size will have to be set really low if an accurate simulation of the low time constant part is desired, even though this low step size is not necessary for the accurate solving of the high time constant part. A compromise needs to be made between accuracy and time necessary to simulate. The extent to which there is a big range of slow- and fast-changing dynamics is called the *stiffness*. Certain ODE's, especially the implicit ones, are much more fit to solve stiff problems. As the MVDN to be simulated also has some degree of slow and fast changing dynamics, here we can also talk about a simulation with a certain stiffness.
The second condition in which a fixed step solver is a good idea is when the changes in the system throughout the simulation happen more or less at a constant rate. Meaning that there are no long periods in which the variables remain the same followed by short periods of rapid change, e.g. a SG switching states of operation every 15 minutes, where during this 15 minute interval, no changes happen. The reason is rather obvious, as the switching behaviour in the MVDN model is quite difficult to solve and requires a lower step size. The (almost) solid state of the 15 minutes after the switch however doesn't require a small step size. In a fixed-step simulation, the step size of the solver is decided by the part of the simulation that requires the lowest step size. This also means that the faster to solve parts are calculated using the same step size, in the end making the time

required to solve the entire model unnecessarily long.

3.3.2.2 Variable-step solver

In this case, the step size is not constant, meaning the time between each computation depends on the state of the model. If the model variables are rapidly changing, the step size will shrink to ensure a correct representation of the model results. For longer periods without notable changes, the step size will increase again without decreasing the accuracy on the results. For dynamic simulations where rapid fluctuations in the results are followed by longer periods without many changes, this type of solver is a clear cut winner. Same as with the fixed-step solvers, there are implicit solvers available for when the problem has a high degree of stiffness. Along with this, multi-step solvers also exist within the variable-step category. These solvers will look at the results of multiple previous steps instead of just the last one. In certain scenarios this can give more accurate solutions.
It can be noted that certain components inside simulink only work when paired with a certain solver type. This is another big reason why the variable-step type was chosen over the fixed-step. A few of the model components that will follow in section 3.4 require the use of a variable-step solver and so even if the argument for the variable-step type was not convincing enough, this definitely seals the deal.

3.3.2.3 Continuous vs discrete

Both the fixed- and variable-step solvers exist in *continuous* and *discrete* variants [30]. As the name suggests, a continuous simulation will describe the state of the simulations and its changes over time in a continuous way, meaning that the time variables according to which the simulation changes are not discrete. A continuous function, most of the time an ODE, is found that can describe the status of the system at any moment. More often then not, especially in complex systems, an analytic approach to finding the matching differential equation for the system is not possible. Here we will have to resort to numerical computational methods that will approximate what would be the analytic function to describe the continuous behaviour of the system. If the right ODE algorithm is chosen for the task, the results can be very accurate. A common usage for the continuous simulation type is physical systems as these don't have discrete states most of the time and have to be modelled in a continuous way. In the case of this dissertation, where we work with voltage, current and power flows a continuous simulation variant was chosen.
This continuous type of running simulations stands in stark contrast to the discrete type. Only a simulation with discrete events can be solved by this type of solver. This variant is perfect when the model has a sequence of events that are related to time. Every time an event happens within the model, the state of the

system is updated at that time. This variant is used often in economical models related to sales, logistics,.. as units sold, trucks on the road, units in warehouse etc. have a discrete nature.

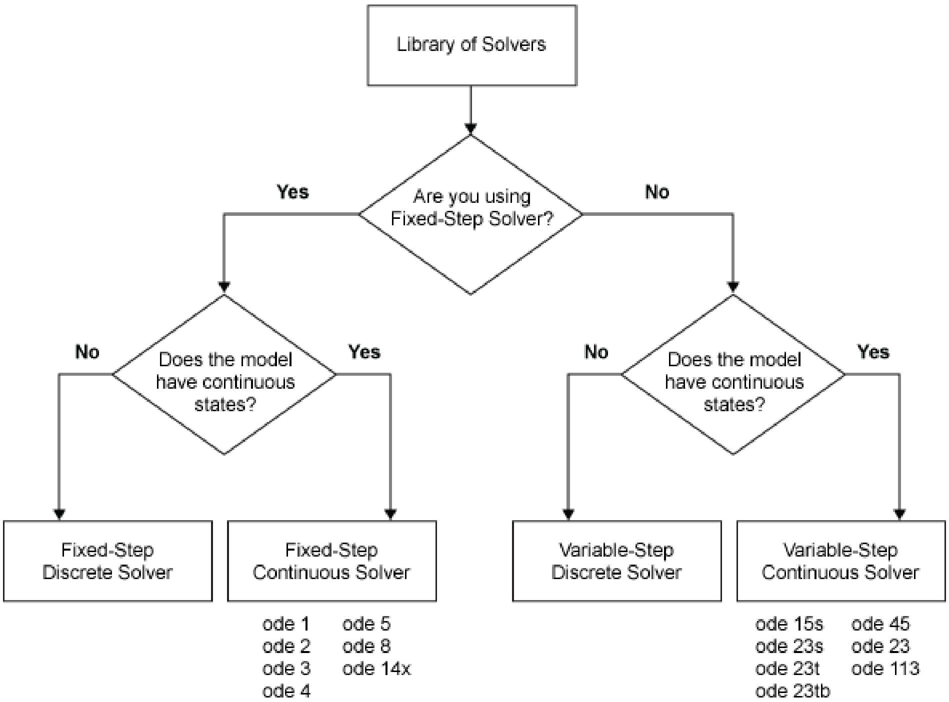


Figure 3.2: Simulink solver flowchart
[27]

3.3.2.4 Selected solver

In the end there are a few solvers that meet the criteria needed to solve the simulation.

- **ODE45(Dormand-Prince)**: This solver is Simulinks jack-of-all-trades variable solver. There are little simulations that this solver cannot compute. Using an explicit Runge-Kutta (4,5) formula (the Dormand-Prince pair) for numerical integration. This is a one-step solver, which means it only needs the solution of the previous time step. It is however not always the fastest or most accurate. In most cases there is a solver that can do a better job in less time. This is why we will look further for a more suitable solver.

- **daessc(Simscape Solver)**: As the name implies, this solver is specifically designed to solve Simscape physical systems.It calculates the state of the model at the next time step by solving systems of differential-algebraic equations resulting from Simscape models. For the MVDN-simulation it did however perform pretty poorly or sometimes not at all. It is still a mystery as to why this exactly is. On paper this one seemed like the best tool for the job, in practise there are a lot of solvers out there that do a better job solving this MVDN smart grid simulation.

- **ODE23tb(stiff/TR-BDF2)**:Now we get to the solver that performed exceptionally well in this MVDN simulation: ODE 23tb. This solver calculates the state of the model at the next time step by making use of a multi-step implementation of the TR-BDF2 algorithm, this is an implicit Runge-Kutta algorithm with trapezoidal rule first stage and the second stage being a backward differentiation algorithm of the second order. For both stages, the same iteration matrix M is used. ODE23tb is very good solving stiff problems. This gives this solver another edge over most of the other solvers. This is why *this solver will be the go-to solver to simulate the MVDN smart grid simulation.*

Below the settings for the ODE23tb solver can be found.

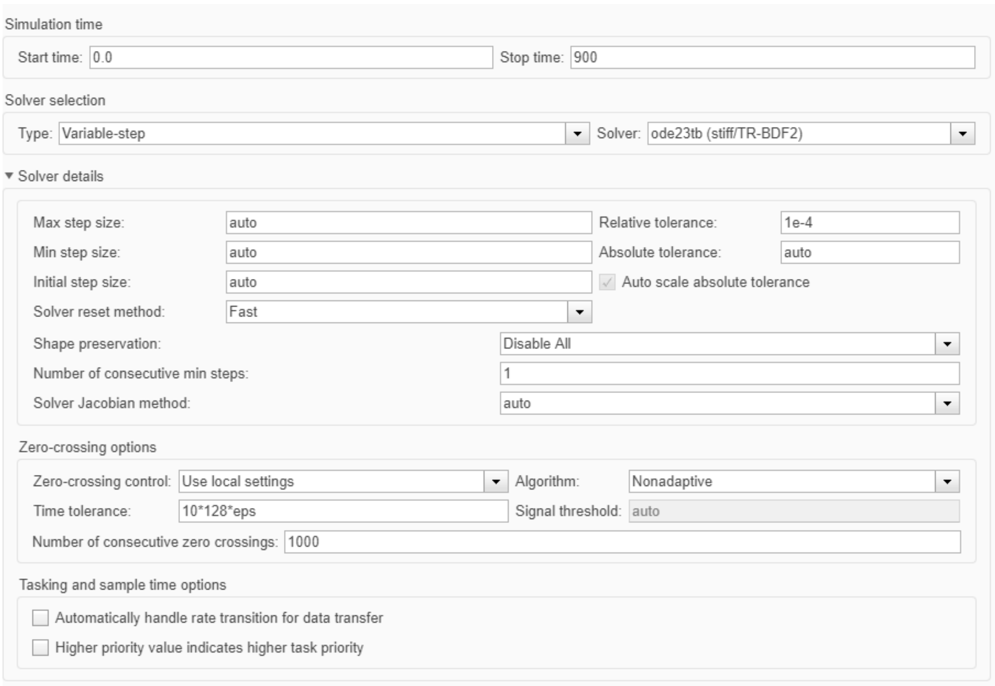


Figure 3.3: Solver settings for ODE23tb

3.3.3 Simscape and powergui

Simscape is a toolbox for Simulink that introduces a lot of components that aid in the development of physical system models within the software. It gives access to extensive libraries containing component models for electrical components such as generators, motors, loads as well as power electronics and more. Alongside the representation of electrical systems, thousands of component models can be found for mechanical, hydraulic and fluid systems, among others. As we are looking at the operation and reconfiguration of a MVDN in this work, the electrical components from the toolbox will primarily be used. At the heart of every Simscape electrical simulation lies the *powergui* block. This block is necessary for the working of all the blocks within this part of the toolbox. The block has a GUI that can display steady-state values of voltages and currents as well as variable states within the network such as inductor currents and capacitor voltages. Initial conditions can be changed in order to powergui gives you the option between 4 types of possible simulations of electrical Simscape systems. The first choice that has to be be made is whether the Simscape parts inside simulation need to be run in a continuous or discrete manner, very similar to the continuous and discrete variants described in 3.3.2. Next the user will have to decide on which type of simulation will be run regarding the representation of voltage and current signals. Here there is the option between the continuous option, that will be calculating the actual wave forms of the currents and voltages. This way harmonics and THD of the network can be studied. As always with more complex representations, the computing times will also be much longer. To keep these to a minimum, the phasor representation can be used. Instead of the actual wave forms of voltage and currents present in the network being calculated, the phasor representation will calculate the voltages and currents by the magnitude and phase angle of the ground frequency component. This simplifies the calculations, but also makes it so that information about the higher harmonics inside these voltages and currents is lost.

3.4 Simulation component models

The full Simulink system making up the MVDN is in essence all the different subsystems of the components in the SG connected to one another. Each subsystem will be quickly introduced and an explanation will be given on how the model for that part of the MVDN came to be. When designing a MVDN in Simulink, a lot of decisions have to be made beforehand. One of them was to model the simulation in p.u. The big advantage here is that the units in the simulation in essence don't have a dimension and are only laid down by the reference units of apparent power and the line voltage. Another advantage is that can not really show it's true colors here as there is only one substation, is that the representation of transformers is greatly simplified using the per unit system. [31]
In the case of this MVDN, there was opted for the following reference units S_{ref} and $V_{ref,LL}$:

$$S_{ref} = 1,000,000VA = 1MVA \tag{3.2}$$

$$V_{ref,LL} = 30,000V = 30kV \tag{3.3}$$

From the reference units for apparent power and line voltage S_{ref} and $V_{ref,LL}$ we can deduce the reference phase current and impedance.

$$I_{ref} = \frac{\frac{S_{ref}}{3}}{\frac{V_{ref,LL}}{\sqrt{3}}} = \frac{\frac{1MVA}{3}}{\frac{30kV}{\sqrt{3}}} = \frac{1MVA}{\sqrt{3} \cdot 30kV} = 19,24A \tag{3.4}$$

$$Z_{ref} = \frac{\left(\frac{V_{ref,LL}}{\sqrt{3}}\right)^2}{\frac{S_{ref}}{3}} = \frac{\left(\frac{30kV}{\sqrt{3}}\right)^2}{\frac{1MVA}{3}} = \frac{30kV^2}{1MVA} = 900\Omega \tag{3.5}$$

3.4.1 Distribution lines

The modeling of a MVDN distribution line in Simulink is rather easy. The current flowing through the line is dependant on the impedance of this line and the impedance of the load following it, and the voltage applied over both. As discussed before, the parasitic capacity to ground is small enough to neglect without effecting the results much. This gives us the following differential equation:

$$v_{in}(t) = R \cdot i(t) + L\frac{di}{dt} + v_{load}(t) \tag{3.6}$$

the following figure gives some more insight into where this differential equation is coming from.

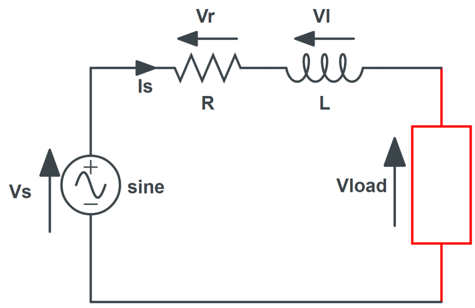


Figure 3.4: Distribution line differential equation

The software environment itself provides a ready to go three-phase RLC branch block, where the R, L and C values can be adjusted so that the resistance, inductance and capacitance of the branch can be exactly modeled after the real power line. As with the rest of the components, the resistance and inductance values are in p.u.

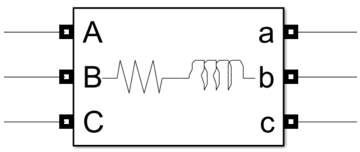


Figure 3.5: Distribution line in Simulink

3.4.2 Circuit breaker

The reconfiguration aspect of a SG is largely dependant on the ability to make and break connections on certain times or whenever necessary to redirect the power flow throughout the network. As we are talking about a MVDN, the currents flowing through the power lines are definitely something that needs to be taken into account when we want to achieve this switching behaviour.
In simulink, inside the Simscape toolbox, there is a *three-phase breaker* block. This block will make or break a connection at certain pre-specified times in the parameters of the breaker. In this block's parameter, it is possible to specify external control. If the box for this control is checked, a new input, called the 'com'-gate appears, you can see this gate in Figure 3.6 If a value of '0' is applied to the gate, the breaker stays open. If the value on the gate however turns into a positive value, the gate will close. It is possible to make the input on the

external control time-dependant. This can be achieved with an array containing timestamps in the first column and values to be applied at those timestamps in the second column. The second way this can be achieved is by using the built in clock and a Simulink look-up table, linked to the same array mentioned above. The added advantage doing it this way, is that if you combine the clock with a start time, you can let the breaker and thus the simulation run from any specific point in time. A setup like this can be found in Figure 3.7.
Both the ability to interrupt high current and the flexibility of switching with respect to time makes the three-phase breaker an interesting contender when it comes to establishing and breaking connections between the different parts of the network.
To ease the transient response when switching occurs, resistive and capacitive snubbers are part of the breaker and can be adjusted in value inside the breaker parameters. The contact resistance of the breaker contacts can also be adjusted here. If you watch Figure 3.6 carefully, you can also see the snubber and contact resistances there.

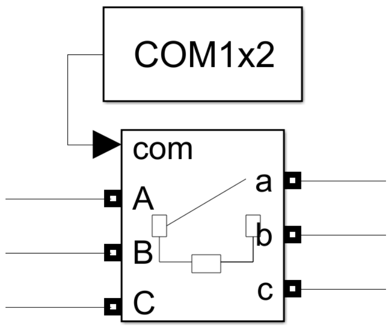


Figure 3.6: 3-Phase circuit breaker in Simulink

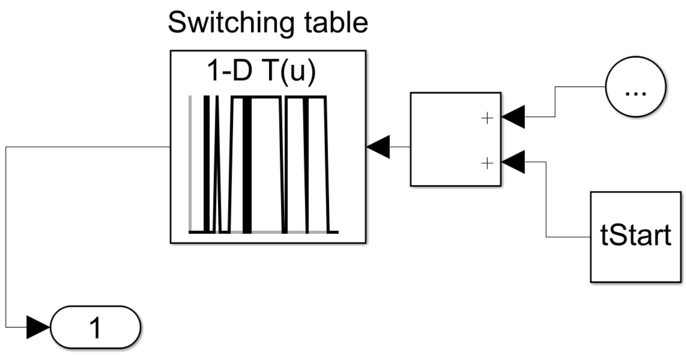


Figure 3.7: Switching setup for 3-phase breaker with respect to time

3.4.3 Substation

The substation establishes the link between the MVDN and the HVDN that feeds it. The lowering of the voltage level is done by 3 separate $10MVA$ YNd transformers. The meaning of the 'YNd11' notation is that the primary coils (high voltage side) are in a star-configuration (capital Y), with an earthed star point (N).The secondary coils are in a delta-configuration(lower case d). The '11' indicates the clock number used. This clock number depends on how the transformer is wired and indicates how much the voltage on the secondary side leads or lags the voltage on the primary. Even though in theory there are 12 possible clock numbers, just like there are 12 hours on an analog clock, it should be noted that not every clock number is used commonly or even at all. A full circle describes 360°, this means that every clock number corresponds with $30 \cdot clocknumber$ difference in phase between primary and secondary windings, with the primary voltage leading the secondary. In a Dy11 configuration for example, the primary voltage will lead the secondary by $11 \cdot 30° = 330°$. More commonly noted as 'lagging by 30°'. In the simulink model, the HV voltage source connected to the primary windings of the transformers has a phase angle of -30° to accommodate for the phase shift caused by the configuration of the windings. This is to make sure that BUS0, the slack bus, has an absolute phase angle of 0°, which makes it easier to see the difference in phase between the slackbus and other busses in the network.
To achieve a total apparent power of 30MVA in the substation, three 10MVA transformers have been installed in parallel. To be able to do this without damaging the transformers over time or inducing circulation currents in the windings, which will reduce the efficiency, a few conditions have to be met:

- **Nominal rated voltage** of the transformers of all the parallel transformers should approximately be the same.
- **Transformation ratio** should be the same for all of the parallel transformers.
- **Clock number** of the transformers should be the same or a compatible one. With the right connection of the transformer terminals, all uneven clock numbers are compatible, the same holds true for all transformers with an even clock number. Uneven and even clock numbers don't mix in a parallel setup however.
- **Short circuit voltage** u_{sc} (in %) of all parallel transformers should approximate each other as closely as possible.
- **the** $\cos\theta_{sc}$ of all parallel transformers should be approximately the same with $\cos\theta_{sc}$ defined as:

$$\cos\theta_{sc} = \frac{P_{sc}}{u_{sc} \cdot S_{nom}} \tag{3.7}$$

Where P_{sc} stands for the short-circuit power measured during the short-circuit test of the transformer. S_{nom} stands for the nominal apparent power of the transformer.

If all these conditions have been met, the transformers will be able to operate at a high efficiency. To make matters simpler and ensure that all conditions are met with certainty, the decision to use 3 identical 10MVA transformers was made.

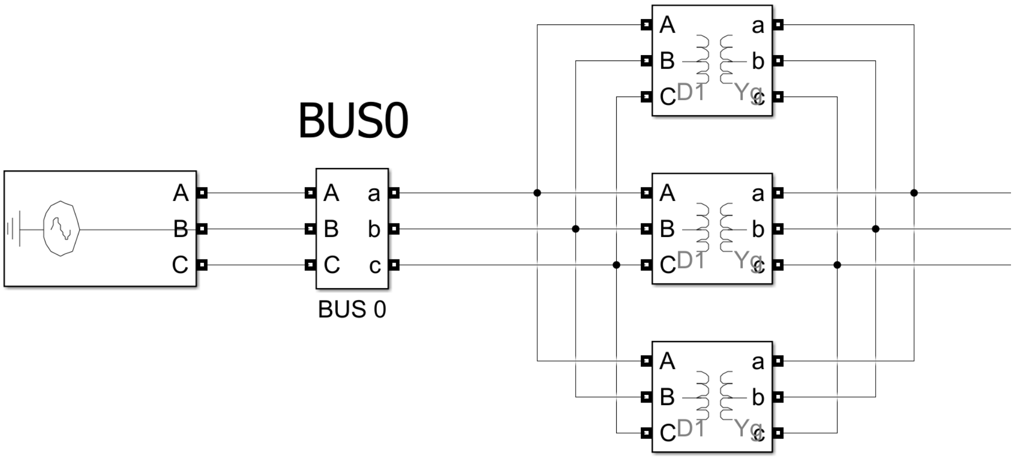


Figure 3.8: Substation in Simulink

An attentive reader might ask himself the question after reading this paragraph about different transformer configurations and clock numbers why the decision to go for a star-delta setup was made. If a simple star-star or delta-delta configuration was chosen, you wouldn't have to deal with the clock numbers, adjustment in winding ratios and phase angle shift to ensure the slackbus has an absolute phase angle of 0°, right? Why both the primary and secondary have this exact configuration has a solid reason.

- **Primary in star with grounded star point:** In theory it would be possible for the primary winding to be connected in a delta setup. The biggest problem with this is that a delta configuration does not have a neutral connection and thus can not be connected to ground. At first this might seem fine, as we have a system that is isolated from ground this way. this is however not a very cheap or effective way to exploit a MVDN. The primary reason why exploiting a MVDN as an isolated system is sub-optimal is because it is quite expensive compared to the commonly used resistance earthed systems. isolated DNs require insulation monitoring devices to detect the first isolation fault. Because the current doesn't have a return path when a first fault occurs, a fault current cannot flow. This fault has to be noticed directly, because in the case of a second fault from another phase, there is a direct phase-phase closed loop through ground, resulting in a very dangerous situation as the ground fault voltage is not the phase voltage but the line voltage, which is $\sqrt{3}$ times larger! The monitoring devices detecting insulation faults are notoriously expensive, especially on such a large scale. The cost of having to hire people around the clock to monitor and intervene when necessary can also not be neglected.

A star configuration without star point connected to ground (or a that star point being connected to ground over a very large impedance) results in the same difficulties. It becomes apparent that earthing the star point is quite advantageous. The question now becomes which kind. At first glance directly earthing the star point to ground seems like a good idea, as it is done the same way in LVDN. We are however talking about way higher voltages and consequently currents. Directly earthing a medium voltage star point will result in very high fault currents whenever a first fault occurs. This high current can be very destructive towards circuits in close proximity due to induced currents. To limit the time of the fault, a differential protection device will need to be installed, which are costly. Now that we know that directly earthing has some considerable downsides, we move on to the last and probably the best option: Resistance earthing. Here, the star point is also connected to ground, but with a small resistance added to the connection. This will make sure that the fault current in case of an unwanted connection to earth are rather limited and don't cause damage to equipment, distribution boards and neighboring electrical circuitry. In the case of unbalanced loads, the zero-shift of the electrical neutral point of the lines is minimal. This also might explain why in practice, the star point of a transformer is almost always connected over a resistance to ground.

- **Secondary in delta**: To understand why the secondary windings are connected in the delta configuration, we have to take a quick trip to *symmetrical component* theory. Each voltage, current and related to both, the impedance, can be split up into a positive-sequence, *negative-sequence* and *zero-sequence component*. In an ideal scenario, only the positive-sequence component plays a role in the voltage and current. Assuming the voltages feeding the network are symmetrical and direct, the current will also be, in the case where the network behaves itself as a symmetrical load (meaning the loads have the same impedance and there are no short-circuits of any kind). The reason we don't want any negative-sequence and zero-sequence components is because they lead to efficiency loss and possible damaging of the network components after exposure to these unbalanced currents for too long. negative-sequence currents are especially bad for synchronous generators. The term synchronous comes from the fact that the rotor and magnetic field rotate at the same speed. If the generator's load is unbalanced though, a second negative-sequence current component comes in that produces a magnetic field with the same speed but opposite rotational direction to the rotor. From the viewpoint of the rotor, it appears as if there are two fields rotating, causing currents at double the magnetic field frequency to be induced into the rotor. These currents will heat up the rotor and cause damage to the rotor or insulation of the rotor windings. Unfortunately setting up the secondary part of every transformer in a delta configuration does not help with this problem, as these negative-sequence currents are passed on from primary to secondary without a problem. This is why it is important to keep the load as symmetrical as possible at all times.

when it comes to the zero-sequence component, the delta configuration of the secondary windings can completely negate this component. The three zero-sequence current phasors all have the same phase angle. This means that they will have to find a path to flow away. In the case of a delta connection, the zero-sequence component is 'trapped' inside the delta configuration so there is no zero-sequence component passed on to the lines. This means the triplen harmonics (order = 3,9,15,21,...) associated with this component are not transferred from the primary to secondary side of the transformer, improving the overall energy distribution efficiency.

Another advantage of the delta connection being on the secondary side of an MVDN substation is that the current is highest on this secondary side. Since the current inside the delta windings $I_f = \frac{I_{LL}}{\sqrt{3}}$, the cross section will have to be less than that of an equivalent star-configuration, making them cheaper when it comes to electrical conducting material.

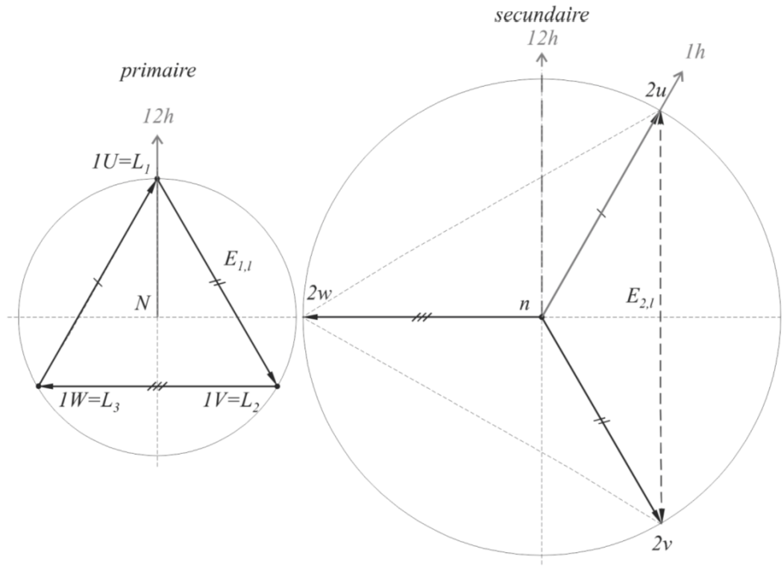


Figure 3.9: Dyn1 vector diagram
[22]

3.4.4 Busbars

Busbars or *Busses* for short are metal bars made out of electrical conductive materials like copper. Their basic function is to distribute upstream current, they are most often used in panel boards and switch gear, accompanied by measuring- and safety-devices to monitor e.g. current, harmonics, voltage quality etc. along with safeties to interrupt the current if it exceeds predetermined parameters. Inside Simulink itself, a busbar does not really exist as a block. There is however a block called the *three-phase VI measurement* This will substitute as a busbar and measuring device, combined into one. It should be noted that a busbar also has an internal impedance. Due to the fact that the cross section of the bar is relatively big, this impedance will be neglected in the Simulink simulation.

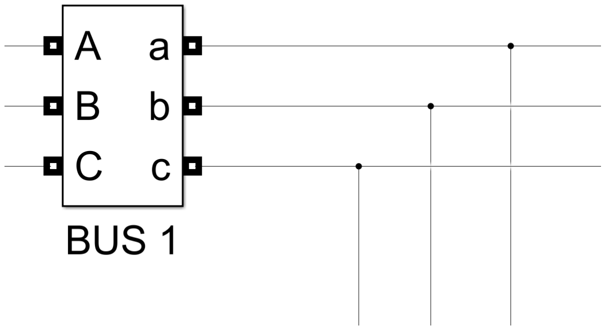


Figure 3.10: Busbar in Simulink

Measurements of the RMS-value of line voltage V_{LL}, line current I_{LL}, three-phase active and reactive power P_{3ph} and Q_{3ph} are collected using the VI-measurement block and some Simulink Math Operation blocks. This can be found in Figure 3.11. As with all MVDN networks, there should be a proper protection in case of faults to protect persons or property from being harmed. Protection of an MVDN or any other distribution network for that matter is a very expansive topic in and of itself and will therefore not be discussed here as it is beyond the scope of this work.

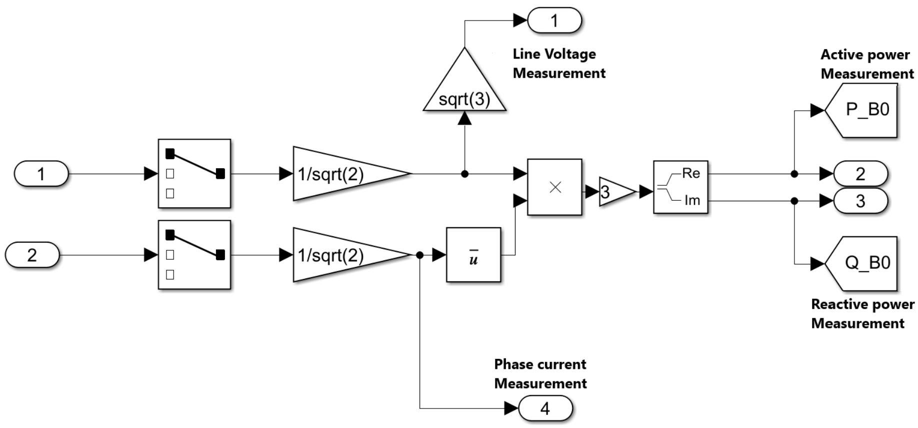


Figure 3.11: Voltage, current, active & reactive power measurement

3.4.5 Dynamic loads and generation

One of the biggest reasons SGs will probably will most likely play a big role in the energy distribution of the future is because of the trend that is decentralized production and the growing popularity of EVs. As such, *dynamic loads and generation* had to be implemented in the Simulink model. Implementing these components and making them work together with the other components was quite a challenge. Most of the blocks ready to be used in Simulink had one or more issues which didn't make them compatible with the breakers and substation. The first apparent solution to implementing dynamic loads and generation was the use of *variable resistors*. This was back when the first versions of the simulation where still ran in a discrete manner. Due to the long time span of the total simulation, and long simulation times paired with a discrete solver, it was not possible to keep the simulation of this type. Variable resistors, which could also adapt a negative value and thus act as a load and a generator, can only be used in discrete simulations. This meant a new solution had to be found. In the end,three-phase dynamic loads were implemented to act as load and generation.

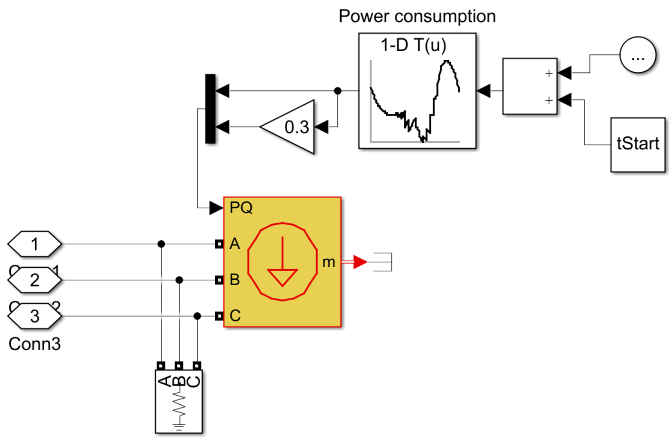


Figure 3.12: Dynamic load in Simulink

Just like with the three-phase circuit breakers, a clock and start time is used to be able to start the simulation at any time. This total time $t_{tot} = t_{clock} + t_{start}$ is then passed on through a lookup table that finds the value of active and reactive energy for that load or generation at that specific time. This is then outputted to the input of the three-phase dynamic load, which is connected to the grid. These Dynamic Loads will then output the specified power at the voltage level of the grid they are connected to. This means that they have to be in connection with a voltage source that determines the voltage level in the grid. This makes them unsuitable for islanding mode. The dynamic loads and generation inside the simulink model can be seen in Figures 3.12 and 3.13.

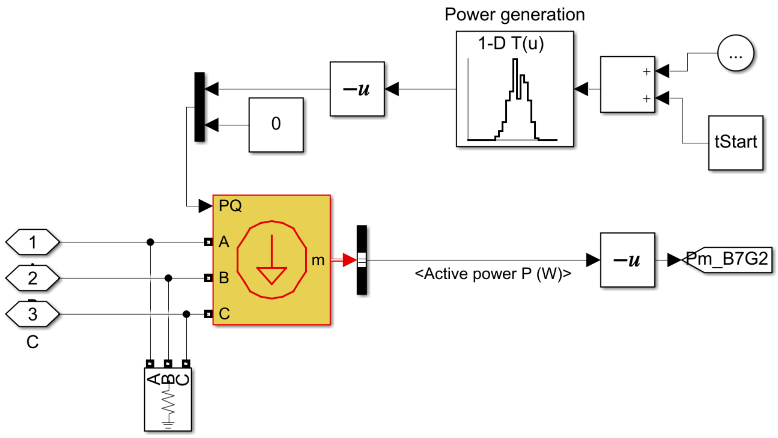


Figure 3.13: Dynamic generation in Simulink

The charging and discharging of electric vehicles also falls under the category of loads and generation as it is actually a combination of both. The three-phase dynamic load block in Simulink supports switching between positive and negative signals and so can be used as such. Depending on how steep and fast the changes occur, Simulink can experience some problems with the simulation of power flows that change in direction according to time. This is why the first implementation of the EV charge-discharge subsystem contained two dynamic load blocks that each had a saturation block attached to the input. A saturation block lets you limit to a certain upper and lower level. The first load block was then implemented as an EV charging station, with the saturation block only allowing positive values into the dynamic load block. The second one was configured as the EV discharge subsystem, with the saturation block only allowing negative values to pass through. In theory this seems like a pretty good idea, but again, due to the sometimes fast and steep changes of the charge and discharge power values, the simulation had great difficulty actually solving this part of the simulation to the point where it was a lot worse than just using a single three-phase dynamic load that takes care of both positive and negative power flows.

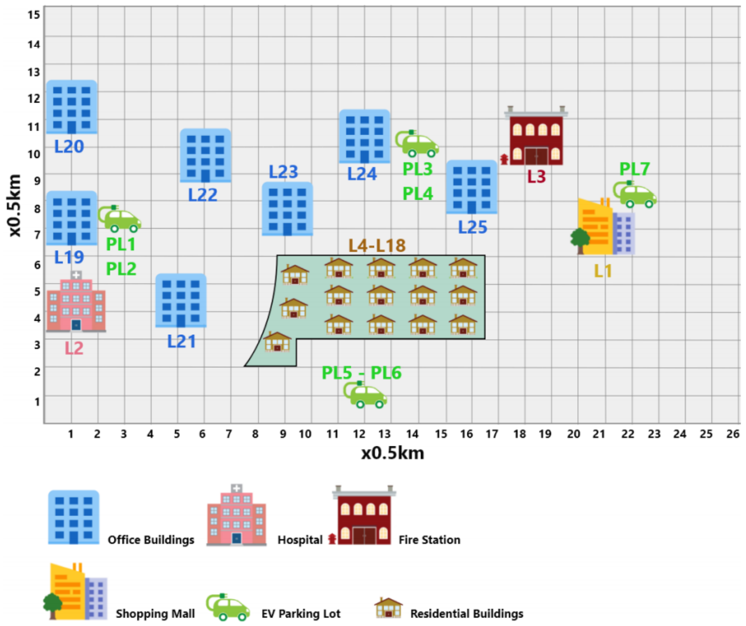


Figure 3.14: Map of the loads of the simulated smart city [9]

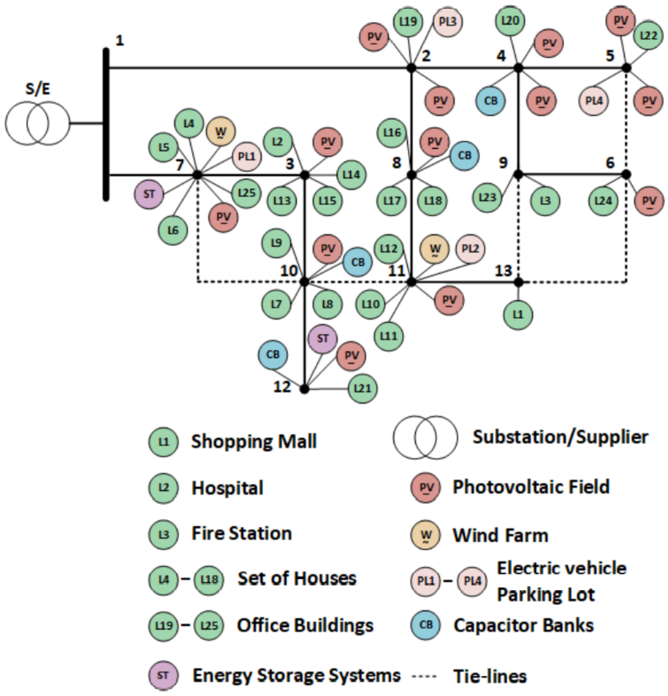


Figure 3.15: Single-wire diagram of the smart city [9]

3.4.6 Capacitor banks

The implementation of capacitor banks can be done quite similarly to how the other loads have been implemented. Since the reactive load of the capacitor banks is constant throughout the whole simulation, a constant capacitor connected to ground was used. If it was necessary for the capacitor banks to change the nominal reactive energy, it is possible to use automatic switch banks. These are capacitor banks that have switches inside to be able to regulate the internal capacity in intervals and in turn regulate the consumed reactive energy. A measuring device inside measures the $\cos\phi$ and switches capacitors in and out to correct it as close as possible to 1.
In a Simulink simulation, this can also be easily achieved. The three-phase dynamic load can be configured as a purely reactive load to act as a dynamic capacitor bank. Since the value for the banks stayed the same throughout the whole simulation, there was opted to use a constant capacitive load to model the capacitor banks. The block in Simulink can be found below in Figure 3.16.

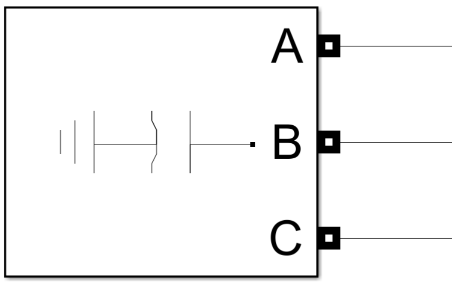


Figure 3.16: Capacitor bank in Simulink

3.4.7 Simulation setup script

In order to feed all the switching-, loads- and generation-data to the switches, loads and generation respectively, some MATLAB .m scripts had to be written in order to communicate the imported data from the MATLAB environment straight into Simulink.

To start off, the different data has to be imported in MATLAB first. This can be done manually, but a more productive way to do this is of course to write some code that does this automatically. In Figure 3.17 you can see an example for loading the active power data file into MATLAB.

```
folder = 'C:\Users\alexa\OneDrive\Documents\School\ISEP\thesis\data\Matlab';
fullMatFileName2 = fullfile(folder,  'PLoad.mat');
if ~exist(fullMatFileName2, 'file')
  message = sprintf('%s does not exist', fullMatFileName2);
  uiwait(warndlg(message));
else
  PLoadFile = load(fullMatFileName2);
end
```

Figure 3.17: File loading script for active power data

Once all the Switching, Loads and Generation data (SLG data) has been imported, we can start manipulating it so that is in the right format for implementation in Simulink. As seen in subsection 3.3.2, the ODEs that are solved are all time dependent, this time-dependency is also reflected in the form of the SLG data. The first column of the data contains timestamps. By default, they increase by increments of 1. The simulation in this dissertation however contains data points every 15 minutes. Meaning that the SLG data will change once every $15 \cdot 60$ or 900 seconds. The timestamps will consequently increase by this increment until the final time value has been reached. In the initialisation it is possible to change and manipulate a ton of variables related to the simulation. Most importantly the amount of periods or times the SLGs change. How long each period lasts can also be adjusted. In theory, with just these two variables it should be possible to run the simulation for any amount of time with any amount of time between periods. The other notable variables used in the simulation will be listed in table 3.2 below.

Table 3.2: Simulation variables, their default value and explanation

Variable	**Default value**	**Unit**	**Explanation**
startPeriod	1	/	Period in which sim starts
tSize	96	/	amount of periods in the simulation
switchPeriod	900	s	total amount of time between two switches
deci	100	/	factor of decimation of the output values
R_{on}	1e-6	Ω	breaker contact resistance in p.u.
R_s	1000	Ω	breaker snubber resistance in p.u.
C_{snub}	3.54e-7	F	breaker snubber capacitance in p.u.

An example of what the SLG data looks like can be found in the appendix.

```
%Declaring simulation variables

tSize = 96; %amount of timestamps in simulation
deci = 100; %decimation for output in workspace and .mat files
switchPeriod = 15*60;%total amount of time between two changes (timestamps) in the simulati

%initialise variables and COM-matrices to control the breakers

tStart    =    0;
t         =    switchPeriod*(0:tSize); %vector that displays every time interval of the tot
Pev       =    [(0:tSize)*switchPeriod;rand(1,tSize+1)-(rand(1,tSize+1))]; %create a matrix
Vin       =    rand(1,tSize+1); %random values between 0-1 for a vector with length 1 to tS
R         =    1e4;
Ron       =    1e-6; %contact resistance for the three-phase breakers
Rs        =    1; %snubber resistance for the three-phase breakers
Csnub     =    3.54e-4; %snubber capacitance for the three-phase breakers; Cs = 1/(Z*2*Pi*f
Ccaps     =    0.1;
ssSwitch0 =    [((1:tSize+1)-0.001)*switchPeriod;zeros(1,tSize+1)];
ssSwitch1 =    [((1:tSize+1)+0.001)*switchPeriod;ones(1,tSize+1)];
ssSwitch  =    sortrows((horzcat(ssSwitch0,ssSwitch1).'),1);
```

Figure 3.18: Initialisation variables of the simulation

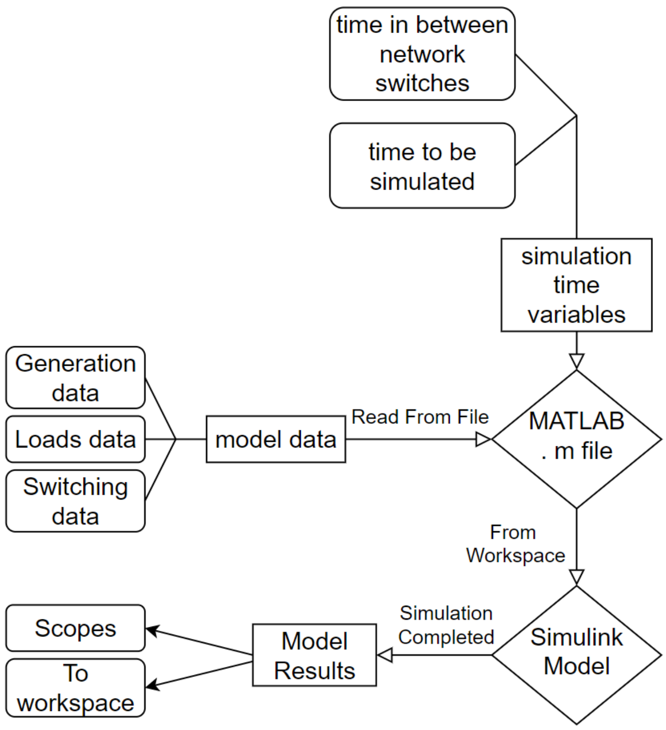


Figure 3.19: Flowchart of the simulation

3.4.8 Conclusions

In this chapter we took a deeper dive into the software, solvers and the process of solving a simulation, the model as a whole and it's different components, the

setup and collection of data using a MATLAB script.
Given the large amount of options between simulation software environments, MATLAB Simulink was chosen over its competitors because of its ready-to-go extensive library of simulation blocks & toolboxes and the detailed help function & support. When it came down to the solvers, there was opted for a variable-step solver and a phasor type continuous simulation. This was the perfect trade-off between simulation speed and accuracy, especially since harmonics are not studied here. ODE23tb in particular came out as the perfect solver for the job because of its fast speed while maintaining a relatively high accuracy at the solver settings implemented (<0.5% relative error margin). The process of how a simulation is exactly solved was thoroughly explained. After this the model of every part of the simulation was shown and explained, along with how all of the components work together to simulate the behaviour of an entire MVDN smart grid. To finalise, the MATLAB .m script used to initialise and collect the data from the simulation was presented with the most important variables being explained and the working of the script briefly mentioned.

Chapter 4

Results of the MVDN Simulation

The fourth chapter of this work will be about the results of the MVDN smart grid simulation in Simulink. Before we get to the results, a section will be dedicated to the encountered problems during this process and the solutions that allowed to continue generating data from the network. After this, the results will be shown and commented on.

4.1 Development issues

The construction of the network was generally speaking pretty straight-forward and didn't involve many problems. Most of the models consisted of simple blocks that were easily implemented and connected with each other. All the blocks were compatible and no conversion blocks where needed inside the simulation since the VI-measurement device took care of the conversion from Simscape signal to Simulink signal. This is necessary because the scopes and To Workspace blocks can not interpret the Simscape signals directly. Even though everything went quite smooth, there was a certain problem with one network component that kept popping up during the construction of the network. This problem was related to the three-phase breakers used to perform switches in the network.

4.1.1 The Breaker Problem

Along the way, a switch from a fixed-step solver to a variable-step solver was made because of simulation time concerns and because the dynamic loads used to simulate the variable loads and generation are more compatible with this type of simulation.
This is when the first problems started to occur. More specifically with one block

in combination with the others: the three-phase breaker. Even outside a simulation environment, interrupting currents is quite a complex thing to do. Opening a switch that is conducting current produces an arc at higher voltages, extinguishing the arc is necessary to completely stop the flow of electricity. After the arc is extinguished, one should take precautions to ensure the arc does not reignite. Transient re-emerging voltages can be much greater than the nominal voltage, causing damage.
In mathematical terms, the differential equation that is linked to interrupting currents is quite complex as well. Combining this with the fact that there are many breakers, loads and generation switching all at once, this simulation proved quite difficult to be solved by a Simulink ODE.
The typical behaviour that could be seen when the ODE was struggling to solve the differential equation of the simulation during a certain period was characterised by:

- The simulation started to slow down. ODE23tb is known for its very high simulation speed. In a period where no breaker switches happened, the simulation of this period took less than 1 second. In periods where the ODE struggled, the simulation was run for over 14 hours and during all this time, three quarters of a period was simulated. if we extrapolate this with all the periods where the ODE struggled, we would end up with 560 hours or more than 23 days worth of simulation just to simulate the behaviour of the network for one day.

- Very strange results regarding the results of voltage, current and power measurement during these periods. In figure 4.1 you can see the active power measurement of an older version of the simulation behaving very strange at certain times. It almost looks like there is resonance going on. This was however impossible at the time since the lines in the simulation had no capacitive component and there were no capacitive loads connected to the grid either. Which means that $C = 0$ and thus the limit value of the resonance frequency f_0 goes to infinity.

$$\lim_{C \to 0} f_0 = \lim_{C \to 0} \frac{1}{2\pi \cdot L \cdot C} \tag{4.1}$$

Now that we know that it is not resonance playing a part in this behaviour, the next plausible idea is that it is just the ODE having a hard time solving the differential equation once a switch happens in the network.

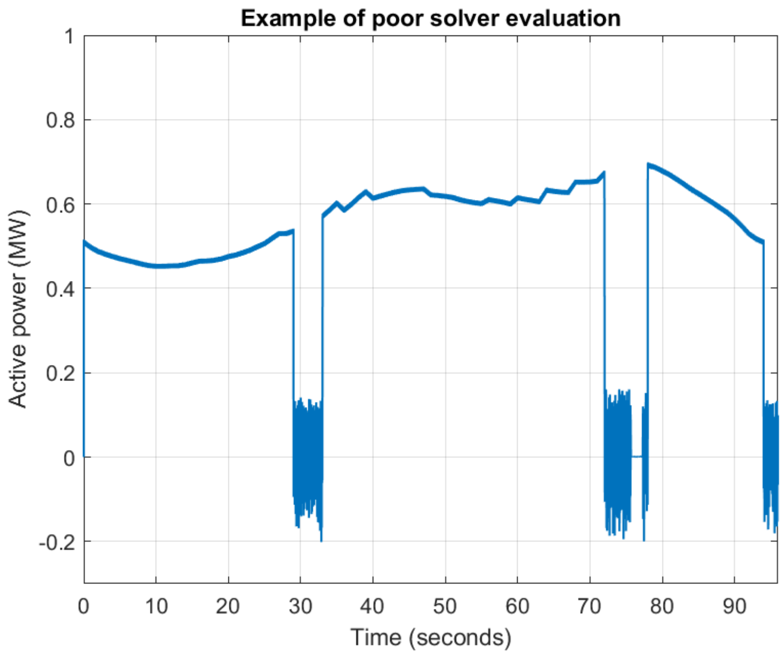


Figure 4.1: Abnormal solver behaviour

The simulation was run manually period per period to check which periods were evaluated poorly. The results can be found in the following table:

Table 4.1: Poor performance simulation periods

Periods with poor evaluation	Periods with crashes
2, 4, 10, 12, 16, 18, 26, 27, 30, 31, 32, 31, 32, 33, 35, 36, 37, 39, 60, 63, 75, 76, 77, 79, 84, 85, 86, 87, 89	34, 74, 78, 96

If we take a look at the switching matrices, it becomes apparent that the periods that show wildly inaccurate results coincide with the periods where a switch of one of the three-phase breakers happens. This confirms the suspicion that the breakers were the ones causing this behaviour in the first place. Trying to find a solution for this problem and implementing it into the simulation is what took up most of the time spent building this simulation.

4.1.2 Possible causes

Fiddling with settings and the simulation itself, trying out new solutions in testing environments and trying to implement them in the main simulation gave quite some insight in what is important to make the simulation works, and if it doesn't work, what might be the cause. This insight was combined in the next subsection where we will look at the possible causes for this problem.

4.1.2.1 Scalability issue

. In the examples found that were built into Simulink containing the use of this breaker, there was only ever one implemented in the models. The simulation contains 16 of them. During some periods, multiple of them switch and so it's harder for the ODE to figure out what is actually going on in the simulation.
Most likely this is not the cause of the Breaker Problem however. In some testing scenario's that are covered later, the problem occurs, even with a single breaker present.

4.1.2.2 Solver settings

The impact of which solver you choose is not to be underestimated. Chapter 3 went deeper into all the different possible solvers and their uses. Trial and error, combined with the MathWorks website gave quite a bit of insight into these solvers and which ones to use where. Using the wrong solver, and especially the wrong type of solver (fixed-step, variable-step, explicit, implicit, stiff, non-stiff etc.) could result in inaccurate results or even premature termination of the simulation. *Through vigorous testing, the used solver (ODE23tb) turned out to be almost certainly the best one for the problem and the solver settings were also nailed down pretty solidly. The problems we were facing most likely don't have anything to do with the solver settings.*

4.1.2.3 The combination of breakers and dynamic loads

Quite early on in the building of the simulation there was a test setup where this same problem occurred and it resulted in a premature termination. Back then, nothing was thought of it as crashes happened very frequently. The model was not yet on point, things were being tested and the solver settings where not yet nailed down. Only later on, the realisation came that the reason the simulation crashed was because of the three-phase breaker in combination with the dynamic loads. The reason this setup is giving trouble probably has to do with the way breaker blocks are programmed combined with the fact that dynamic loads need a source that supplies them with information on the voltage and frequency at which they operate. Meaning they can never be isolated. This is where the breaker comes in. If the path of the power flow changes within the network, probably due to how the breakers or even Simulink itself is programmed, the dynamic loads become isolated for a split-second. This means the dynamic loads don't receive the information about voltage level and frequency and the simulation slows down. The produced network results also show strange behaviour such as in Fig 4.1. *As this behaviour was confirmed in a test scenario with as little outside factors as possible, it seems very likely that this is the cause of the Breaker Problem.*

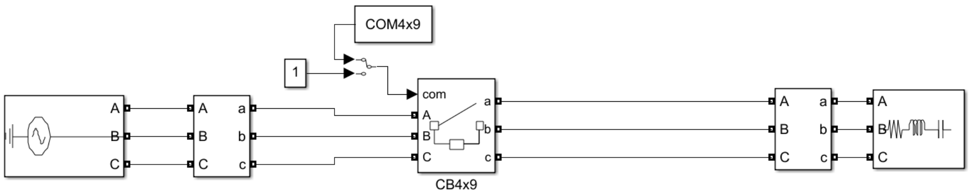


Figure 4.2: Breaker Problem test setup

4.1.3 Possible solutions

Now that the problem regarding the three-phase breakers is kind of known, we will have to find a implementable and working solution. Below, some of the methods that were tried and how effective they were at solving the simulation are summed up.

4.1.3.1 Changing the breakers with variable resistors

Since the switching behaviour of the three-phase breakers is causing problems to the solving of the entire simulation, the first method tried involves swapping this component with another one that behaves in a similar way and can accomplish the same function inside the simulation. A variable resistor could work in theory. The idea is that a breaker is basically a resistor with a very low resistance value when it is closed and one with a very high resistance value when it is open. The only loss of accuracy comes down to the capacitive snubber not being there anymore. This can easily be solved by adding a capacitor in parallel over the variable resistor.
The variable resistor would then be implemented almost the same way as a breaker that can be manipulated with a control signal. A signal applied to the control gate of the variable resistor could change it from a very low to a very high resistance value and vice versa, mimicking the switching behaviour of a breaker in time.
This approach did not work however. The first reason is that the Variable Resistor block can only be used in a fixed-step simulation. As discussed before, this is simply not possible due to simulation time constraints and compatibility with

the other blocks in the simulation. Even if it was possible to implement the variable resistor in the first place, it is not certain that this would solve the Breaker Problem in it's entirety. As the change in resistance value would also happen discretely, this might cause the same faulty behaviour as the three-phase breaker

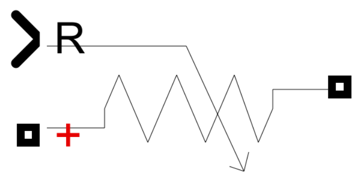


Figure 4.3: Variable resistor in Simulink

4.1.3.2 Switching off the main power during the breaker reconfiguration

The idea for this possible solution is that since the main power source, the substation, dictates and adjusts the nominal voltage and frequency for the dynamic loads and generations in the network. If we could somehow temporary isolate the whole network, make the breakers switch and take the network out of isolation, we could fix the problem. A MATLAB script was written to accomplish this. *It became apparent very fast though that this was a bad idea. To isolate the network, the substation would have to be switched off or disconnected, both of which would facilitate the same Breaker Problem that we had in the first place. This method is decisively crossed off the list.*

4.1.3.3 Composite block simulation

For the three-phase Simscape blocks, there are two different kinds. One where all three phases are modeled separately and one where the three phases are all condensed into one line, the so called *composite* three-phase Simscape blocks. If the breaker from one type is causing problems, it might be a valid approach to try and make the simulation work in the other type. Both are three-phase blocks but other than that, the two variants differ quite a bit. To begin with, the composite blocks don't work with the powergui block. They operate using a standalone system called the 'solver configuration'. This block is used to indicate the zero reference in the network. The parameters of the block also have to be configured properly to make the simulation run. This proved to be quite difficult as the simulink help pages didn't specify much about what each parameter exactly meant or which values should be filled in. Even running a simple simulation with a constant generator, distribution line and constant load resulted in an immediate termination of the simulation. *The composite blocks felt quite obsolete and the non-composite blocks were just more advanced and easier to work with* *in general. Besides this, the physical signals inside the composite blocks had to be converted to simulink signals using a physical-simulink converter, vice versa when you wanted to introduce a simulink signal as an input in the composite blocks. These converters gave quite a bit of errors and problems too. Overall this seemed like a sub-optimal solution. Other, better solutions can probably be found.*

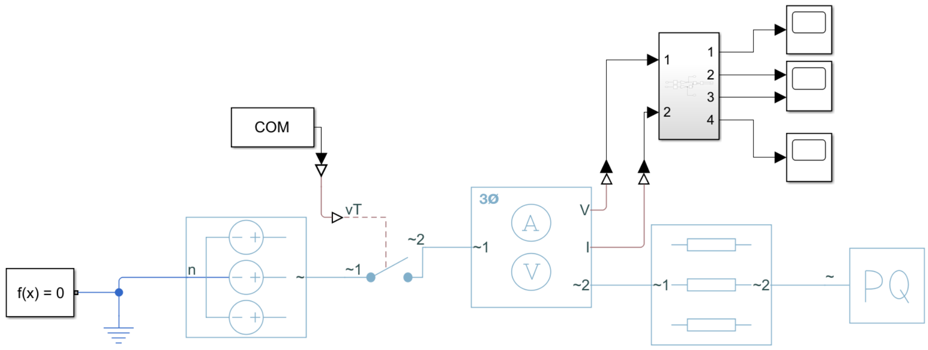


Figure 4.4: Three-phase composite blocks in simulink

4.1.3.4 Change detecting MATLAB script

For the next method we definitely enter the realm of MATLAB simulink limitations. The idea behind this method is that the strange simulation behaviour occurs in a period where one of the breakers just switched to another state. *Even if you run the simulation after the breaker already performed the switch in the previous period*, the strange behaviour occurs. If we could find a way to implement a "buffer period" after every switch, somewhere in the simulation, it could theoretically be possible to get this data from the switching matrices and run the simulation period by period without encountering the above mentioned problem. After quite some work on a MATLAB script that performed this feature, it turned into a buggy coding nightmare. This was an ambitious effort but there were a few unforeseen difficulties popping up when attempts were made to implement this method:

- Whenever a switch were to happen, somewhere in the network, a new timestamp would have to be created in all of the switching arrays, otherwise the switching arrays for the different breakers would not have the same amount of timestamps after a while.

- If a timestamp is created somewhere, the timestamps following this one would have to move up. This makes it so the actual time is not correct anymore, which would require a table to be made with the exact simulation times corresponding to times within the newly created "buffered" arrays.

This was pretty hard to implement as is, let alone MATLAB and simulink taking it in to produce usable results. This method was shelved after efforts to fix the unforeseen problems where not very successful.

4.1.3.5 Commenting breakers in/out using a MATLAB script

Just like it is possible to comment out code in MATLAB, it is possible to comment out blocks or parts of a simulation in Simulink. This brought up some ideas about implementing the connecting and disconnecting of lines inside the grid by commenting the breakers that separate them in and out at the appropriate times. This is quite an intensive and hard to implement method, especially to do this automatically. The first thing necessary was to comment breakers in and out from a MATLAB .m script. There is a function inside the software that can do this. This function has to take in 3 arguments:

1. The block handle of the block that is to be targeted. This is so MATLAB knows which block has to commented in or out.
2. The string argument *'commented'*, giving notice to MATLAB that this targeted block needs to be commented in or out.
3. The argument *'on'* or *'off'* to indicate that the commenting out of the block should be on or off.

The first Simulink block handle for the first argument can easily be achieved with a single line of code:

```
CB1x2 = getSimulinkBlockHandle('thesis/CB1x2');
```

The connection between Bus 1 and Bus 2 was just randomly picked, analogous code applies for all the breakers. The second argument is self-explanatory. The third argument however requires some more code. As we want the commenting to be done automatically we have to grab the data of when to comment in or out which breaker from somewhere. Obviously we can get this data from the switching matrices, or even easier: from the switching array of each individual breaker that is derived from the switching matrices.
A table is implemented with both options 'on' and 'off'. With a single line of code, a 0 or 1 in the switching array is then linked to an 'on' or 'off' respectively in the newly created commenting array. The code for this can be found below:

```
onOffTable = {'on', 'off'};
p1x2 = onOffTable((M1x2(2,:)) +1);
```

This timestamped commenting array is now ready to give timestamped instructions on when to comment every breaker in or out. These commenting arrays are then used inside a loop to comment the breakers in and out at the appropriate times. Every period of the simulation, the parameter update is then sent to Simulink, after which the simulation is called upon from the MATLAB script to run with the newly applied parameters using the following code:

```
for i = 1:(tSize+1-startPeriod+1)
     set_param(CB1x2,'Commented',string(p1x2(1,i)));

     set_param('thesis','SimulationCommand','Update')

     simOut = sim('thesis');
end
```

The data from every period of the simulation can then be saved in the MATLAB workspace, where it can then be concatenated to achieve all the data for the full simulation. This way the voltage, current, active-, reactive- and apparent power in every Bus, Load and Generation can be studied. It took some workarounds but this is how the Breaker Problem was finally fixed.

4.2 Results

Now that the simulation data can be generated from the model, we will take a look at results. The simulation itself collects the line voltage and current, active, reactive and apparent power in every bus, load and generation. BUS0 and 1 collect the data for the primary and secondary side of the substation respectively. The other busses are spread around the network, always accompanied by a breaker switching the network. The whole simulation was run for a time span of 24 hours. The time axis is in seconds, with 86400 seconds spanning one entire day.

Due to the way the data was collected, concatenating the data from the separate simulation of every period together, something odd is going on when you look at the data. Every simulation starts with a spike in value for a split second. This most likely has something to do with the initial condition of the differential equation computed being way off the steady-state value. As the spike only persists for a few fractions of a second, it doesn't spoil the data. It does however produce a noticeable pattern in all the figures.

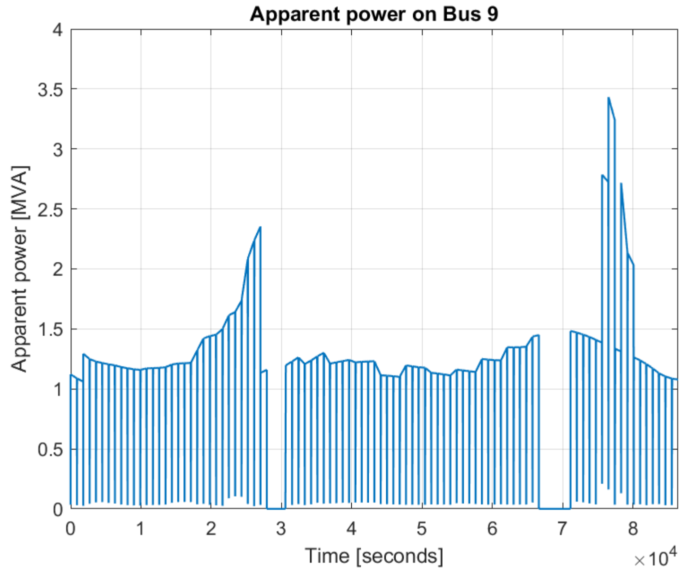


Figure 4.5: Example of spikes in figure

Because of the way MATLAB stored the data (in what is called a MATLAB Timeseries object), combined with the fact that the simulation was ran as a variable type, meaning the timestamps between data points don't have a fixed time interval, it proved surprisingly difficult trying to remove these spikes. MATLAB provides a method to remove samples from a timeseries dataset. It does however require the index of these samples. As the index doesn't match exactly with the multiple of a timestamp, it's hard to remove them this way.
Taking the moving average of the data could filter out the spikes. This doesn't work here either because a variable solver decreases the time in between samples as the rate at which changes in value happen increases. This means that there are a lot of samples taken when the spike happens over a very short time. Taking the moving average here would not result in the spikes disappearing. The spikes will be visible in the figures but they don't impair the reader from interpreting them.

4.2.1 Power flows inside the MVDN

We will take a look at the power flow inside the energy system now.
Below you can find the total active power generated by the renewable generation. Since there is no reactive power generated by this source, the reactive power will be zero and the apparent power generated by this generation will be equal to the active power.
The first thing that immediately catches the eye is the irregularity of solar and wind power production. During the night there is no solar irradiation and even

the wind turbines don't inject any active (or reactive for that matter) power into the grid. This last one has merely to do with the data supplied. Wind turbines generally produce even more power at night, the night of data collection, the wind speeds were too low for the turbines to operate ($v_{wind} < v_{cut-in}$) and inject power into the grid. The model was tested with much higher injection of power into the grid by the wind turbines (up to 3MW by the turbines alone) and was able to handle this without a problem.
At around 42,000 seconds into the simulation or at 11h40, a peak in generation is reached of around 3.6 MW. If we look at the dynamic loads in Figure 4.9, the renewable generation is able to supply about half of the active power demand at that time.

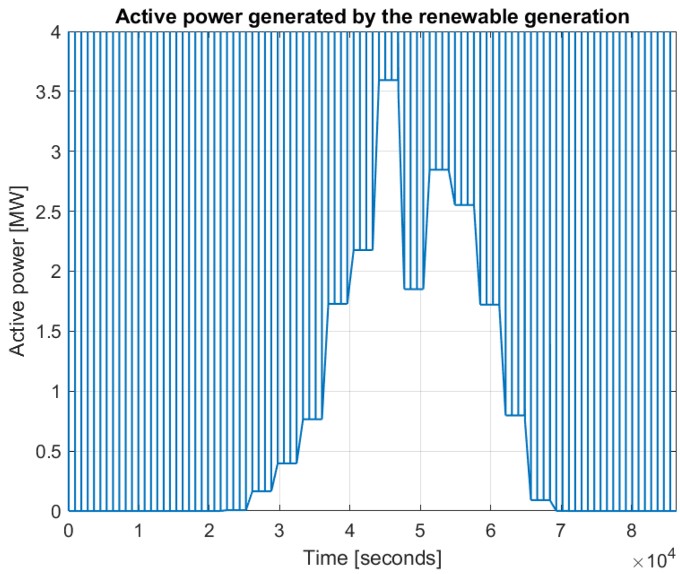


Figure 4.6: Active power generated by renewable generation

As the renewable generation can not provide all of the active power demanded from the loads, the substation will have to fill in. The active power supply from the substation can be found in Figure 4.7. Here you can clearly see the trough in active power supplied around the same time peaks occur in active power generation by the renewables.

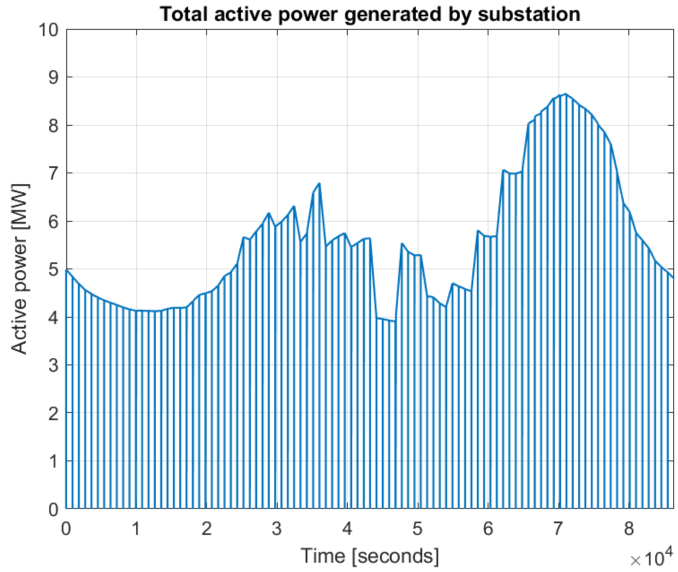


Figure 4.7: Active power generated by the substation

By adding up the active power from the renewable generation and the substation together, the total active power generation of the network is found.

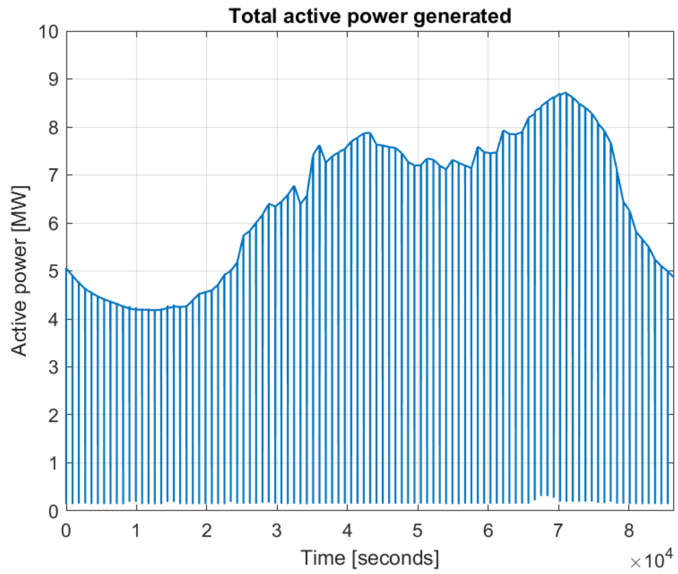


Figure 4.8: Total Active power generated by renewable generation and substation

Looking at the active power consumption by the loads, the graph has a pretty normal looking shape with peaks during the day and a trough during the night. The two noticeable peaks occur at 42,000 seconds and 71,000 seconds or around

11h40 and 19h45 respectively. A trough can be found at around 12,000 seconds or 3h20.

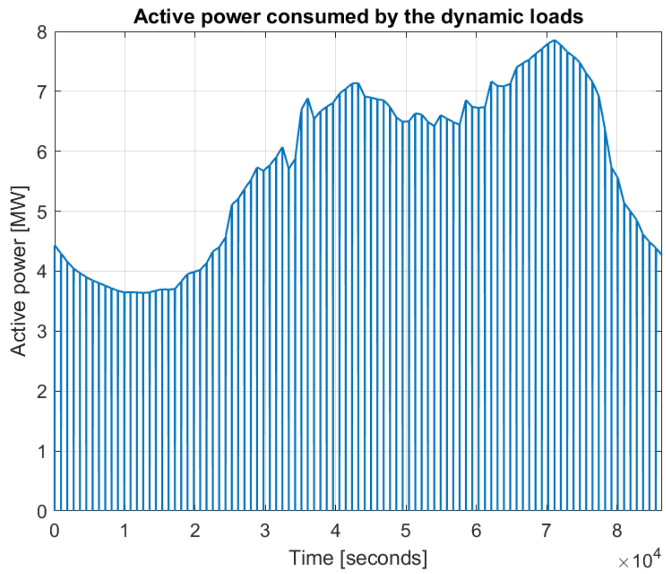


Figure 4.9: Active power consumed by the dynamic loads

If we compare the two (Fig 4.10), there is clearly a discrepancy between them. The discrepancy remains mostly constant and is attributed to the grid losses. At the peak of the generation and consumption, the grid losses are also the highest. As they are proportional to the square of the current, this makes sense. The grid losses of active power can be found in figure 4.11. These losses are quit significant (up to almost 1MW during peak demand) and are probably a little bit too high to be accurate, as the transmission losses on active power are 9,7% during this time.

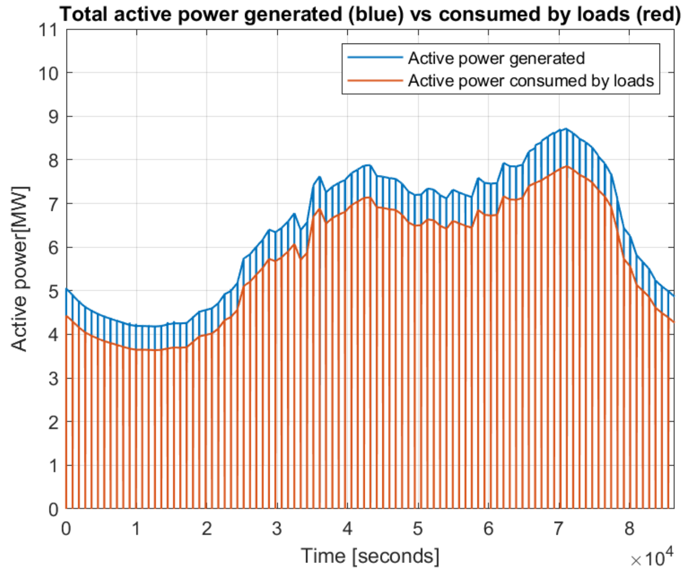


Figure 4.10: Comparison between active power generated and consumed

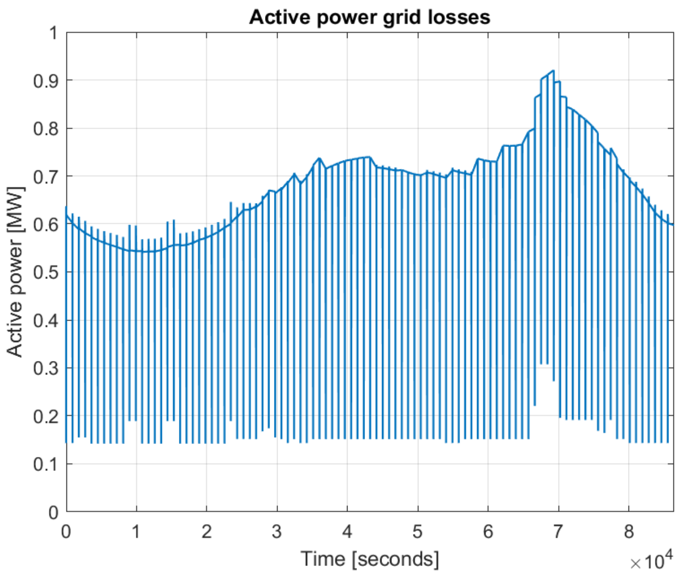


Figure 4.11: Active power consumed by grid losses

Grid losses can be divided into two kinds. Fixed losses and variable losses. Fixed losses are not proportional to the amount of current that is flowing. A notable example of a fixed loss in this simulation is the no-load losses of the transformer. As can be seen in Figure 4.12, the difference between active power going in to the substation and exiting out of it, is almost the same. This means the active transformer losses (which are partially made up of no-load losses)

are minimal. Losses like corona losses and leakage current losses are also of significant importance in a real MVDN, they are however not incorporated into the simulation.
Variable losses do change according to the current and are proportional with the square of this current. Examples are Joule losses in the lines, losses due to the contact resistance of network components etc. In Figure 4.11, it is apparent that the losses are highest when the active power demand is highest too. Due to the quadratic relationship between current and variable losses however, it's clearly visible that not all losses can be attributed to variable losses. As the only fixed losses modeled in into the simulation are the no-load losses from the transformer and these appear to be almost zero, they have to come from somewhere else. It is however not entirely clear what the cause of these losses is. It might possibly be a simulation error or some losses that occur in certain network element, which are not being disclosed in the Simulink documentation. It might be interesting to look into where this ghost consumption of power is coming from and how to solve it, in the future.

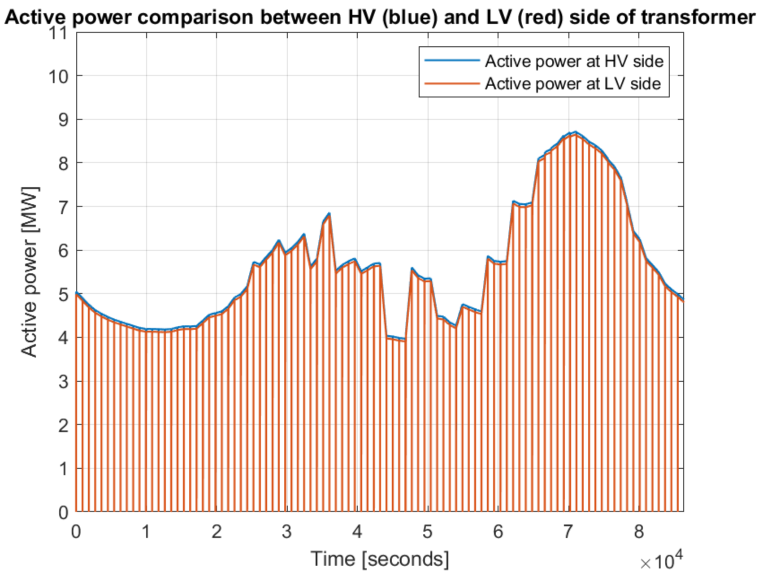


Figure 4.12: Active power comparison between both sides of the substation

Power generated and consumed should always be equal. If not, this will increase or decrease the voltage and frequency of the network, which is unwanted. This means that:

$$P_{renewablegeneration} + P_{substation} = P_{dynamicloads} + P_{gridlosses} \tag{4.2}$$

The same holds true for the reactive power.

When it comes to the reactive power in the system, you would assume to the same train of thoughts can be made for the reactive power. There is however a slight difference between the two when it comes to the renewable generation as it doesn't generate any reactive power if we take a look at the generation data. This means all of the reactive power demanded by the loads will have to come from the substation and the capacitor banks. The latter are supplying reactive power to the system. As the load of a run-of-the-mill MVDN has an inductive nature most of the time, this also helps with reducing the overall reactive power in the grid, and thus current flowing through the lines, in turn lowering the grid losses.

We once again take a look at power consumption of the loads. This time for the reactive power. It can be noted that in the loads data, the active and reactive power consumed by the loads are proportional. This explains why both are very similar in shape. The capacitor banks are also added to the loads (Fig 4.13), a figure without capacitor banks can be found below as well (Fig 4.14). The difference between the two is quite noticeable. Even though the reactive power supplied by the banks is constant, they do a good job at lowering the total reactive power in the MVDN and should attribute to a higher grid efficiency.

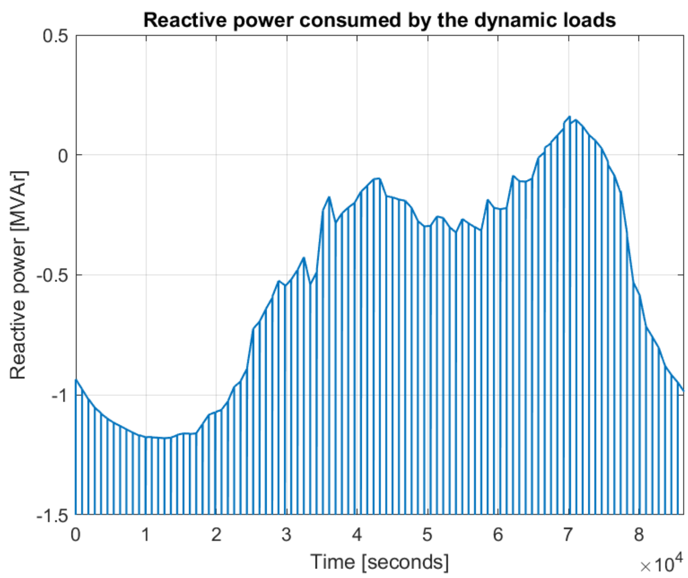


Figure 4.13: Reactive power consumed by the dynamic loads

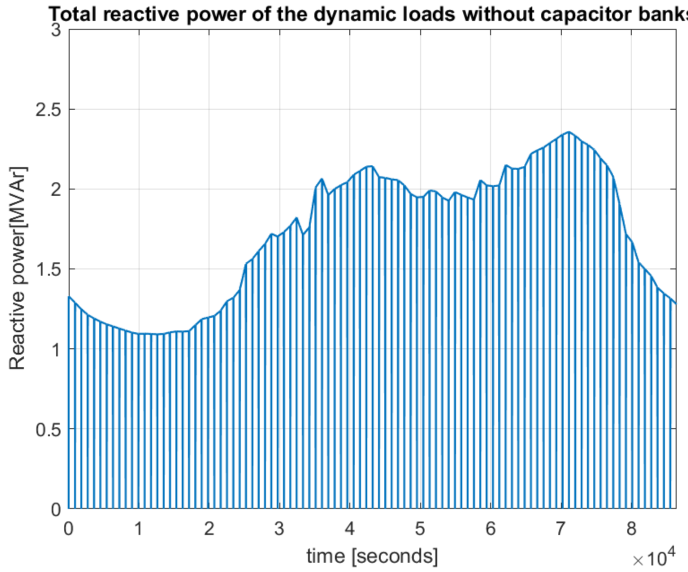


Figure 4.14: Reactive power consumed by the dynamic loads, excluding the capacitor banks

The total reactive power generated is compared to the power consumed. A disparity is found between the two, just like it was the case with the active power generation and consumption comparison. This can be attributed to the losses in the grid.

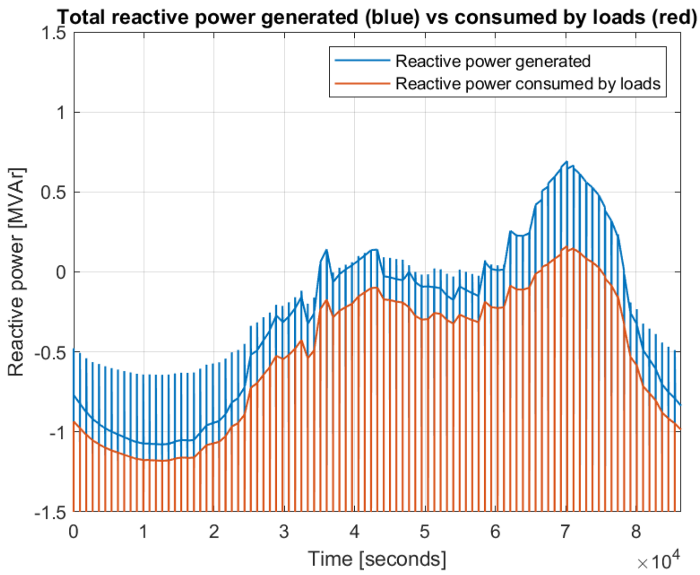


Figure 4.15: Comparison between reactive power generated and consumed

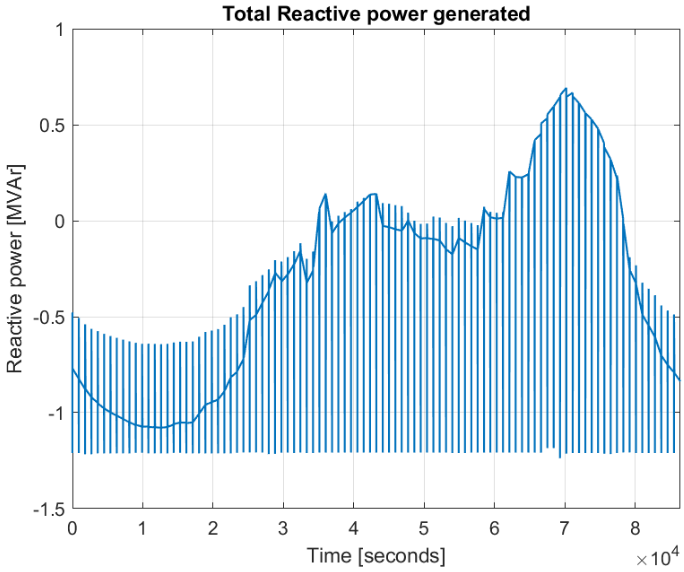


Figure 4.16: Total Reactive power generated

When it comes to the reactive losses in the grid that once again there is a peak at the moment of highest energy demand, revealing that variable losses play their part in the total losses. The fixed losses in the substation play a noticeable role this time, as can be seen in Figure 4.18. The fixed reactive power losses here can be attributed to the magnetisation losses in the core. This is clearly visible when we compare the high voltage and low voltage side of the substation in Figure 4.18.

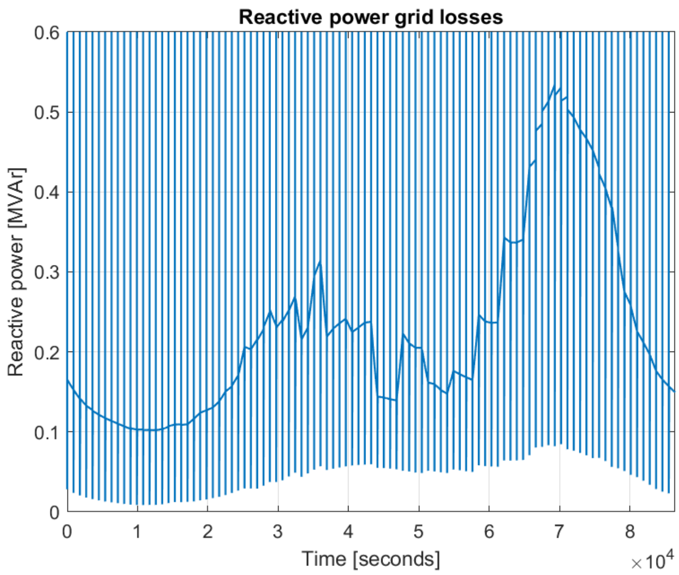


Figure 4.17: Reactive power consumed by grid losses

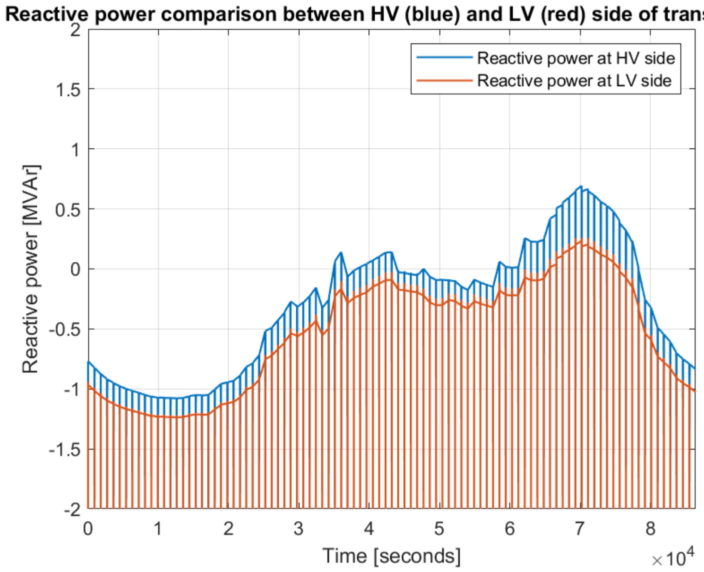


Figure 4.18: Reactive power comparison between both sides of the substation

4.3 Conclusion

This chapter started off with a look at the Breaker Problem, a complication in the simulation that resulted in very long simulation times and incorrect evaluation of the Simulink model. This problem was, as the name suggests, caused by the three-phase breaker component. As this problem only persisted in certain scenarios, further tests scenarios were explored to find the exact cause of the issue. It became apparent after some tests and a lot of hours spent constructing the simulation that the root of the problem lied in the three-phase breaker being used in conjunction with the dynamic load block. Multiple routes to solving this problem were explored and in the end there was decided to go with the approach of running every period separately. The data from all the separate periods is then concatenated together. This also explains the spikes (caused by incorrect estimation of the initial value of the models differential equation) and inverse colors in some of the graphs. Next, the obtained results were discussed. A look was taken at the active and reactive power generated by the renewable generation and the substation, the power consumed by the loads. A comparison between the total generated and consumed power, the difference between the two yielding the grid losses. The difference between the high voltage and low voltage side of the transformers to check for transformer losses. The data all looked plausible, except the grid losses. In real grids, these losses usually consume 2-6% of the total power in the grid. In the simulation, the grid losses came close to 10%. Some component with undisclosed losses or a simulation error is clearly eating up a portion of the power in the grid.

Chapter 5

Conclusions

In this chapter, the general conclusions regarding the results of the simulation will be discussed. The predefined objectives will be evaluated and to conclude this dissertation, a few suggestions will be made on how to expand on this work in its current form.

5.1 Conclusions on the simulation of a MVDN in a smart grid context

This dissertation began with the a small introduction where the problem that work tries to help solve was explained, along with a motivation for writing it in the first place. In this chapter, the objectives for this work were also determined. It talks about how it is important to model the network components as close as possible to their real behaviour as only then one is able to extract a sensible result from the simulation constructed with these network component models. This is what was attempted in the second and third chapter. In the second, the network components of a MVDN were laid down and discussed to give a bit more insight into their inner workings and how they all come together to form a MVDN as a whole. A deeper dive was taken into what exactly smart grids are and how they work, discovering on the way that they are a very promising innovation for the future of distribution networks. As the share of renewable energy will increase even more in the future, the network will have to adapt to these rapidly changing power flows and longer distances over which the energy from this generation will have to travel. The longer distance has to do with the geographical dependence of most types of renewable generation. As the electrical distribution infrastructure is aging and the share of electrical vehicles on the road increases each year, the distribution grid will slowly start having difficulties keeping up with the increased demand. One way of keeping up with this increased future demand is by reinforcing the network through replacing the distribution lines with heavier ones. As we don't know what the future holds, this might not be the most cost-effective or durable solution out there. Smart grids are most likely a more future-proof solution.

When it comes to the simulation process itself, a software environment to construct the model had to be chosen. Because of its vast amount of libraries and toolboxes and high support in the form of forums, help functions and manuals, MATLAB Simulink was the software environment of choice. The only downside it has compared to its competitors was that it is not open source and the software itself costs $2000 for a perpetual license. In Simulink, the algorithm taking care of computing the different states of the simulation is called a solver. There are quite a lot of solvers available that all have their respective strengths and weaknesses. The chosen solver is ODE23tb. A variable, implicit solver that is able to solve stiff problems as well. Besides all these properties, the solver was also chosen because of its exceptional solving speed while maintaining a reasonably high accuracy. The high solving speed is very welcome because of the time simulated being upwards of 24 hours. Once the solver is chosen, the different models for all the network components have to be decided. The Simscape toolbox provided a lot of compatible model blocks that could be combined to construct an entire MVDN model.

Once the model was constructed, the simulation of the MVDN could begin. There were some obstacles on the way in trying to collect usable data from the network model. The Breaker Problem, named after the 'three-phase breaker' block that caused the issue, made the simulation time extremely lengthy and above that also resulted in a poor evaluation of the network model by the solver, yielding unusable network data. After formulating some hypotheses regarding the cause of the issue and testing those hypotheses, it became quite apparent that the issue always occurred when the breaker was used in combination with the three-phase dynamic load block. The issue was solved by running each period of the simulation separately and concatenating the data from all these periods together to obtain the resulting network data.

When plotting and analysing the data, the network seems to perform well. The renewable generation produces active power and injects it into the grid. The remaining power demand is met by the substation. It appears as if the wind turbines are not working at night. This is actually the case but only has something to do with the data supplied being from a nearly windless day. When tested with higher wind speeds, the model could handle these fine. When the total active generation and power consumed by the loads is compared, the is a clear disparity between both. This is caused by grid losses taking up a small part of the power consumed. The grid losses here are almost 10%. This is far higher than expected. It was discovered this way that there is a component with undisclosed losses or a simulation error eating up active power in the grid. When looking at the reactive power data and once again comparing power generated with power consumed, there is once again a disparity between both. Power at both the HV and LV side of the substation was compared and the transformer losses were quite small compared to the total grid losses when it came to active power. The reactive losses in the transformers were quite noticeable however and could very well explain a big portion of the reactive grid losses.

5.2 Suggestions and future works

As with every work, there can always be improved and expanded on. In this section, suggestions will be given on which can be build in further works.

1. The first thing that comes to mind is finding a way to run this simulation in a continuous way. As of now a workaround was used to gather results from the simulation. It would require a lot less work and remove the spikes in the data if the simulation were to be able to run without the use of a MATLAB script simulating several periods one after the other in a loop and concatenating the results.

2. Further incorporating electric cars and their charging stations. During the construction of the simulation, test cases were made where charging and discharging of EV's was incorporated. There was however no collected or simulated data to input into the model so random numbers were used. It would be interesting to see the effect on the MVDN smart grid when charging and discharging values based off real data or the results of a human charging behaviour simulation.

3. As with every MVDN or distribution network for that matter, the operational reliability, as well as the well-being of the persons maintaining the network are of the highest priority. For this reason it would be interesting to take a look at how to properly secure an MVDN like this one to keep operational reliability and safety as high as possible. This will probably prove to be a challenge as the power flows change in magnitude and direction with the dynamic behaviour of the smart grid. Implementing this in a simulation might require some work as the safety devices available range from simple over-current protection to directional and impedance relays, to even very advanced energy monitors using a microprocessor that monitor currents and voltages and can detect faults immediately by doing some calculations using the voltage and current data.

4. As can be seen in Figures 4.11 and 4.17, the losses are quite considerable. Distribution losses lie somewhere in between 2% and 6% in practice. The losses we see here are close to 10%. This means there is a ghost consumption of power somewhere in the network. As the Simulink documentation doesn't disclose any other components having losses modelled in them besides the distribution lines and the transformers, it's not that easy to find the culprit. As a future work it would be interesting to do research regarding these undisclosed losses in the network. Where they come from and how to fix the power leak.

Bibliography

[1] S. Shafiee and E. Topal, "When will fossil fuel reserves be diminished?," *Energy policy*, vol. 37, no. 1, pp. 181–189, 2009. [citado na p. 1]

[2] A. Zaidi, K. Sunderland, and M. Conlon, "Impact assessment of high-power domestic ev charging proliferation of a distribution network," *IET Generation, Transmission & Distribution*, vol. 14, no. 24, pp. 5918–5926, 2020. [citado na p. 1]

[3] G. Mokryani, Y. F. Hu, P. Papadopoulos, T. Niknam, and J. Aghaei, "Deterministic approach for active distribution networks planning with high penetration of wind and solar power," *Renewable energy*, vol. 113, pp. 942–951, 2017. [citado na p. 1]

[4] F. Orecchini and A. Santiangeli, "Beyond smart grids – the need of intelligent energy networks for a higher global efficiency through energy vectors integration," *International journal of hydrogen energy*, vol. 36, no. 13, pp. 8126–8133, 2011. [citado na p. 1]

[5] P. Barker and R. De Mello, "Determining the impact of distributed generation on power systems. i. radial distribution systems," in *2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134)*, vol. 3, pp. 1645–1656 vol. 3, 2000. [citado na p. 1]

[6] R. Cossent, T. Gómez, and P. Frías, "Towards a future with large penetration of distributed generation: Is the current regulation of electricity distribution ready? Regulatory recommendations under a European perspective," *Energy Policy*, vol. 37, pp. 1145–1155, March 2009. [citado na p. 1]

[7] F. Adinolfi, F. Baccino, M. Marinelli, S. Massucco, and F. Silvestro, "Model of a real medium voltage distribution network for analysis of distributed generation penetration in a smartgrid scenario," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pp. 1–7, IEEE, 2012. [citado na p. 2]

[8] A. Singhal and R. Saxena, "Software models for smart grid," in *2012 First International Workshop on Software Engineering Challenges for the Smart Grid (SE-SmartGrids)*, pp. 42–45, 2012. [citado na p. 2]

[9] B. Canizes, J. Soares, Z. Vale, and J. M. Corchado, "Optimal distribution grid operation using dlmp-based pricing for electric vehicle charging infrastructure in a smart city," *Energies (Basel)*, vol. 12, no. 4, p. 686, 2019. [citado na p. 2, 38]

[10] G. P. Verbong, S. Beemsterboer, and F. Sengers, "Smart grids or smart users? involving users in developing a low carbon electricity economy," *Energy Policy*, vol. 52, pp. 117–125, 2013. Special Section: Transition Pathways to a Low Carbon Economy. [citado na p. 5]

[11] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010. [citado na p. 5]

[12] M. J, *Smart Grid: Fundamentals of Design and Analysis.* John Wiley Sons, 2012. [citado na p. 5]

[13] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine*, vol. 7, no. 2, pp. 52–62, 2009. [citado na p. 5]

[14] A. G. Phadke, "Synchronized phasor measurements in power systems," *IEEE Computer Applications in power*, vol. 6, no. 2, pp. 10–15, 1993. [citado na p. 5]

[15] L. Ferreira Costa, G. De Carne, G. Buticchi, and M. Liserre, "The smart transformer: A solid-state transformer tailored to provide ancillary services to the distribution grid," *IEEE Power Electronics Magazine*, vol. 4, no. 2, pp. 56–67, 2017. [citado na p. 5]

[16] M. Liserre, G. Buticchi, M. Andresen, G. De Carne, L. F. Costa, and Z.-X. Zou, "The smart transformer: Impact on the electric grid and technology challenges," *IEEE Industrial Electronics Magazine*, vol. 10, no. 2, pp. 46–58, 2016. [citado na p. 5]

[17] A. Khan, S. Memon, and T. P. Sattar, "Analyzing integrated renewable energy and smart-grid systems to improve voltage quality and harmonic distortion losses at electric-vehicle charging stations," *IEEE Access*, vol. 6, pp. 26404–26415, 2018. [citado na p. 5]

[18] H. Lund, A. N. Andersen, P. A. Østergaard, B. V. Mathiesen, and D. Connolly, "From electricity smart grids to smart energy systems – a market operation based approach and understanding," *Energy (Oxford)*, vol. 42, no. 1, pp. 96–102, 2012. [citado na p. 5]

[19] M. Brenna, E. De Berardinis, L. Delli Carpini, F. Foiadelli, P. Paulon, P. Petroni, G. Sapienza, G. Scrosati, and D. Zaninelli, "Automatic distributed voltage control algorithm in smart grids applications," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 877–885, 2013. [citado na p. 6]

[20] C. J, *Hoogspanningsnetten*. Acco, 2020. [citado na p. 7]

[21] C. J, *Energetische Aspecten van de Elektriciteitsproductie*. Acco, 2020. [citado na p. 8, 13, 14]

[22] P. L, *Elektrische Machines 1: Transformatoren en Gelijkstroommotoren*. Acco, 2019. [citado na p. 9, 10, 34]

[23] CanadaTransformer, "Single Phase Transformers https://canadatransformers.com/transformer-phase/," 2020. [citado na p. 9]

[24] W. Roshen, "Ferrite core loss for power magnetic components design," *IEEE Transactions on Magnetics*, vol. 27, no. 6, pp. 4407–4415, 1991. [citado na p. 10]

[25] Y. E. Guide, "Equivalent circuit of transformer https://www.yourelectricalguide.com/2017/08/equivalent-circuit-of-single-phase-transformer.html," 2017. [citado na p. 12, 13]

[26] A. R. Klee H, *Simulation of Dynamic Systems with MATLAB and Simulink*. Taylor Francis, 2018. [citado na p. 20]

[27] I. The MathWorks, "Choose an ODE Solver https://www.mathworks.com/help/simulink/ug/choose-a-solver.htmlf11-69449," 2021. [citado na p. 20, 24]

[28] I. The MathWorks, "Solver https://www.mathworks.com/help/simulink/gui/solver.html/," 2021. [citado na p. 22]

[29] I. The MathWorks, "Fixed step solvers in Simulink https://www.mathworks.com/help/simulink/ug/fixed-step-solvers-in-simulink.html/," 2021. [citado na p. 22]

[30] I. The MathWorks, "Compare Solvers https://www.mathworks.com/help/simulink/ug/compare-solvers.htmlbsfsnf4/," 2020. [citado na p. 23]

[31] A. G, *Modeling and Analysis of Electrical Power Systems*. 2004. [citado na p. 27]

Appendix A

Switching, Load and Generation data

In this appendix, you can find a few examples of the switching, loads and generation data used as inputs for the simulation to give a better understanding in how imported data actually alters the simulation and possibly how to alter the data so that you can run your own simulation with new data input.

A.1 Switching data

To feed data regarding when to open and close to the breakers, and in a later phase of the simulation process, when to comment breakers in and out, a carrier for this data has to be used. Since the connections that are made and broken always situate themselves in between two busses, the open and closed statuses can be captured by a matrix. The idea is that the row and column of the data convey in between which two busses the connection is situated. In this respective cell, a '1' or '0' differentiates the closed and open status respectively. It might come to mind that there are 2 possible cells when it comes to the location of the switching data for a random connection point, as rows and column are interchangeable. As a rule of thumb, looking from the perspective of the substation, the row number indicates the bus closest to the substation. In other words, this is the bus where the power from the substation flows 'first' the column number then indicates to which bus this power flows. To make sure that all the data was read correctly by the MATLAB script, both possible switching data locations where read in and a function determining the maximum of both cell values is used to make sure no data is overlooked. Below you can find some examples of the switching matrices for the 1st, 33rd, 54th and 92nd period, these periods where completely chosen and random and just serve to show the switching changes happening throughout the simulation.

switchFile.ResultsFinal(1).radBin

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A.1: 1^{st} Period of the switching data

switchFile.ResultsFinal(33).radBin

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	0
11	0	0	0	0	0	0	0	1	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0

Figure A.2: 33^{rd} Period of the switching data

switchFile.ResultsFinal(54).radBin

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A.3: 54^{th} Period of the switching data switchFile.ResultsFinal(92).radBin

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A.4: 92^{nd} Period of the switching data

A.2 Loads and Generation Data

Loads and generation data will be joined into one section as they don't differ too much from one another. The data has the possibility to change every period, just like the switching data. As discussed before, the loads and generation data is read from a matrix and fed into a Simulink lookup table that then passes on the appropriate loads and generation data to a dynamic load which handles everything from there. This done for both the active and reactive energy data for loads and generation. An example for the active loads and generation can be found below with increments of 900 seconds or 15 minutes per period. Every column stands for a load or generation somewhere in the network.

PLoadFile.tActiveLoadData

	1	2	3	4	5	6	7	8	9
1	0	0.2570	0.0700	0.0101	0.0638	0.0650	0.0609	0.0685	0.0674
2	900	0.2555	0.0661	0.0098	0.0614	0.0625	0.0586	0.0659	0.0648
3	1800	0.2264	0.0635	0.0096	0.0592	0.0603	0.0565	0.0636	0.0625
4	2700	0.2029	0.0600	0.0094	0.0568	0.0579	0.0542	0.0610	0.0600
5	3600	0.1963	0.0600	0.0093	0.0545	0.0555	0.0520	0.0585	0.0575
6	4500	0.1963	0.0600	0.0092	0.0519	0.0528	0.0495	0.0557	0.0547
7	5400	0.1961	0.0600	0.0091	0.0500	0.0509	0.0477	0.0537	0.0528
8	6300	0.1961	0.0600	0.0090	0.0486	0.0495	0.0464	0.0521	0.0513
9	7200	0.1916	0.0600	0.0089	0.0475	0.0484	0.0453	0.0510	0.0501
10	8100	0.1906	0.0600	0.0088	0.0463	0.0471	0.0442	0.0497	0.0488
11	9000	0.1899	0.0600	0.0088	0.0450	0.0458	0.0429	0.0483	0.0474
12	9900	0.1893	0.0600	0.0087	0.0437	0.0445	0.0417	0.0469	0.0461

Figure A.5: Example of loads data used as input to the network

PGenFile.tActiveGenData

	1	2	3	4	5	6	7	8	9
40	35100	0	0.1911	0	0.0478	0.0478	0.0478	0.0478	0.0478
41	36000	0	0.1911	0	0.0478	0.0478	0.0478	0.0478	0.0478
42	36900	0	0.4319	0	0.1080	0.1080	0.1080	0.1080	0.1080
43	37800	0	0.4319	0	0.1080	0.1080	0.1080	0.1080	0.1080
44	38700	0	0.4319	0	0.1080	0.1080	0.1080	0.1080	0.1080
45	39600	0	0.4319	0	0.1080	0.1080	0.1080	0.1080	0.1080
46	40500	0	0.5440	0	0.1360	0.1360	0.1360	0.1360	0.1360
47	41400	0	0.5440	0	0.1360	0.1360	0.1360	0.1360	0.1360
48	42300	0	0.5440	0	0.1360	0.1360	0.1360	0.1360	0.1360
49	43200	0	0.5440	0	0.1360	0.1360	0.1360	0.1360	0.1360
50	44100	0	0.8985	0	0.2246	0.2246	0.2246	0.2246	0.2246
51	45000	0	0.8985	0	0.2246	0.2246	0.2246	0.2246	0.2246

Figure A.6: Example of generation data used as input to the network

Appendix B

Network Model

In this appendix you can find the figures of the whole Simulink model, including the model divided into smaller figures for clarity. The different components should be recognisable by their respective symbols or pictures. Understanding every detail is of little importance, this appendix is here to give the reader insight in how the network is constructed and which parts are connected to others. The busses, loads and generations and substation should are the most important parts.

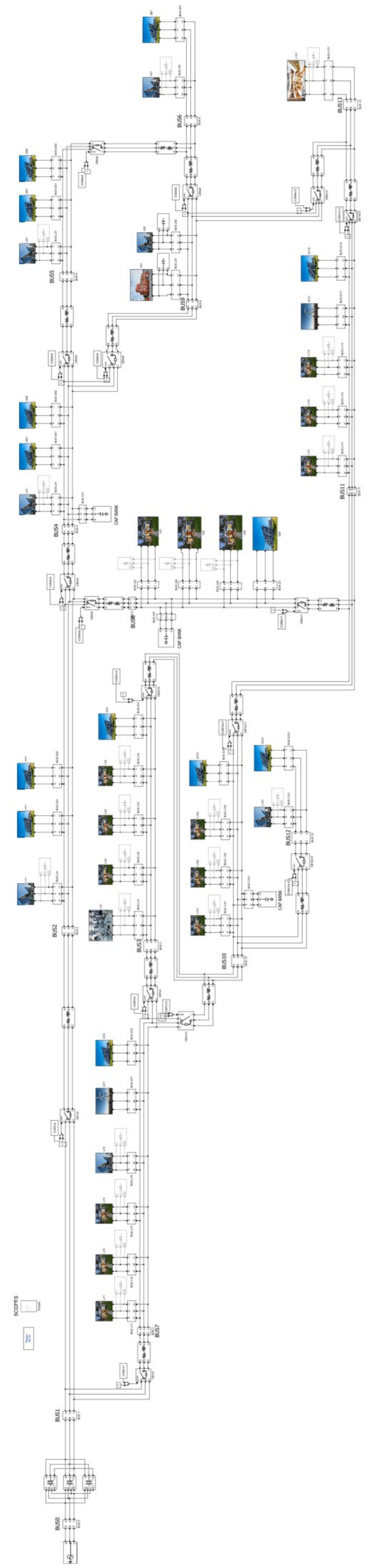


Figure B.1: Figure of the entire Simulink model

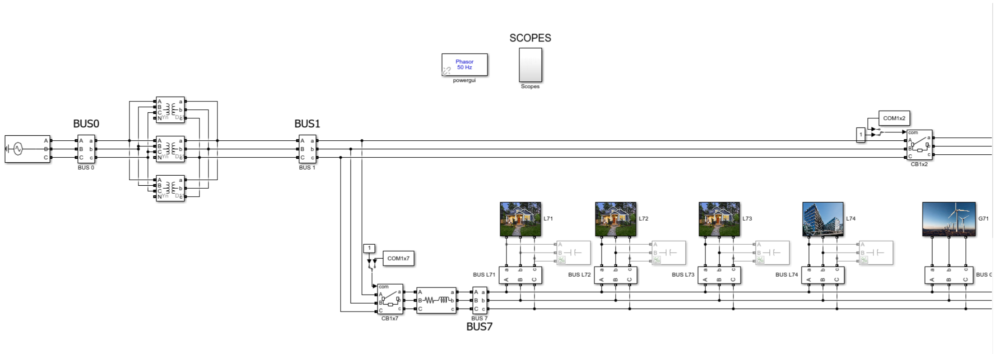


Figure B.2: Close up figure of the network part 1

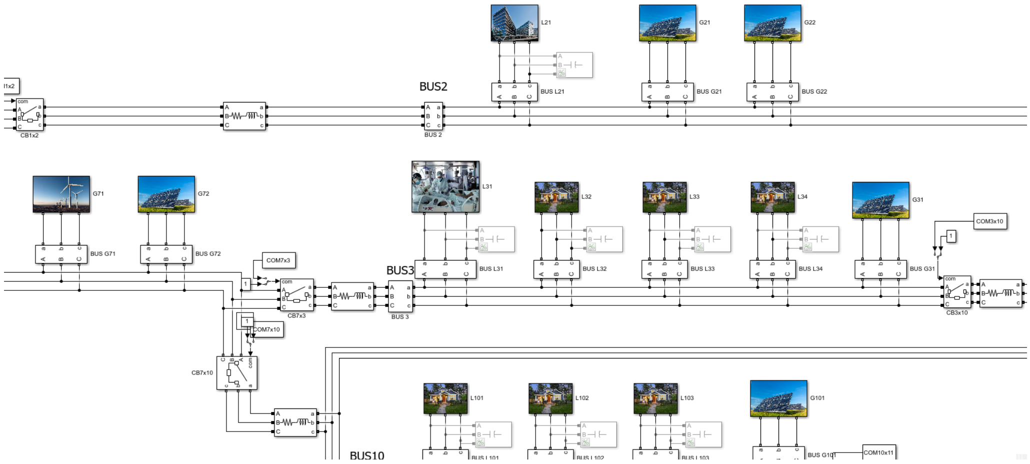


Figure B.3: Close up figure of the network part 2

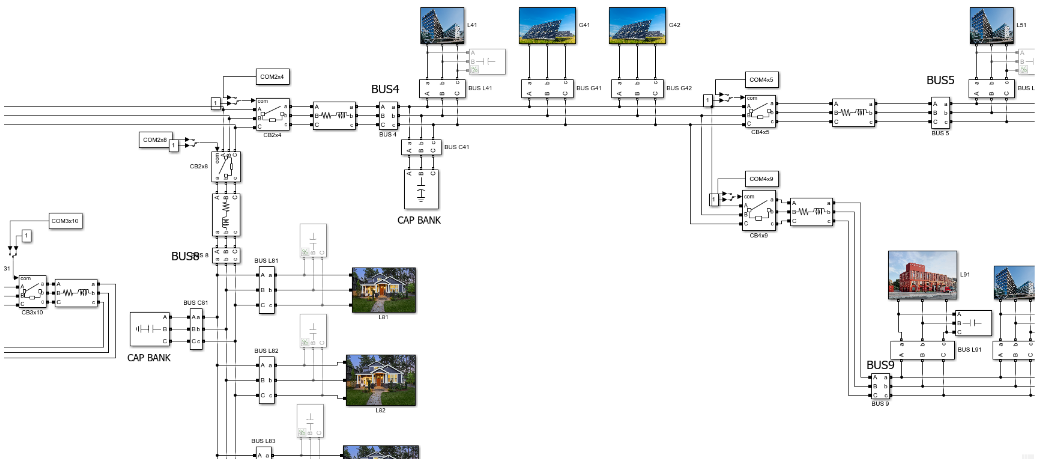


Figure B.4: Close up figure of the network part 3

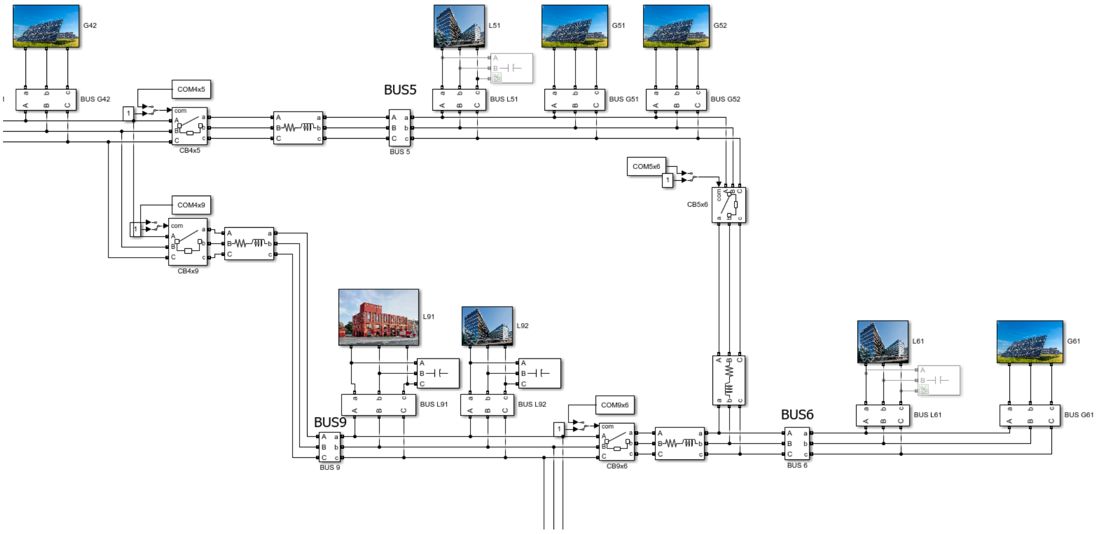


Figure B.5: Close up figure of the network part 4

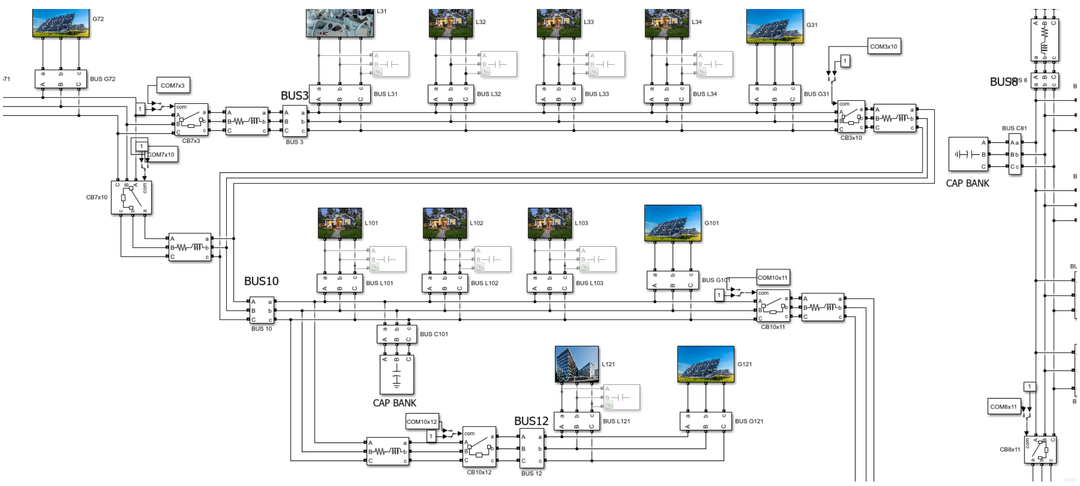


Figure B.6: Close up figure of the network part 5

Figure B.7: Close up figure of the network part 6

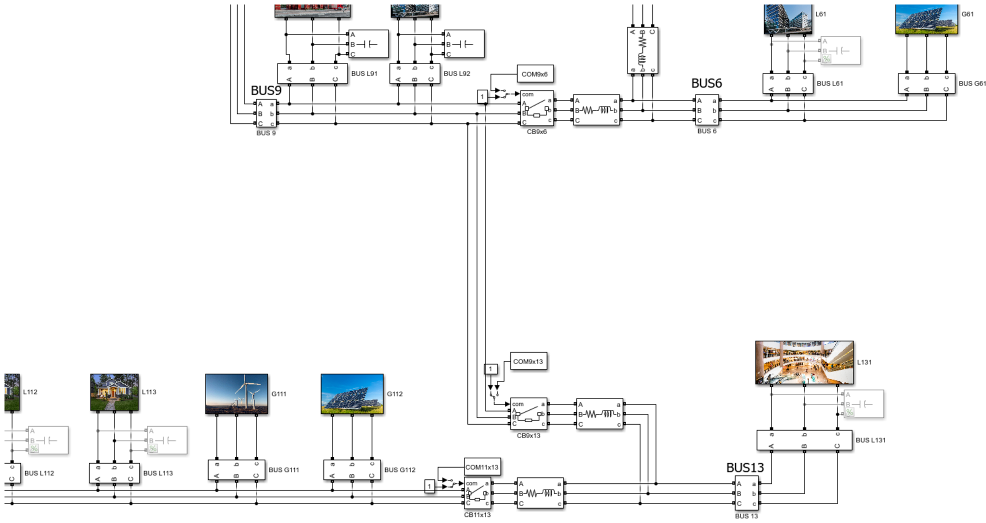


Figure B.8: Close up figure of the network part 7

Appendix C

Charging station example

Due to lack of actual charging station data, they were not included into the full simulation. The model for the charging station was developed nonetheless. An example with random data of what the charging behaviour might look like (for 96 hours) can be found below.

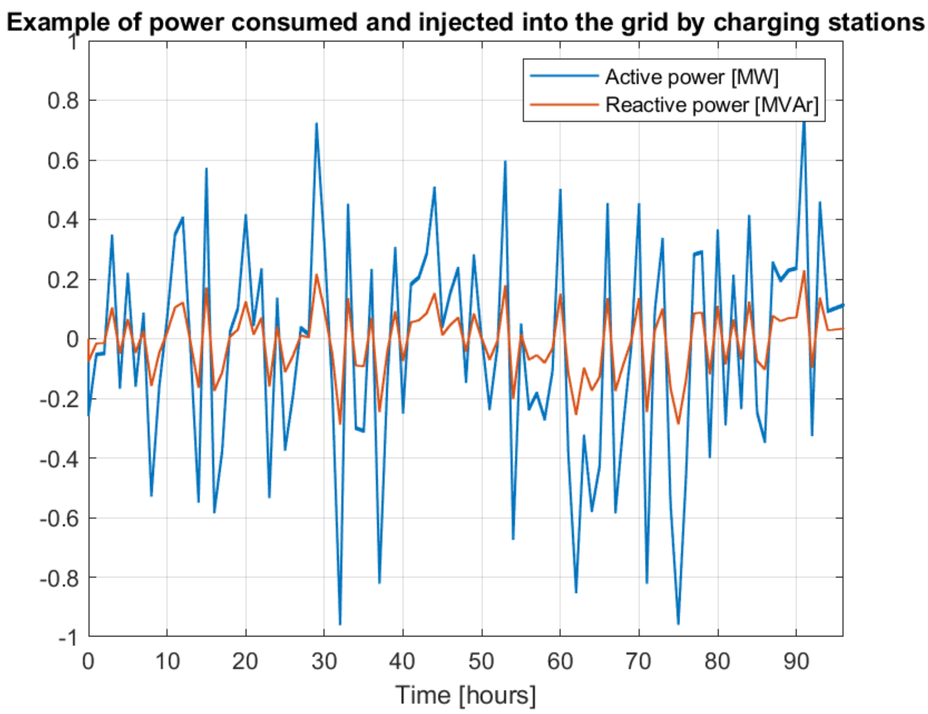


Figure C.1: Charging station example with random power inputs