Impact of High Penetration of Electric Vehicles on Low Voltage Network Stability and Role of Management Programmes for Electric Vehicle Charging

Stian Namtvedt Gjelsvik

Supervisor
Professor Mohan Lal Kolhe

University of Agder, 2015

Faculty of Engineering and Science

Department of Engineering

Abstract

The number of electric vehicles (EVs) have increased in Norway over the last couple of years. Low emissions and reducing petroleum dependence are some of the advantages achieved by using EVs. This master thesis presents the impact of high penetration of EVs on a low voltage distribution network. Uncontrolled charging patterns of EVs will likely stress the distribution network and could cause power outages, voltage fluctuations, thermal stress on the lines and harmonic pollution. As an alternative to grid reinforcement, smart load management (SLM) and demand side management (DSM) were introduced for coordinating the charging of large numbers of EVs.

Simulations of uncoordinated and coordinated charging of EVs were conducted with the power grid calculation software DIgSILENT PowerFactory and the proposed management algorithms written in MATLAB. The network was simulated using none, two and eight EVs connected. The Nissan Leaf battery specifications were used as a reference, where 3,3 kW and 6,6 kW charging power were tested. The impact of the original scheduled EV loads were compared with the results from applying the management programmes to the system, and discussed with respect to network stability and user satisfaction.

The transformer was the main bottleneck for the system, but also overloading of overhead lines limited the charging of the EVs. Of the programmes tested, the minute-based SLM programme gave the best results. The constraints were adhered, and the delay of the charging was reasonable. Increased charging power was also possible for this programme during the most power demanding periods with similar results.

Preface

The following report represents the Master Thesis "Impact of High Penetration of Electric Vehicles on Low Voltage Network Stability and Role of Management Programmes for Electric Vehicle Charging”. The report is related to the EU FP7 Scalable Energy Management Infrastructure for Aggregation of Households (SEMIAH) project, where Mohan Lal Kolhe is technical manager. www.semiah.eu/project-organisation/. The organization work towards commercial, technological and scientific breakthrough within demand response programmes in households.

The University of Agder gratefully acknowledge the support for this work of the European Commission under the FP7 program with contract number FP7-ICT-2013-11-619560 for project SEMIAH ‘Scalable Energy Management Infrastructure for Aggregation of Households’. http://www.semiah.eu.

Firstly, I would like to express my gratitude and thanks to my supervisor Professor Mohan Lal Kolhe for presenting an interesting and relevant thesis topic whilst providing valuable guidance through my research. Secondly, I would like to acknowledge and thank our study coordinator Stein Bergsmark, for his guidance through the master programme and master thesis.

I would also like to thank Knut Rydland at "Skagerak Energi", Ole Gaute Hovstad and Kristen Skrivarvik at "Sogn og Fjordane Energi" and Rolf Erlend Grundt at “Agder Energi” for providing useful technical information and data.

Thanks to my cohabitant and family who have been helpful and supportive through the master thesis writing period.

Thank you,

Grimstad

Spring 2015

Stian Namtvedt Gjelsvik

Contents

List of Tables

List of Figures

1 Introduction

Electric vehicles (EVs) have become a popular means of transportation during the last years due to several environmentally friendly, practical and economic advantages. There are current benefits to buying and using EVs in Norway, such as lower annual vehicle duties, also the removal of toll road costs and parking costs compared to vehicles with combustion engines. On the other hand the increasing demand of EVs are presenting some disadvantages. The EVs are often charged in residential areas and with uncoordinated charging, the low voltage distribution networks are facing challenges meeting the new load requirements. The distribution network operators want to avoid the potential for power outages and voltage fluctuations, which could be the result of uncoordinated EV charging. Without assessing the impact of this, power lines may need to be replaced or reinforced which is expensive and time consuming.

In this chapter, the motivation and problem definition for this thesis will be explained. The main goals will be presented along with assumptions and limitations, literature review, solution strategy and the thesis outline.

1.1 Motivation

In many countries such as Norway the presence of EVs are increasing significantly. In December 2014, there were 43 442 EVs and plug-in hybrid electric vehicle (PHEV) registered in Norway and the numbers are increasing every month [5]. Some locations in Norway where the electricity network is especially weak the high penetration of EVs presents an immediate challenge. Non-programmed charging of EVs will bring more demand on the power system. These uncontrolled charging patterns of EVs will impact significantly on the low voltage network stability and demand management. The cottage area in Hvaler is one example, where reinforcement of power lines could be postponed 10 years with a 25 percent reduction of the peak load [6]. The cost of reinforcing power lines are often unaffordable and also unnecessary. The power peaks should be avoided by a smart demand response programme.

In addition to the distribution network operator advantages using a smart demand response programme, the Norwegian Government together with the European Union have several environmental goals to be achieved by 2020. 20 percent reduction of greenhouse gas emissions compared to 1990 levels and 20 percent share in renewable energy in the energy sectors are two of them. Using EVs instead of vehicles with a combustion engine will help in reaching these goals.

The implementation of intelligent technologies in the electricity network such as advanced metering infrastructure and smart grid technologies are promising to increase. Demand response programmes for coordinated charging of EVs could be one of the implementations in the low voltage distribution network.

1.2 Problem Definition

There are possibilities for shifting the charging of electric vehicles through demand response programmes to off-peak time. Analysis of the impact of EVs charging patterns in a distributed network is very important especially for power system stability and power quality. These uncoordinated EVs power consumptions and random charging patterns significantly stress the distribution network causing voltage fluctuations, decreased power quality, power generation dispatching, degrading of system efficiency and so on. In order to prepare for an increased proliferation of EVs into the distributed network, the low voltage network must evolve with new infrastructure with demand response programmes and controlled charging patterns to support that.

1.3 Main Goal

This master thesis will analyze the impact of high penetration of electric vehicles in a low voltage distribution network with respect to stability and the role of demand response programmes for electric vehicle charging patterns.

The main goal will be solved through the following objectives.

Objective 1 Analysis of uncoordinated EV power consumptions and random charging patterns on distribution network for potential overloads, voltage fluctuations, stresses and degrading of system efficiency.

Objective 2 Development of a possible solution for smart load management (SLM) of EVs in a distribution network from the utilities perspective. Development of SLM control strategies and an algorithm for shifting charging of EVs to off-peak hours.

Objective 3 Development of a smart charging algorithm including line power loss minimization for future scenarios with larger share of EVs in the same distributed network.

Objective 4 Development of a smart charging algorithm based on demand side management (DSM). A possible demand limit provided by the distribution network operator applies to all residential power consumers connected to the low voltage distribution network.

1.4 Key assumptions and limitations

These are the following assumptions for this master thesis:

- The development of the models are based on information found in relevant literature.
- A typical Norwegian network is selected and it may vary with real conditions and situations.
- Introduction of new EVs with different characteristics may affect the network power quality and energy management methodology.

These are the following limitations for this master thesis:

- There are limitations in the simulation software, DIgSILENT PowerFactory, which only allows 50 busbars in the simulated system.
- Asymmetrical loads in the simulation are not accounted for.
- The EV loads are simulated as active power with linear battery characteristics.

1.5 Literature Review

This thesis is inspired by the short master thesis "Intelligent Energy Management of Electric Vehicles in Distribution Systems", which analyzes a Danish electrical network, while this project analyzes a typical Norwegian electrical network with high penetration of EVs [4]. A similar simulation has been observed in Frankfurt, where charging of EVs have caused overload and under voltage situations. A load management system has further shown positive effects on the low voltage network [1].

An IET Generation, Transmission & Distribution publication have done studies on peak shaving, improving voltage profile and minimization of power losses with large share of EV charging [3]. The same paper analyzes the impact of EV battery size on a smart load management system, which is of interest in this thesis. Also the paper "An Electric Vehicle Charging Management and its Impact on Losses" [2] contains relevant information concerning power line losses in the electrical grid and analysis of driving pattern of passenger cars in Denmark. The same method can be used for Norway.

An ongoing project, "ChargeFlex", is also working with a control system and hardware that collects power consumption data. An analysis of this data can be used to predict and control the power consumption of EVs [7]. The project is a co-operation between Smart Innovation Østfold AS, NTNU, Proxll, Østfold Fylkeskommune, Værste AS, Fredrikstad Energi Nett AS and SFE Nett AS.

1.6 Solution Strategy

In this work, the above-mentioned objectives will be achieved through analysis of typical Norwegian scenarios. The system will be developed and tested by DIgSILENT PowerFactory. Steady state, bipolar short circuit current and voltage analyzes of the low voltage network integrated with EVs will be carried out.

The development of smart charging algorithms will be done in MATLAB. It will also consider SLM control strategies and algorithms for coordinating charging of EVs based on DSM.

Official statistics will be found in Statistics Norway (SSB) or other relevant databases, such as [5].

1.7 Thesis Outline

The thesis focuses on the possibility to charge a large amount of EVs in a low voltage distribution network. The report is organized in chapters and subchapters where every chapters begins with an introduction. The chapters are arranged in the following way.

Chapter 2 describes relevant concepts and components to the project. Technical information regarding EVs, network topology and the smart grid concept will be given.

Chapter 3 presents a suggested solution to the problem. A low voltage network is described along with data for the system. Further, SLM algorithms and DSM strategies that will be tested are explained.

Chapter 4 presents the results of the different strategies and validation data for all of the different simulation scenarios.

Chapter 5 gives the discussion, and chapter 6 the conclusion with the different findings and implications. Finally, some further work is suggested.

2 Problem Analysis

In this chapter an overview of electric vehicles, charging of EVs, EV battery technology, Norwegian distribution network topology and the smart grid concept will be presented. The following will give understanding and knowledge of the EV concept and the distribution network used in this project.

2.1 Electric Vehicles

An electric vehicle (EV) is by definition a vehicle, which uses one or several electric motors for propulsion. The propulsion is powered by a rechargeable battery pack. The term EV applies to electric cars, trucks, trains, airplanes, boats and motorcycles. This project will focus on electric cars, which is the most usual power consumers of the mentioned in residential areas. Figure 2.1 shows a selection of different EVs which are powered only by electricity. There are also plug-in hybrid electric vehicles (PHEVs) available as an alternative, which have an internal combustion engine in addition to the electric motor [4]. With the possibility to recharge the battery, PHEVs are also an unpredicted load that can influence the distribution network. A selection of PHEVs are shown in Figure 2.2.

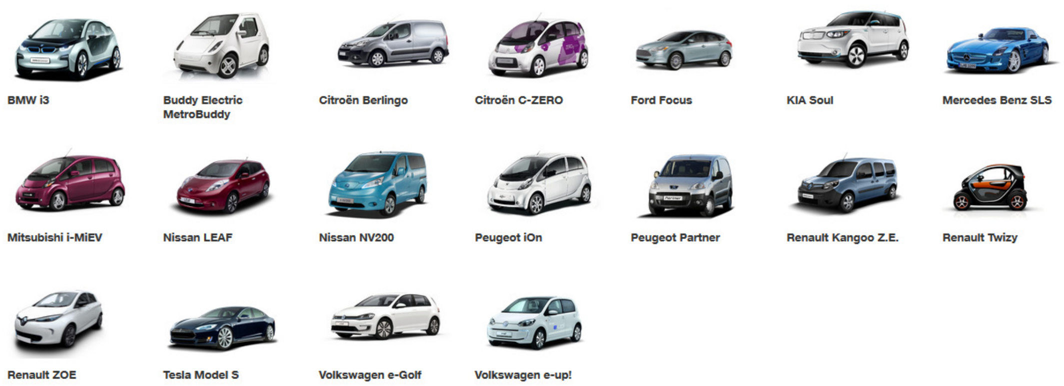

Figure 2.1: *Selection of different EVs available on the Norwegian marked [8].*

Figure 2.2: *Selection of different PHEVs available on the Norwegian marked [8].*

The main components of the power transmission in an electric car are the electric motor, the battery, the electric motor drive and the controller. Either direct current (DC) motors or alternating current (AC) motors are used to provide traction. Since batteries delivers DC voltage, the AC motors require more costly electronics for the AC to DC conversion. The power ratings of electric motors ranges from a couple of watt (W), to hundreds of kilowatt (kW) depending of the size. The DC motors are often heavier for the same power compared to the AC motors, but on the other hand needs less complicated electric drives. During previous years, permanent magnet motors have been introduced with very high efficiency and compact framing compared to the competitors, but due to rare-earth elements in the magnets they are more expensive.

The battery is the energy storage of the EVs. There are several types of different rechargeable batteries and ongoing research on new battery technologies, but this will be explained closer in Section 2.3.

Power electronics are used to manage the power and are dependent on car model and type of electric motor. In most cases, pulse width modulation (PWM) technology is used to control the voltage supplied from the battery. Transistors, such as insulated-gate bipolar transistors (IGBTs), are chopping the voltage so that desired frequency is achieved. The desired frequency is dependent on how much power the car user requires. An inverter is used to convert battery supplied DC voltage to AC voltage, and a DC/DC boost converter used to increase the voltage.

2.2 Charging EVs

The charging of EVs are essential and one of the bottlenecks in the electric vehicle marked. Two of the criteria for the end user buying an EV are often fast charging combined with long-range driving capacity.

The charging time should match with the EVs battery characteristic in order to perform an optimal charging and retain a long lifetime. To protect from current ripples inductors are usually integrated in EV chargers as a current filter. Secondly, the chargers should keep harmonic distortion within the standards of the electrical network, or as low as possible. It should also obtain a high power factor to maximize the real power available from the utility outlet. To achieve this the battery chargers often contain a boost converter for power factor correction (PFC) [11].

EV battery chargers must detect state of charge (SOC) of the battery, and it is often desired that the chargers can adapt to various battery types and car models. The manufacturer also needs to make sure the chargers are efficient and reliable since they are handling large amounts of power. The chargers are either located inside the vehicle, on-board, or outside of the vehicle, off-board. The on-board chargers require space inside the vehicle and are therefore made as small as possible which limits the power. The off-board chargers are often designed for fast charging and are less constrained by size [11].

The International Electrotechnical Commission (IEC) Committee for “Electric road vehicles and electric industrial trucks” and the Committee for “Plugs, Socket-outlets and Couplers for industrial and similar applications, including for Electric Vehicles” have defined four charging modes and three types of socket outlets. These standards applies to on-board and off-board equipment for charging and assists the international compatibility.

The IEC standard 61851-1 describes the four charging modes [9]:

Mode 1: Slow charging from a household-type socket-outlet in AC.

Mode 2: Slow charging from a household-type socket-outlet with an in-cable protection device in AC.

Mode 3: Slow or fast charging using a specific EV socket-outlet

Mode 4: Fast charging using an external charger in DC.

The IEC standard 62196-2 describing the three socket-outlets [10]:

Type 1: Single phase vehicle coupler

Type 2: Single and three phase vehicle coupler

Type 3: Single and three phase vehicle coupler with shutters

2.3 EV Battery Technology

An electric vehicle battery (EVB) is an electrochemical storage that can be charged and discharged, allowing storage or delivery of electric energy [11]. Since the battery is a large part of an EV both physically and economically, manufacturers are doing research to improve the technology whilst lowering the prices. This section contains an introduction of the most common batteries in the EV industry and some further explanation of the best-suited battery for EVs at present.

2.3.1 EV Batteries

There are several types of batteries, but Lead-Acid, Nickel-Metal Hydride and Lithium-ion are shown best suited for EVs. The most commercially used battery is the Lithium-ion, and the exact chemistry varies from the different manufacturers. Of the selection of EVs in Figure 2.1, only the “Buddy Electric MetroBuddy” is powered by a Lead Acid battery.

The Lead-Acid batteries have well proven technology, low cost and very low standby losses. They are often used as a start battery for combustion engines in cars since they can deliver high current for short periods. The drawbacks of the battery is that it is sensitive to temperature, it has large volume and weight compared to other batteries and they have a short cycle life.

Nickel-Metal Hydride (NiMH) batteries have higher power to weight ratio compared to the Lead-Acid batteries, as shown in Figure 2.3, and also longer life cycles. Due to a high self-discharge these type of batteries are not as commonly used in EVs as the Lithium-ion batteries.

Lithium-ion batteries have the highest power to weight ratio of the three compared batteries, low self-discharge and high energy efficiency. In addition to EV applications, these batteries are used in cell phones, laptops and other electronics [13].

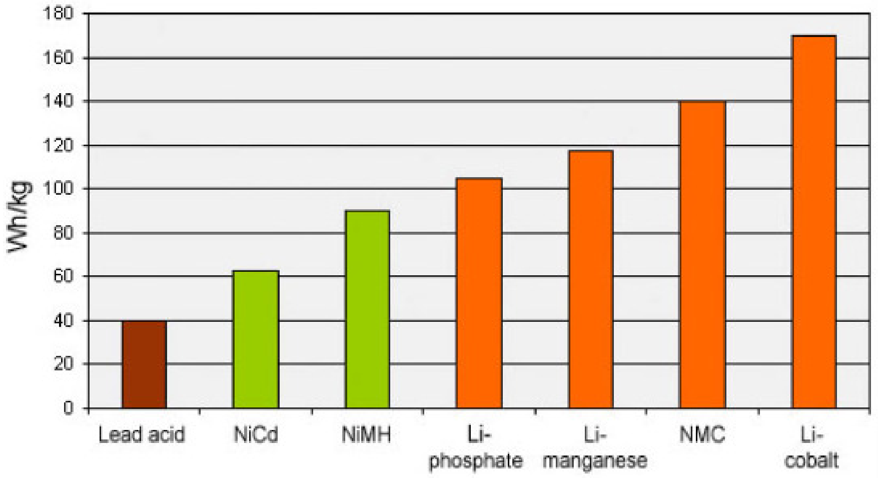

***Figure 2.3:** Energy densities of Lead-Acid, Nickel-Metal Hydride and Lithium-ion batteries [13].*

Battery characteristics that are important for end user, manufacturer and the environment are given below [14].

Safety is very important dealing with batteries mainly because of the thermal runaway. It should be robust designed with safety features such as pressure reliefs and shut-down separators. A battery must also be safe when exposed to misuse and advancing age.

Life span, which is a product of cycle count and longevity, are desired to be as high as possible in EVs. Capacity loss due to aging is challenging especially in hot climates. To compensate for this, manufacturers often increase the size of the batteries to allow for some degradation.

Specific energy is the energy per mass ratio of the battery which are one of the factors determining the driving range of the EV. The specific energy establish the amount of energy the battery can hold per kg.

Performance are affected by ambient temperature. EV owners have experienced a large reduction of driving miles per charging in the winter season when the temperature is low.

Specific power is a product of acceleration and most EV batteries are designed for this criterion. In most cases, an electric motor has higher output torque compared to a combustion engine. The power electronics drive or controller enables the motor to achieve high torque at low rotations.

Cost is the last factor and a major drawback. With today's manufacturing process, the battery represents a large share of the EV price. With the additional protection and safety features for the battery the price increase even more.

The cycle life and charging time are also two hot topics among EV manufacturers and customers. Most batteries will drop in capacity after an unknown number of cycles, but methods are carried out to extend these cycles without the capacity drop. Most batteries for EVs can be charged reasonably fast if suitable electrical power outlet is available [14]. The IEC standard 61851-1 also facilitates fast charging in two of the four charging modes [9].

2.3.2 Lithium-Ion Battery for EVs

The best suited battery for EV applications have this far shown to be the Lithium-ion battery, with energy capacities included between 5 kWh and 53 kWh according to [11]. A Li-ion battery package delivered in Toyota PHEVs are shown in figure 2.4 below.

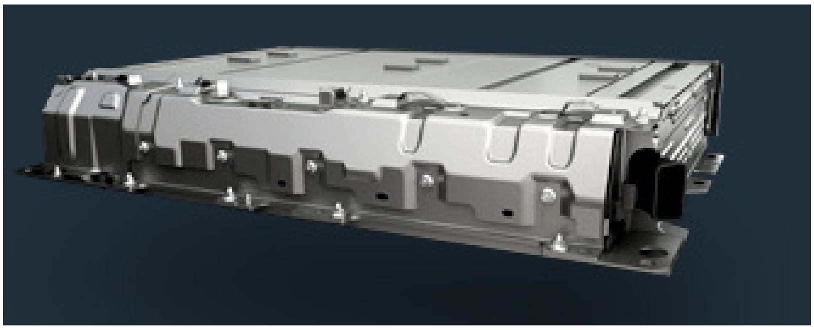

Figure 2.4: *Lithium-ion battery package of a Toyota PHEV [12].*

Like most batteries, the cells in the Lithium-ion battery consist of an anode and a cathode. The anode is usually graphite, but there are ongoing research trying to improve or replace it. Which cathode material used varies from which manufacturer producing the cells. In EVs Lithium Manganese Oxide, Lithium Iron Phosphate and Lithium Nickel Manganese Cobalt Oxide are most common, mostly because of safety reasons, high specific power and long lifetime [13]. The car manufacturer also vary the amount of cells in a battery, and placing the cells in series and parallel such desired voltage and current are achieved. The spider web diagrams in Figure 2.5 shows the most important characteristics of the Lithium-ion batteries.

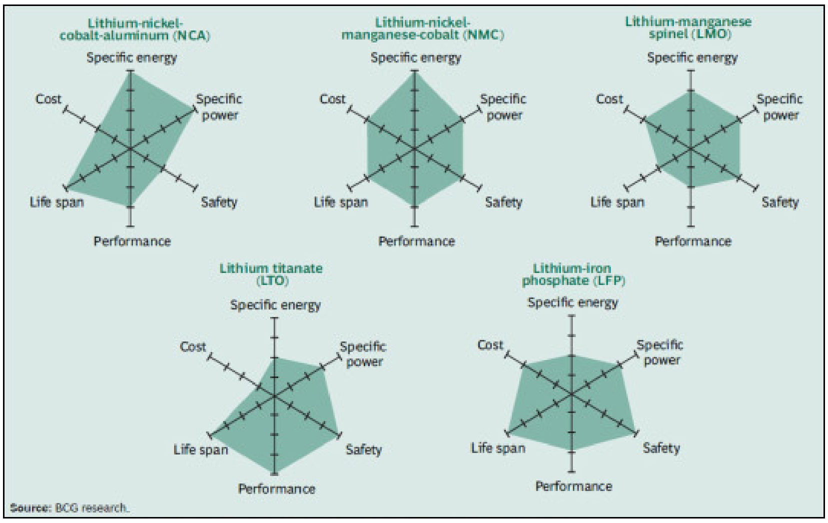

Figure 2.5: *Tradeoffs among the five Lithium-ions technologies, where Lithium Manganese Oxide, Lithium Iron Phosphate and Lithium Nickel Manganese Cobalt Oxide are most common [14].*

The main elements in a Lithium-ion battery are a negative electrode, a positive electrode and a chemical electrolyte in between. When the battery is charging the anode is positive and lithium ions moves from positive to negative electrode. When the battery is discharging, the cathode becomes the anode and lithium ions moves back. In reality, the charging and discharging are dependent of many factors. For example, the most efficient charging period will be in the first period of the charge cycle and the discharge depends on the driving pattern and external conditions. However, in this project the charging and discharging will be assumed linear to simplify the simulations.

2.4 Distribution Network

Analysis shows that high penetration of EVs mostly affect the low voltage distribution networks [1]. Scenarios that could negatively affect the distribution networks are large charging power, high charging frequency and sinusoidal distortion caused by the EV chargers. If the electrical networks have high impedance and therefore low short circuit performance, voltage disturbance may easily occur and the scenarios mentioned above will give even larger impact [15].

If the charging power and charging frequency are too high, which arise with high penetration of EVs within a given area, the power quality will decrease and voltage root mean square (RMS) value may drop below accepted level at the residences. If the EV charger cause distortion in the supplied current sinusoidal, heat losses may occur in transformers and other electrical devices [15].

2.4.1 Distribution Network Topologies

The Norwegian distribution network is the network that provides power to the end users. High voltage lines with capacity up to 22 kV are used at long stretches to avoid power losses, and low voltage lines with 230 V or 400 V are supplying residences. The high voltage distribution network normally operates with 22 kV or 11 kV and are design as radial network or meshed network. The lines are provided with overload protectors that disengages if short-circuiting occur [16].

The low voltage distribution networks are divided into three main categories: terra terra (TT), isolated terra (IT) and terra neutral (TN). The difference of the systems is the relationship between the low voltage network and earth as shown in Figure 2.6. In the TT system, the first terra tells us that the neutral of the transformer is grounded and the other terra means that loads in the network are grounded. In the IT system the neutral of the transformer is isolated and the loads are grounded. Both systems have voltage potential of 230 V between two of the conductors.

The TN systems are divided into TN-S and TN-C where the N tells us that loads of the network are connected to the neutral of the transformer with a PEN-/PE-conductor. In a TN-S system, the PE conductor is separated from the N conductor and in the TN-C system the PE and N conductor are combined until the first distribution as shown at right in Figure 2.6. In this system, both three phase 400 V and one phase 230 V can be utilized. Most new installations are TN systems, but many old installations nowadays are outdated IT systems with low power capacity.

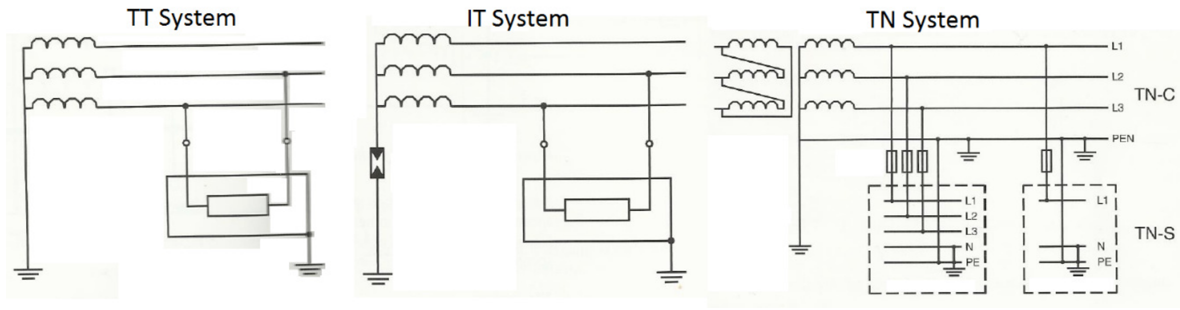

Figure 2.6: *Line diagram of the different low distribution network systems; TT, IT and TN [16].*

Equation (2.1) defines the resistance in an overhead line or cable. Inductance for two-phase lines and symmetrical three-phase lines can be calculated by Equation (2.2). Asymmetrical three-phase arrangements are calculated by Equation (2.3). Equation (2.4) gives the reactance per phase and kilometer. These data can usually also be found for specific conductors in suppliers datasheets. Capacitances in overhead lines needs to be included if the rated voltage exceeds 66 kV. Since distribution networks operates with rated voltage below 22 kV, the capacitances are excluded in this project [16].

$$R = \frac{P_f}{I^2} \quad [\Omega] \tag{2.1}$$

Where

P_f: Power loss in conductor
I : Conductor current
R: Resistance in conductor

$$L_1 = \left(0{,}5 + 4{,}6 * lg\frac{2D}{d}\right) * 10^{-4} \quad [\frac{H}{km} and\ conductor] \tag{2.2}$$

$$L_1 = \left(0{,}5 + 4{,}6 * lg\frac{2\sqrt[3]{D_1 D_2 D_3}}{d}\right) * 10^{-4} \quad [\frac{H}{km} and\ conductor] \tag{2.3}$$

Where

L_1: Inductance for each conductor
D : Distance between conductors
d : Conductor diameter

$$X = \omega * L_1 \quad [\frac{\Omega}{km}] \tag{2.4}$$

Where

Ω: 2*π*f
f : frequency
X : Reactance per phase and km

Voltage drop in lines can be found using Equation (2.5) [16]. The voltage drop is calculated from one bus bar to another where low letter “1” indicates the first bus bar and “2” the second.

$$\Delta U = \sqrt{3}\left(U_{f1} - U_{f2}\right) = \sqrt{3}(R * I_2 * \cos\varphi_2 + X * I_2 * \sin\varphi_2) \quad [V] \quad (2.5)$$

Where
$\left(U_{f1} - U_{f2}\right)$: Phase voltage. Voltage drop between bus bars
I_2 : Phase conductor current
φ_2 : Phase angle

Phase conductor current can be calculated using Equation (2.6) [16].

$$I_2 = \frac{P_2}{\sqrt{3 * U_2 * \cos\varphi_2}} \quad [A] \quad (2.6)$$

Short circuit currents is also of interest in distribution networks, and by definition characterized by conductors in a system making contact over an impedance that is much less than the load impedance during normal operation. If bipolar short circuit occur, impedances of anterior network must be included. Equation (2.7) is used to find the total impedance, and Equation (2.8) to find the bipolar short circuit current [16].

$$Z = R + jX \quad [\Omega] \quad (2.7)$$

Where
Z: Impedance

$$I_{k2} = \frac{U_n}{2 * Z} \quad [A] \quad (2.8)$$

Where
I_{k2}: Bipolar short circuit current
U_n: Line voltage

Current ratings of conductors must be within the limits of the electro technical norm, NEK400:2010. Equation (2.9) arrange the allowed load current, fuse rating and cable conductivity [21].

$$I_b \leq I_N \leq I_z \quad [A] \quad (2.9)$$

Where
I_b: Load current
I_N: Fuse nominal current
I_z: Cable conductivity

Last, the power loss in the distribution lines is given by Equation (2.10) [16].

$$P_{loss} = R * I_b{}^2 + X * I_b{}^2 \quad (2.10)$$

2.4.2 Regulation of Power System Supply

The Norwegian regulation of supply in the power system defines frequency, voltage levels, voltage asymmetry, overharmonics and transients among other things. Extracts of relevant regulations for this project are given below [17].

§ 3-2. Voltage frequency: System operator shall in areas that are temporarily without physical connection to adjacent transmission grids, ensure that the voltage frequency is normally kept within 50 Hz ± 2%.

§ 3-3. Slow variations in voltage RMS value: The distribution network operator shall ensure that slow variations in voltage RMS value is within a range of ± 10% of nominal voltage, measured as an average of ten minute, in connection point in low voltage network.

§ 3-4. Transient overvoltage, under voltage and voltage drop: The distribution network operator shall ensure that the voltage deviation by transient overvoltage, under voltage and voltage drop do not exceed the following limits in connection point with the respective nominal voltage level, U_N, for the respective time interval:

Table 2.1: *Transient overvoltage, under voltage and voltage drop values given by the Norwegian regulation of supply in power systems, § 3-4 [17]. The intervals are divided into voltage between 230 V and 35 kV, and voltage level higher than 35 kV.*

Transient overvoltage, undervoltage and voltage drop	**The maximum number allowed per. floating 24-hour period [kV]**	
	$0,23 \leq U_N \leq 35$	$35 < U_N$
$\Delta U_{stationary} \geq 3$ %	24	12
$\Delta U_{max} \geq 5$ %	24	12

§ 3-6. Voltage asymmetry: The distribution network operator shall ensure that degree of voltage asymmetry do not exceed 2% in connection points, measured as an average of ten minutes.

§ 3-7. Overharmonics: The distribution network operator shall ensure that the total harmonic distortion of the voltage waveform does not exceed 8% and 5%, measured as an average of ten minutes and one week, in connection points with nominal voltages from 230 V to 35 kV.

§ 3-10. Transients: Norwegian Water Resources and Energy Directorate may order those covered by these regulations to implement measures to reduce the scope or the consequences of transient overvoltages.

2.5 Smart Grid

Advanced metering infrastructure and real time data collection are necessary for successful management systems for EVs, and part of the upcoming smart grid technology. A smart electrical system will perform four essential tasks illustrated in Figure 2.7, which will provide more electricity to meet rising demand. This is done by increasing reliability and capacity of power supplies, integrating low carbon energy sources into the networks and increase energy efficiency [18].

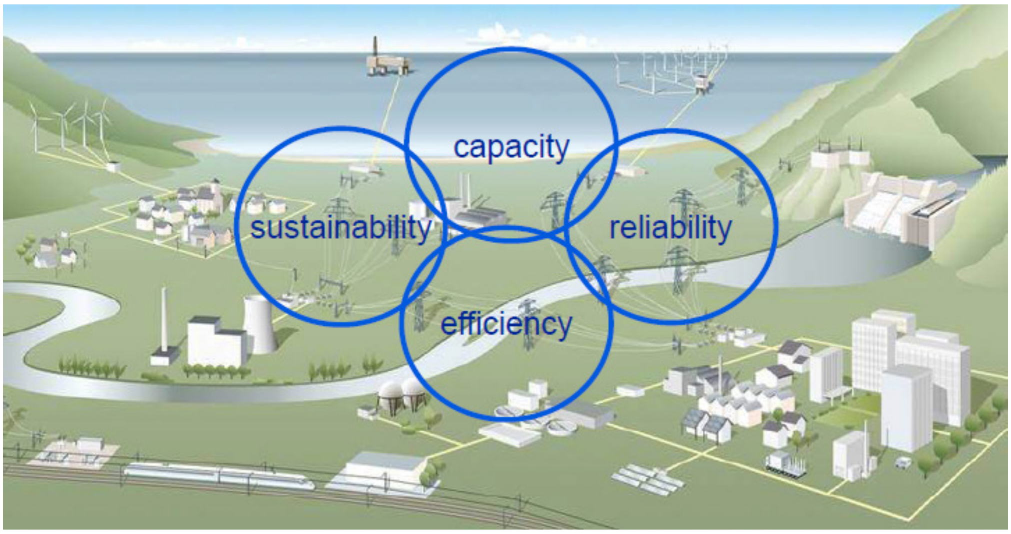

Figure 2.7: *The smart electrical systems four essential tasks marked in circles, with an illustration of where smart grid applications can be carried out in the background [18].*

According to [18], a smart grid is an intelligent system, based on industry-wide standards that provide stable, secure and environmentally sustainable electrical energy. Electrical systems will experience a major evolution, which will lead to a smarter grid. Some examples that is related to Figure 2.7 is remote monitoring and control of wind farms, demand response which means balancing load to supply, management of greenhouse gases, remote monitoring and control of solar farms, dynamic energy storage for load support and frequency regulation and remote grid operation with distributed generation [18].

The Institute of Electrical and Electronics Engineers (IEEE) organization are guiding the evolution towards a smarter grid by holding events, doing publications and writing standards. To provide the industry with information and resources to realize the smart grid, the "IEEE Xplore digital library" is available and includes journal articles covering the most current research and conference proceedings. Some of the standards developed recently are presented below [19].

- IEEE P2030 Draft Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), and End-Use Applications and Loads
- IEEE 802 LAN/MAN Standards Series
- IEEE SCC21 1547 Standards for Interconnecting Distributed Resources with Electric Power Systems
- IEEE Standard 1159 for Monitoring Electric Power Quality
- IEEE Standard 762: Standard Definitions for Use in Reporting Electric Generating Unit Reliability, Availability, and Productivity
- IEEE SCC 31 Automatic Meter Reading and Related Services

Of all the smart grid features, the smart metering is most relevant for this project. The PoweRline Intelligent Metering Evolution (PRIME) alliance provide advanced metering solutions which measures site-specific information. Traditional electrical meters only measures total consumption, while the PRIME solution include measurements of voltage and harmonic distortion, allowing diagnosis of power quality problems [20].

The Norwegian “ChargeFlex” project is using technology similar to the PRIME alliance, where the goal is to control the power output when charging EVs. Data is collected in one-minute intervals, as illustrated in Figure 2.8, which creates the foundation of the control system [7].

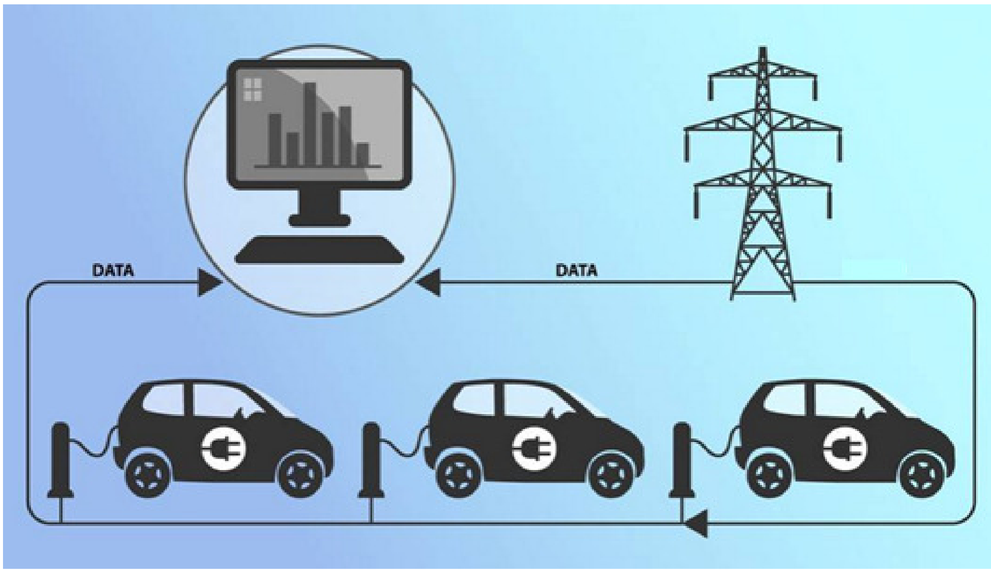

Figure 2.8: *Data collection of EV charging which is part of a smarter electrical system [6].*

3 Problem Solution

In this chapter a low voltage distribution network, load analysis, charging scenarios and management programmes will be presented. A low voltage distribution network that is suited for this project is chosen. Load profiles and data are established and simulation scenarios determined. The load analysis and charging scenarios will set the groundwork for the proposed management programmes. Lastly, future scenarios with increased penetration of EVs in the network will be carried out, and data transferred to the DIgSILENT PowerFactory software.

3.1 Network Design and Data

Figure 3.1 illustrates the distribution network analyzed in this project. All network data for the simulation was provided by Sogn og Fjordane Nett AS. This is an older residential area, where the network subscribers are marked with nine-digit tags, framed in black boxes. The figure also contains feeder line dimensions and locations of where the residential power consumers are located in the network. It is an IT system with nominal voltage of 230 V. The transformer supplies 20 households and is connected in wye-wye, which is the most common connection for distribution transformers. All transformer data is listed in Table 3.1. Due to internal losses, the low voltage output of the transformer is 238 V.

Table 3.1: *Transformer data for the low voltage distribution network.*

Transformer data:	
Rating	150 kVA
2-Winding Transformer	
Rated Voltage HV	22 kV
Rated Voltage LV	0,24 kV
Phase	3 phase
Frequency	50 Hz subject to fluctuation of ± 5%
Connection HV	Star
Connection LV	Star
Vector group	Yy0

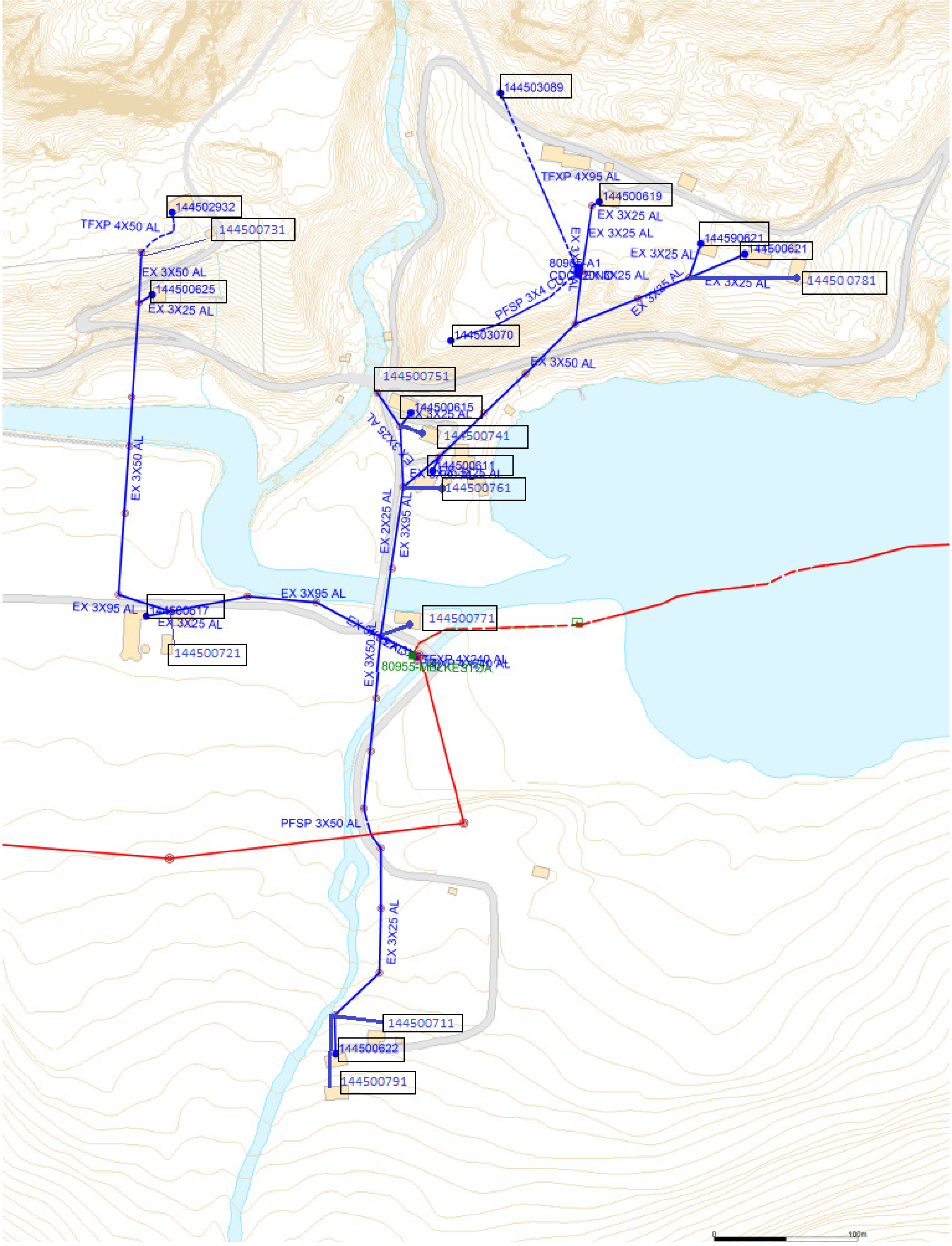

Figure 3.1: Low voltage distribution network in "Sogn og Fjordane" area, where all network subscribers are marked with tags in black boxes. Feeder lines and dimensions are marked with blue and the red line represents the high voltage network supplying the transformer.

In Table 3.2 specifications for each line connected from feeder to the residences are presented. These cables will be a bottleneck for each household at large power consumptions, and feeders will overload if all connected households consume a large amount of power at the same time. The transformer has a limit of 150 kVA, which is the overall constraint. Cable data for the low voltage distribution network can be seen in Appendix A. According to Equation (2.9) load current should be less than the capacity of the cable given in Table 3.2. Since most loads in the residences are two-phase, the total load current is divided by the square root of three when comparing with the three-phase power input.

Table 3.2: *Cable specifications for each residential power consumer in the low voltage distribution network.*

Network Subscriber	Cable	Type	Length [km]	Resistance [Ω]	Reactance [Ω]	Conductivity[A]
144503089	TFXP 4X95 AL	Ground	0,1350	0,0432	0,0101	220
144503070	PFSP 3X50 AL	Ground	0,1030	0,0661	0,0103	150
144502932	TFXP 4X50 AL	Ground	0,0410	0,0263	0,0032	150
144500731	EX 3X25 AL	Overhead	0,0420	0,0266	0,0033	115
144500771	EX 3X25 AL	Overhead	0,0018	0,0216	0,0024	115
114500751	EX 3X25 AL	Overhead	0,0016	0,0192	0,0013	115
144500615	EX 3X25 AL	Overhead	0,0011	0,0156	0,0011	115
144500741	EX 3X25 AL	Overhead	0,0012	0,0144	0,0095	115
144500619	EX 3X25 AL	Overhead	0,0005	0,0072	0,0005	115
144500621	EX 3X25 AL	Overhead	0,0034	0,0504	0,0034	115
144590621	EX 3X25 AL	Overhead	0,0021	0,0300	0,0021	115
144500781	EX 3X25 AL	Overhead	0,0044	0,5304	0,0036	115
144500761	EX 3X25 AL	Overhead	0,0021	0,0300	0,0021	115
144500611	EX 3X25 AL	Overhead	0,0008	0,0120	0,0008	115
144500625	EX 3X25 AL	Overhead	0,0009	0,0132	0,0009	115
144500721	EX 3X25 AL	Overhead	0,0017	0,0227	0,0017	115
144500617	EX 3X25 AL	Overhead	0,0015	0,0216	0,0015	115
144500622	EX 3X25 AL	Overhead	0,0129	0,1884	0,0129	115
144500791	EX 3X25 AL	Overhead	0,0268	0,3216	0,0220	115
144500711	EX 3X25 AL	Overhead	0,0129	0,1884	0,0129	115

3.2 Load Analysis

The loads will be divided into hour-based load data for the households and EV loads separately. The load data for households are divided into power consumptions for four seasons, and are based on estimates found in relevant literature and statistics.

3.2.1 Household Power Consumption

First, the electric energy consumption data is estimated based on research done by “XRGIA” [22] and “Statistics Norway” [26]. Since the energy consumption changes with the seasons and climate for the selected area, four profiles are used which is summer, autumn, winter and spring. Table 3.3 shows a 24-hour period for each profile. By looking at the cross section of the overhead lines supplying the residences, one can assume that these are residences above 149 m². The distance between the residences indicate that this is single unit dwellings, which must be taken into account when estimating the power consumption.

Table 3.3: *Hour-based active power consumption for four seasons, in an average single unit dwelling [22],[26].*

Hour	Summer	Autumn	Winter	Spring
	Active power [kW]	Active power [kW]	Active power [kW]	Active power [kW]
00.00	0,97	1,58	2,77	1,29
01.00	0,97	1,58	2,77	1,29
02.00	0,97	1,58	2,77	1,29
03.00	0,97	1,58	2,77	1,29
04.00	0,97	1,58	2,77	1,29
05.00	0,97	1,58	2,77	1,29
06.00	1,62	2,97	3,24	1,29
07.00	3,24	4,45	4,04	2,16
08.00	3,24	4,45	4,04	2,16
09.00	0,81	1,58	3,24	1,48
10.00	0,86	1,58	3,24	1,48
11.00	0,86	1,58	3,24	1,48
12.00	0,86	1,58	3,24	1,48
13.00	0,86	1,58	3,24	1,48
14.00	0,86	1,58	3,24	1,48
15.00	0,86	2,97	3,24	1,48
16.00	3,88	5,93	6,93	4,71
17.00	3,53	5,93	8,09	4,31
18.00	2,16	5,93	6,93	3,70
19.00	2,16	5,08	6,93	3,24
20.00	2,16	4,45	6,47	2,88
21.00	1,94	3,95	4,85	2,59
22.00	1,62	2,97	4,85	2,35
23.00	1,62	2,97	4,04	2,16
Sum	39	71	100	50

Reference avarage energy consumption [kWh/day]	Summer	Autumn	Winter	Spring
[26]	38,82	71,18	97,06	51,76
[22]	40,00	70,00	103,33	48,33

3.2.2 EV Profile

According to [5], Nissan hold almost 40 percent market share of EVs and PHEVs in Norway. Of the two Nissan models available, Nissan Leaf is the most common. Specifications of this car, which will be used in simulations, is given in Table 3.4 [8]. “Max_bat” is the total battery capacity, “Min_Bat” is the absolute zero of the battery, “Cons_C” is the energy consumption per km for the car and “Dist_C” is the total driving distance on a fully charged battery.

The EV is charged with either 3,3 kW or 6,6 kW , “P_Charger”. “MAX_SOC” and “MIN_SOC” are values for maximum and minimum state of charge. One assume a very large power factor of the charger, which will result in only active power delivered to the battery. The charge and transportation efficiencies, “Charge_eff” and “Trans_eff”, may differ from real scenarios.

Table 3.4: *Nissan Leaf EV specifications, with parameters for simulation purpose [8].*

Parameter	Value
Max_Bat	21 kWh
Min_Bat	0 %
Cons_C	0,12 kWh/Km
Dist_C	175 Km
P_Charger	3,3 / 6,6 kW
MAX_SOC	100 %
MIN_SOC	0 %
Charge_Eff	93,00 %
Trans_Eff	86,50 %

3.2.3 EV User Profiles, Recharge Rate and Share of EVs

To establish the total power consumption caused by charging of EVs, user profiles, an estimated daily recharge rate and share of EVs are defined in this section.

EV User Profiles

Four user profiles are selected below which will determine the connection and disconnection of the EVs. Figure 3.2 shows graphically when the user profiles are connected to the network.

EV_User_1 Normal family with frequently and unpredictable EV usage.
EV connection: 16.00 to 19.00 and 22.00 to 06.00.

EV_User_2 Single user for EV.
EV connection: 16.00 to 07.00.

EV_User_3 Unemployed user.
EV connection: 14.00 to 18.00 and 00.00 to 11.00.

EV_User_4 Night worker.
EV connection: 07.00 to 20.00.

Different probabilities for which EV user connected are set to 50 percent for EV_User_1, 35 percent for EV_User_2, 10 percent for EV_User_3 and 5 percent for EV_User_4.

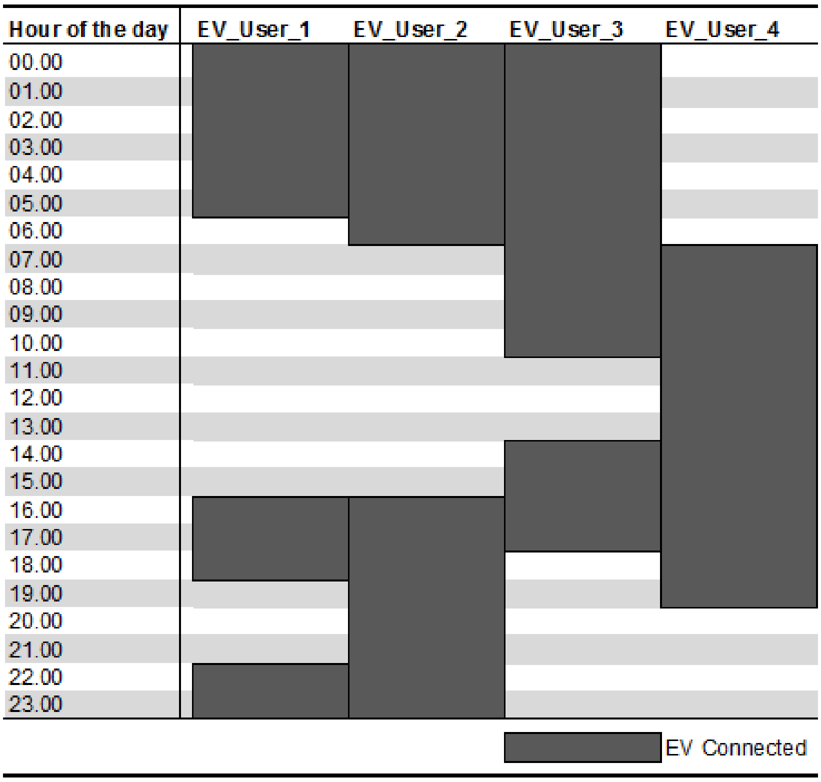

Figure 3.2: *Connection of different user profiles each hour in the distribution network. The grey box indicates connection and duration of connection.*

Recharge rate

Only three percent of single journeys done by car are longer than 80 km in Norway [25]. Using the theoretical energy consumption of a Nissan Leaf given in Table 3.4, the required energy for 97 percent of the journeys are less than 9,6 kWh [8]. Due to external conditions, this value is unrealistic. In a worst-case scenario, the 80 km distance requires 15 kWh. According to Statistics Norway, the average driving distance of a passenger car is 37 km per day [23]. Theoretical required energy for this distance is 4.4 kWh, and 7 kWh in a worst-case scenario [8]. The worst-case scenario occur at low ambient temperature or when using the EVs in varying terrain, and will be utilized in simulations for this project.

Share of EVs

In Norway, the number of EVs have more than doubled every year since 2012. At the same time the total number of passenger cars increased with only 2,3 percent annually [24]. At the end of 2014, the share of EVs and PHEVs among passenger cars were approximately 1.8 percent. One can expect that this share will be closer to 7,3 percent in 2020 if as many EVs and PHEVs are sold as in 2014. Figure 3.3 shows the increase of EVs and PHEVs together with a prediction of future numbers [5],[24]. One assumes that sales of new EVs and PHEVs will stabilize through the next five years and the total increase of passenger cars will stay the same.

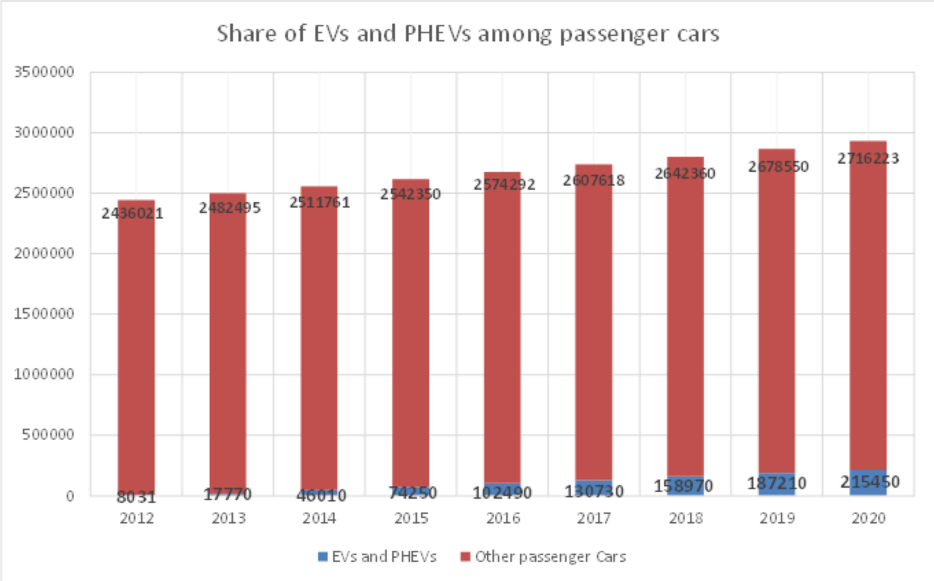

Figure 3.3: *Share of EVs and PHEVs among passenger cars in Norway using stacked diagram where blue represents EVs and PHEVs and red represents other passenger cars. The exact numbers are listed in front of the columns.*

3.3 Distribution System Constraints

There are constraints in conjunction with the smart load management system with respect to overloading and regulations given in Section 2.4.2. Equation (3.1) sets the upper and lower voltage level. Equation (3.2) limits the current flowing through the distribution lines based on Equation (2.9). The limits are dependent of the cable conductivity of each line and feeder in the network given in Appendix A. Equation (3.3) limits the active power of the system due to the transformer capacity, where the apparent power of the transformer is given in Table 3.1. If DSM algorithms are calculated, Equation (3.4) defines the limits.

$$V_{min} \leq V_n \leq V_{max} \quad [V] \tag{3.1}$$

Where

V_{min} : 207
V_n : 230
V_{max} : 253

$$I_b \leq I_z \quad [A] \tag{3.2}$$

Where

I_b: Load current
I_z: Cable conductivity

$$\sum_0^n P_{Load} \leq P_{Transformer} \quad [W] \tag{3.3}$$

Where

$\sum_0^n P_{Load}$: Sum of all power consumer loads
$P_{Transformer}$: Transformer capacity
$P_{Transformer}$: S * φ_2

S : Apparent power [VA]
φ_2: Phase angle

$$\sum_0^n P_{Load\ residence} + P_{Charger} \leq Demand\ limit\ [W] \quad (3.4)$$

Where
$\sum_0^n P_{Load\_residence}$: Sum of all active residence loads
$P_{Charger}$: EV charging power

3.4 Management Programmes

In this section, algorithms for each objective given in Section 1.3 will be explained along with flowcharts.

3.4.1 Objective 1

To achieve objective one, a scenario with data given in Section 3.2 are simulated without a load management system. The constraints given in Section 3.3 will have no impact on the system. To make the test credible the MATLAB algorithm is randomly selecting where the EVs are located in the network and when the charging takes place. The flowchart of the uncoordinated charging algorithm can be seen in Figure 3.4.

The total number of EVs and season of the year is manually selected as an input in the algorithm. As given in Figure 3.3, the share of EVs and PHEVs among passenger cars are 2,8 percent currently. To see if this model has any impact on the low voltage distribution network two EVs are used in the simulations. The EV user profiles given in Section 3.2.3, and placement of the EVs in the distribution network are computed with MATLAB functions, whilst the algorithm calculates the load data in a 24-hour period.

As mentioned in Section 3.2.3, the average driving distance in Norway is 37 km, which results in a SOC of 66 percent, assuming the battery was fully charged before use. Since the algorithm is hour-based, intervals of 3,3 kW are selected to determine the charging time of the EV. If the EV SOC is less than 84 percent, which means less than 17,7 kWh the EV will be placed in an interval. The results are transferred to DIgSILENT PowerFactory for analysis. Please find the MATLAB script computing the uncoordinated charging in Appendix B.

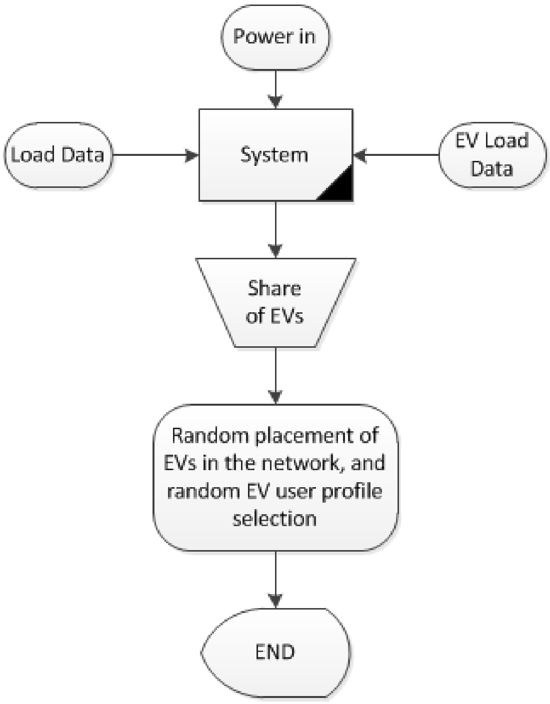

Figure 3.4: Flowchart of uncoordinated charging, objective 1.
As an input, the model receives load data, EV load data and power from the transformer. The algorithm places a selected number of EVs randomly in the network, and selects EV load profiles for each residence.

3.4.2 Objective 2

Objective 2 is achieved through an SLM algorithm based on the first objective and data given in Section 3.2.

The structure of the SLM can be seen in Figure 3.5 given as a flowchart. As input, the system will have a given amount of power available, the hour-based load data and stipulated EV data. The algorithm is a continuation of the algorithm in objective 1, where profiles and placement of the EVs are decided. First, the SOC is determined, which decides if the EV should be scheduled for charging. The algorithm monitors the constraints given by Equation (3.1), (3.2), and (3.3) and reschedules EV charging in the case of any contraventions. Similar to the first objective, the results from the algorithm are transferred to DIgSILENT PowerFactory for analysis and verification. Please find the MATLAB script computing the coordinated charging in Appendix C. Note that this script must be used together with the script for objective 1.

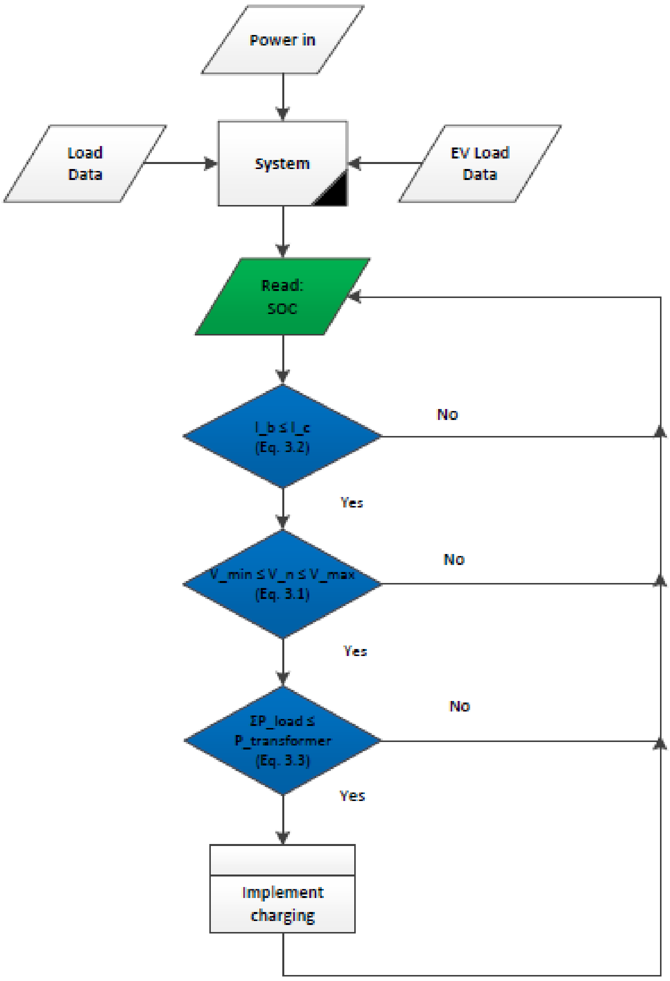

Figure 3.5: Flowchart of SLM algorithm, objective 2. *The system receives external data which is controlled through the following loop. The SOC of each connected vehicle are determined and the constraints of the system must be met before the vehicle is scheduled for charging.*

3.4.3 Objective 3

Objective 3 is achieved by simulating a larger share of EVs in the same distribution network. As an additional feature, the users should also be able to charge their cars immediately in case of an extraordinary situation. A “fast charge button” is implemented and gives the specified user priority. When there are no EVs with priority, the algorithm will try to minimize system power losses and the EVs that will cause the lowest power losses when charging will be scheduled first.

Unlike the two first scripts, eight chosen EVs are given a residence tag in the distribution network. The load calculations will be carried out for both hour-based and minute-based scenarios. The minute-based load data origin from the original data given in Table 3.3, but with more variations. Please find the MATLAB script computing the coordinated charging in Appendix D. A flowchart for the algorithm can be seen in Figure 3.6.

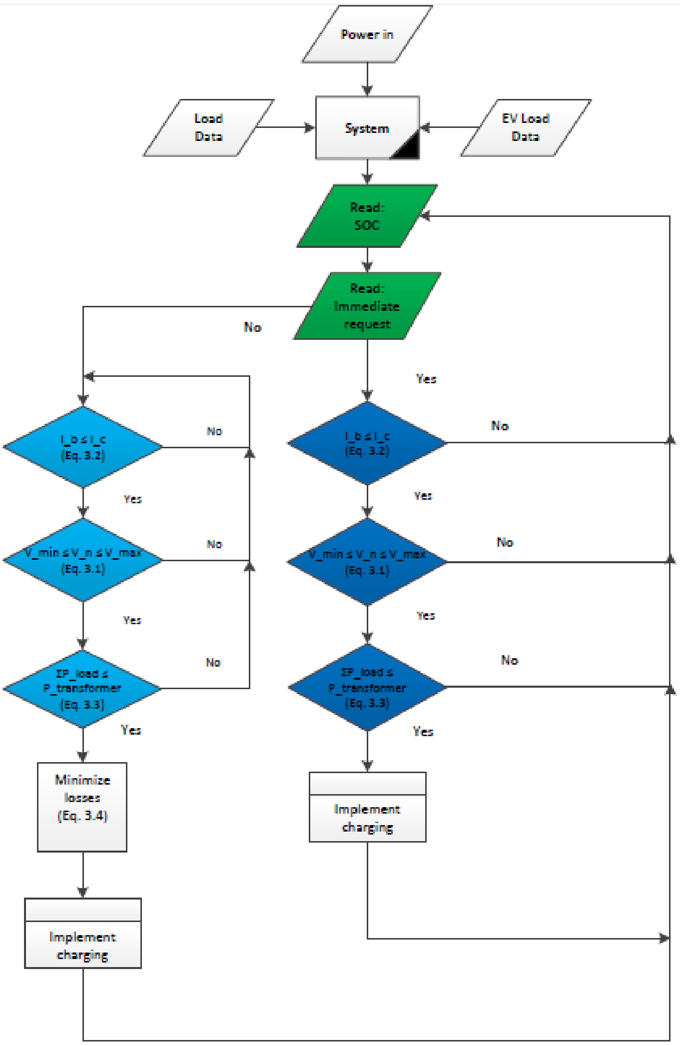

Figure 3.6: Flowchart of SLM algorithm, objective 3. *The system receives external data which is controlled through the following loop. First, the SOC of each connected vehicle are determined. Secondly, the algorithm searches for immediate charge requests. Last, the constraints of the system must be met before the vehicle is scheduled for charging.*

3.4.4 Objective 4

This objective is achieved using DSM and will be divided into two scenarios. In the first scenario, both household loads and EV loads are shifted. In the second scenario only EV loads are shifted.

First Scenario

A fictional demand limit profile is given in Table 3.5, which gives the residential power consumers the ability to select which loads they will use within the limited power capacity provided by a distribution network operator. The demand limit creates changes in the load shape while the residential power consumer still has the freedom to choose the loads. Power intensive loads, such as heating and cookers, may be turned off if charging of an EV shall take place at the same time. The flowchart for the first DSM scenario is given in Figure 3.7.

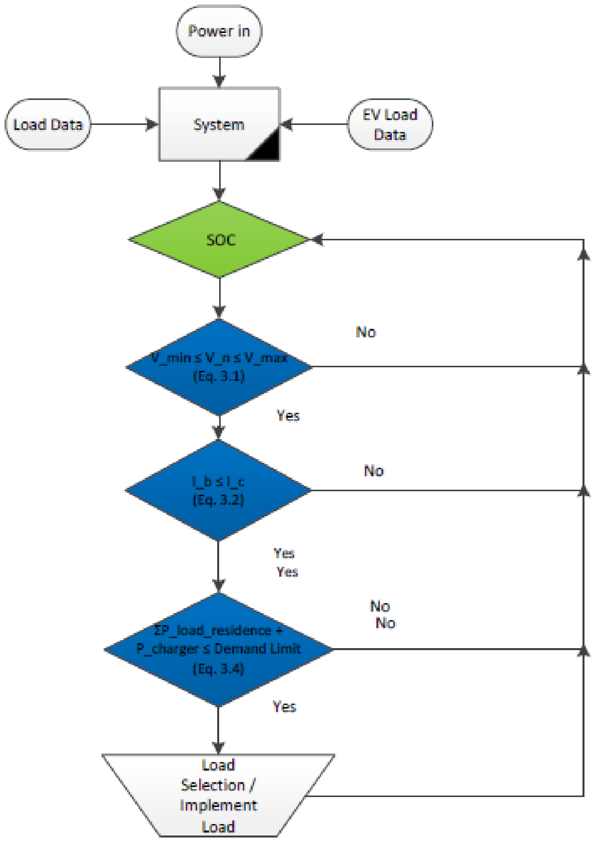

Figure 3.7: Flowchart of the first DSM scenario, objective 4. *The system receives external data which is controlled through the following loop. First, the SOC of each connected vehicle are determined. Secondly, the constraints in the network must be met before implementing loads.*

The demand limit profile can be seen graphically in Figure 3.8. The power rating is low when the industrial sector is active, and vice versa. The upper limit of 6,75 kW per consumer equals the transformer maximum power capacity of the given low voltage distribution network with a power factor of 0,9.

Table 3.5: *Hour-based demand limit profile with maximum power consumption at each residential power consumer at given periods.*

Period [Hour]	Demand limit [kW]
00.00 - 05.00	6,75
05.00 - 07.00	5
07.00 - 11.00	4
11.00 - 12.00	5
12.00 - 16.00	4
16.00 - 19.00	6
19.00 - 00.00	6,5

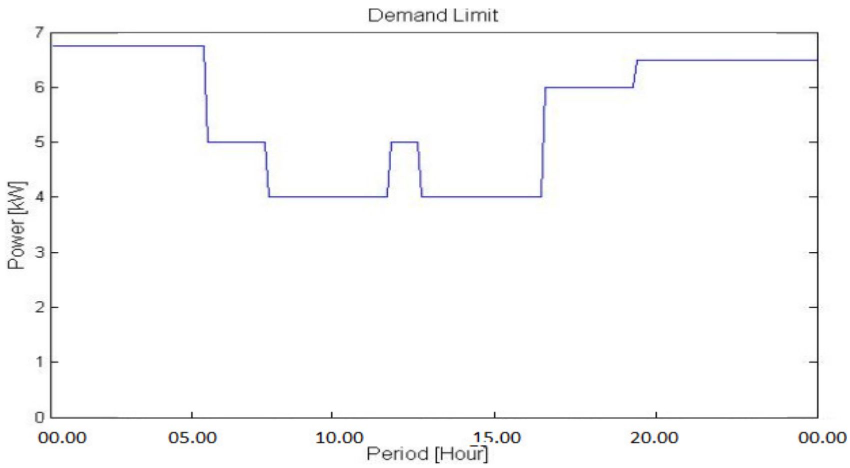

Figure 3.8: *Demand limits provided by the distribution network operator, where the x-axis represents the hour of the day and the y-axis the total power consumption allowed.*

Table 3.3, which is the sum of all the loads in the household, will be divided into critical, heating and miscellaneous loads. The EV specifications will be the same as in the previous objectives. Critical loads are loads that shall not be disconnected, such as lighting and refrigerators. Heating is all electric heating including the hot water heating system and is disconnected if necessary. Miscellaneous loads are the remaining components, such as chargers and small electric equipment. The data used in the simulation is for winter season and can be seen in Table 3.6. The algorithm for the first scenario can be found in Appendix E.

Table 3.6: *Hour- based critical, heating and miscellaneous loads used in DSM simulation.*

Hour	Total load [kW]	Critical [kW]	Heating [kW]	Miscellaneous [kW]
00.00	2,77	0,97	1,39	0,42
01.00	2,77	0,97	1,39	0,42
02.00	2,77	0,97	1,39	0,42
03.00	2,77	0,97	1,39	0,42
04.00	2,77	0,97	1,39	0,42
05.00	2,77	0,97	1,39	0,42
06.00	3,24	1,13	1,62	0,49
07.00	4,04	1,42	2,02	0,61
08.00	4,04	1,42	2,02	0,61
09.00	3,24	1,13	1,62	0,49
10.00	3,24	1,13	1,62	0,49
11.00	3,24	1,13	1,62	0,49
12.00	3,24	1,13	1,62	0,49
13.00	3,24	1,13	1,62	0,49
14.00	3,24	1,13	1,62	0,49
15.00	3,24	1,13	1,62	0,49
16.00	6,93	2,43	3,47	1,04
17.00	8,09	2,83	4,04	1,21
18.00	6,93	2,43	3,47	1,04
19.00	6,93	2,43	3,47	1,04
20.00	6,47	2,26	3,24	0,97
21.00	4,85	1,70	2,43	0,73
22.00	4,85	1,70	2,43	0,73
23.00	4,04	1,42	2,02	0,61

Second Scenario

In the second scenario, the DSM algorithm will be executed without shifting the household loads and with minute-based time calculations. The demand limit of a 24-hour period will be divided in 15 minutes steps where only charging of the EV is shifted if necessary. Due to the large amount of data, the loads and demand limits for winter season are shown graphically in Figure 3.10. The minute-based load data and demand limits are based on the original data given in Table 3.3 and Table 3.5. The flowchart for the first DSM scenario is given in Figure 3.9. Please find the MATLAB script computing the DSM second scenario in Appendix F.

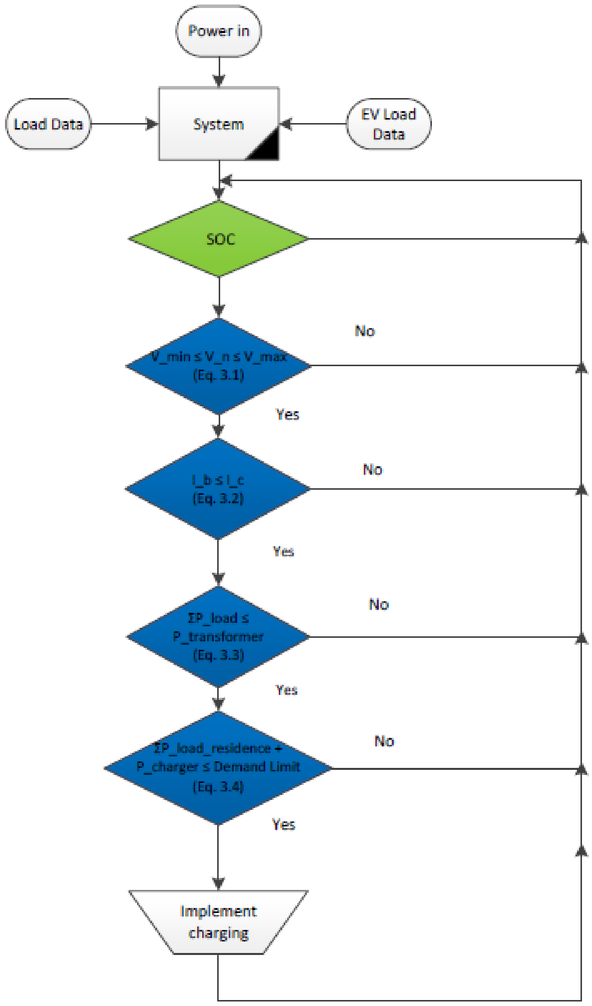

Figure 3.9: Flowchart of the second DSM scenario, objective 4.
The system receives external data which is controlled through the following loop. First, the SOC of each connected vehicle are determined. Secondly, the constraints in the network must be met before executing EV charging.

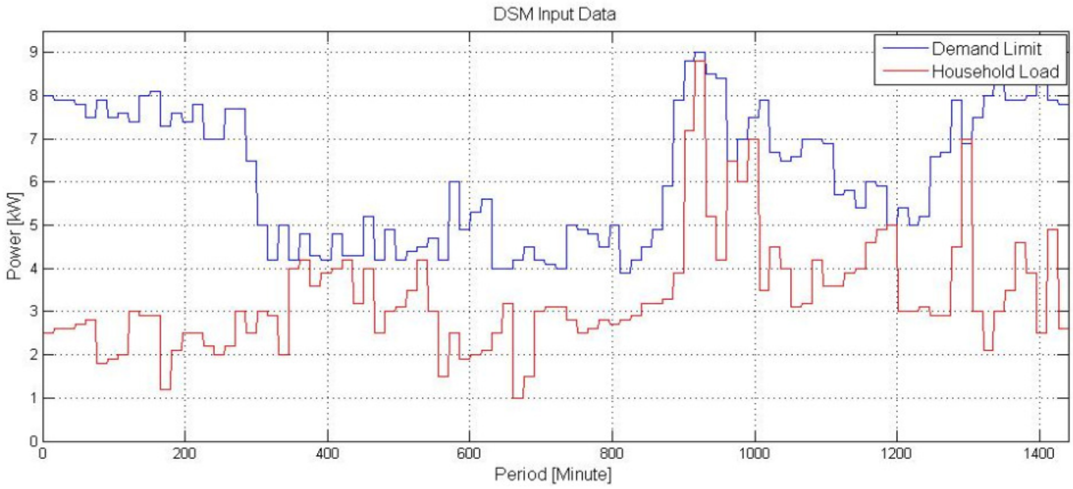

Figure 3.10: *Minute-based load data and demand limit for DSM calculations. The x-axis represents the period, and y-axis power.*

3.5 Test and Validation of Algorithms

This section will give a further explanation of the algorithm parameters inputs and outputs.

3.5.1 Uncoordinated Charging – Objective 1

The EV user profiles are selected through the "datasample" function. The input data is a vector with the four user profiles given in Section 3.2.3. "Datasample" returns the same number of observations randomly. The placement of the EVs are set by "unidrnd", which is a discrete uniform random number function. This function generates random numbers, which are linked to the respective residence nine-digit tag. The SOC is set to 67 percent which places the EVs in a vector interval. This determines the preferred charging pattern of each EV depending on the EV user profile. Last, the stipulated load data given in Table 3.3 are added up with the EV power demand for the given residence.

Table 3.7 shows the input and output data from the script that are transferred to Chapter 4. The load data will be the same as the data given in Table 3.3, except for the intervals where the EV is connected. In this period, 3.3 kW are added which is the charging power. The charging lasts for three hours since the SOC is initially 6,67 percent.

Table 3.7: *Input and output data computed from the algorithm in objective 1. The season, SOC and charging power is manually selected. The algorithm places the EVs randomly in the network with a given user profile.*

Input	
Season Profile:	Winter
SOC:	0,67
Charging power:	3.3 kW
Output:	
Residence with first EV:	144500771
Residence with second EV:	144500619
EV User Profile no. 1	EV_User_2
EV User Profile no. 2	EV_User_3

3.5.2 The First SLM Algorithm – Objective 2

This SLM algorithm uses the data from objective 1 and reschedules the charging if necessary. To monitor the constraints the algorithm calculates 24 hours of load currents and voltages at each residential power consumer using Equation (2.5) and (2.6). The currents of the residences with EVs that were chosen in objective 1 are listed in Table 3.8.

Table 3.8: *Calculated currents at the network subscribers with EV.*

Hour	144500771 Current [A]	144500619 Current [A]
00.00	16,93	16,93
01.00	16,93	16,93
02.00	16,93	16,93
03.00	7,73	7,73
04.00	7,73	7,73
05.00	7,73	7,73
06.00	9,04	9,04
07.00	11,27	11,27
08.00	9,04	9,04
09.00	9,04	9,04
10.00	9,04	9,04
11.00	9,04	9,04
12.00	9,04	9,04
13.00	9,04	9,04
14.00	9,04	9,04
15.00	19,33	19,33
16.00	31,77	31,77
17.00	28,53	28,53
18.00	28,53	28,53
19.00	18,05	18,05
20.00	15,03	15,03
21.00	13,53	13,53
22.00	13,53	13,53
23.00	11,27	11,27

After calculating all data for the system, the algorithm runs through a loop for each residence to monitor voltage, current and power limitations using Equation (3.1), (3.2) and (3.3). A section of the MATLAB script can be seen in Figure 3.11 where a "while" loop is used to limit the calculations to 24 hours.

```
while N<25
    % ONE/ONEONE represents if the residence have an EV connected
    if  ONE==1 & EV_User_1(N)==1
    USE_144503089=1 ;                    % Residence with active EV
    elseif  ONEONE==1 & EV_User_2(N)==1
    USE_144503089=1;                     % Residence with active EV
    else USE_144503089=0;                % Residence with no EV or inactive EV
    end
    % Line, feeder, voltage, transformer constraints:
    if I_b_144503089(N) <= I_z_144503089 & I_b_80955_A1(N) <= I_z_80955_A1
        & I_b_80955_23(N) <= I_z_80955_23 & I_b_80955_22(N) <= I_z_80955_22
        & I_b_80955_19(N) <= I_z_80955_19 & I_b_80955_16(N) <= I_z_80955_16
        & I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0
        & U_144503089(N) >= v_min & U_144503089(N) <= v_max
        & P_load_tot(N) <= P_transformer & USE_144503089==1
    EX_144503089 = 1;                    % Execute charging this hour
    else
    EX_144503089 = 0;                    % No charging due to constraints
    end
    Ma_144503089(N) = EX_144503089;  % Puts values in a matrix
    N = N+1;
end
```

Figure 3.11: *Section of MATLAB script computing coordinated charging. A "while" loop limits the calculations to 24 hours.*

After running the algorithm, Table 3.9 shows the desired charging hours of the chosen residences compared to the charging hours allowed by the algorithm. Zero represents no charging and one represents charging. As seen in the table, the time between the fifteenth hour and the nineteenth hour are critical and often called peak hours.

Table 3.9: *Desired charging hours of the network subscriber are compared with the charging hours allowed by the SLM algorithm. 1 represents charging and 0 no charging.*

144500771 desired charging hours												
1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0			
144500771 allowed charging hours												
1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0			
144500619 desired charging hours												
1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0			
144500619 allowed charging hours												
1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0			

3.5.3 The Second SLM Algorithm – Objective 3

Like the two first objectives, the household power consumption in Table 3.3 are added up with the possible power consumption from the EVs. In this scenario, the eight first network subscribers in Table 3.2 are chosen to have one EV each. The first four users have user profile number 1, the fifth and sixth have profile number 2, seventh number 3 and eight number 4. In this scenario, the charging power and season of the year are set to 3.3 kW and winter. This is selected in the input section of the script along with the user priority. There are also a constraint and minimization bypass, which allows the algorithm to run without monitoring constraints or minimize line power loss.

The total impedances in the distribution system are calculated together with initial voltages, currents and power losses. Table 3.4 are used to determine requested charging time of the EVs when the SOC is given. When the total charging time is settled, the EV is part of a calculation “while” loop which runs for a 24 hour period. If a given EV is charged for one hour the charging time decreases with one hour. If there are any constraints or limitations which delays the charging, the charging time stays the same until the EV is scheduled for charging.

The total power of the system is added up in a matrix using the desired charging patterns. Using Equation (2.5) and (2.6), voltage and current at each residence are calculated. Total feeder currents are calculated together with power loss of every residence using Equation (2.10). The algorithm will monitor voltage, current and power limitations using Equation (3.1), (3.2) and (3.3). With no constraint contraventions, the EVs with priority will be scheduled for charging. The EVs with no priority will be scheduled for charging if there are capacity in the network. If the total power of the network is above 80 percent of the rated transformer capacity, only the three EVs that will cause the lowest line power loss will be charged. The criteria for loss minimization can be chosen for each scenario.

Some key results from algorithm can be seen in Table 3.10, when there are no residences with priority. The table shows allowed charging hours for the EVs. One represents the allowed charging and zero no charging. If zero occur, there are capacity limitations in the network. If the algorithm is executed with a minute-based time period, the output will show more detailed charging opportunities. Further explanation will be given in Section 4.2.2.

Table 3.10: *Allowed charging of the EVs in the low voltage distribution network. The residences with EVs are presented with the nine-digit tag. One represents the possibility for charging and zero no charging at the given hour.*

	Residence:							
Hour:	144503089	144503070	144502932	144500731	144500771	114500751	144500615	144500741
00.00	1	1	1	1	1	1	1	1
01.00	1	1	1	1	1	1	1	1
02.00	1	1	1	1	1	1	1	1
03.00	1	1	1	1	1	1	1	1
04.00	1	1	1	1	1	1	1	1
05.00	1	1	1	1	1	1	1	1
06.00	1	1	1	1	1	1	1	1
07.00	1	1	1	1	1	1	1	1
08.00	1	1	1	1	1	1	1	1
09.00	1	1	1	1	1	1	1	1
10.00	1	1	1	1	1	1	1	1
11.00	1	1	1	1	1	1	1	1
12.00	1	1	1	1	1	1	1	1
13.00	1	1	1	1	1	1	1	1
14.00	1	1	1	1	1	1	1	1
15.00	1	1	1	1	1	1	1	1
16.00	0	0	0	0	0	0	0	0
17.00	0	0	0	0	0	0	0	0
18.00	0	0	0	0	0	0	0	0
19.00	0	0	0	0	0	0	0	0
20.00	0	0	0	0	1	0	0	0
21.00	1	1	1	1	1	1	1	1
22.00	1	1	1	1	1	1	1	1
23.00	1	1	1	1	1	1	1	1

3.5.4 Demand Side Management – Objective 4

This objective is based on Equation (3.4), where the algorithm searches for the best possible solution that benefits both the network subscriber and the distribution network operator. Since the DSM algorithm is equal for every residence in the given low voltage distribution network, only one chosen residence will be part of the simulation. The constraints in Section 3.3 applies for the system except Equation (3.3), since the power limitation is given by the demand limit.

DSM First Scenario

In the first scenario, the DSM algorithm proposes a solution that enables charging of the EV at the expense of other residential loads. The first EV user profile given in Section 3.2.3, and residence tag 144502932 are used in the simulation. Comparing the demand limits and the residential loads in wintertime there are few possibilities for charging an EV, but when mixing the critical, heating and miscellaneous loads defined in Table 3.6, there are several combinations that enables charging. The critical loads will always be the base load, but heating and miscellaneous loads can be disconnected in periods where the EV is connected. The algorithm will try to maximize the load within the demand limit and one assume that it is desirable to shift the requested loads that are disconnected due to the demand limit. For simulating other seasons the critical loads are set to 35 percent, heating loads to 50 percent and the miscellaneous loads to 15 percent of the total load given in Table 3.3.

The current and voltage constraints for the residence are monitored for the largest value of the demand limit. If there are contraventions with this value the demand limit needs to be lowered. Further, the initial time to fully charge the EV is calculated, and initial voltages and loads are given. The DSM logic consists of a “while” loop which selects the different loads in the specific order given in Table 3.11. The number in the left column represents the priority, where 1 is the highest priority and 16 the lowest. In addition, the algorithm prioritize heat loads if the EV has been charged at the expense of heat loads the previous hour. This is represented with the number 0. Ordinary loads are loads that should be executed that specific hour originally. Shifted loads are loads that are not executed according to the original schedule due to the demand limit. After the load priority selection, the loads are maximized within the demand limit. This means a share of heating loads or miscellaneous loads are included in the final total load and the other share is shifted to the next hour.

Table 3.11: *Differentiation of loads in DSM simulation with given priority in the left column. Shifted loads are loads from previous hours and ordinary loads are loads that are scheduled for that specific hour. The critical load will always be included.*

Load Priority	Type of loads
0	Critial, ordinary heating and shifted heating loads
1	All ordinary loads, all shifted loads and EV charging
2	All ordinary loads, shifted heating loads and EV charging
3	All ordinary loads and EV charging
4	Critial, ordinary heating, shifted heating loads and EV charging
5	Critial, ordinary heating loads and EV charging
6	Critical, ordinary misc., shifted misc. loads and EV charging
7	Critical, ordinary misc. loads and EV charging
8	Critial loads and EV charging
9	All ordinary loads and all shifted loads
10	All ordinary loads and shifted heating loads
11	All ordinary loads
12	Critial, ordinary heating and shifted heating loads
13	Critial and ordinary heating loads
14	Critical, ordinary misc. and shifted misc. loads
15	Critical and ordinary misc. loads
16	Critical loads

The inputs and results of this scenario can be seen in Table 3.12. Specifications of which loads that are executed, load priority, can be read out of Table 3.11. The connection periods of the EV is shown graphically to show where it is possible to charge the EV. As programmed, the load never exceeds the demand limit. At the first peak hour load priority 8 is executed which enables one hour of EV charging, before heating is prioritized again due to household comfort. The loads that are not executed are shifted and rescheduled for the next available hour.

Table 3.12: *Input and hour-based results of DSM simulation, where the loads executed are specified in Table 3.11.*

Input	
Season Profile:	Winter
SOC:	66,66 %
Connection EV:	EV_User_1
Charging Power:	3,3 kW

Results

Load Priority	Hour	Demand limit [kW]	Connection EV
1	00.00	6,75	
1	01.00	6,75	
1	02.00	6,75	
9	03.00	6,75	
9	04.00	6,75	
9	05.00	5,00	
9	06.00	5,00	
12	07.00	4,00	
12	08.00	4,00	
9	09.00	4,00	
9	10.00	5,00	
9	11.00	4,00	
9	12.00	4,00	
9	13.00	4,00	
9	14.00	4,00	
9	15.00	4,00	
8	16.00	6,00	
0	17.00	6,00	
0	18.00	6,00	
16	19.00	6,50	
16	20.00	6,50	
11	21.00	6,50	
7	22.00	6,50	
0	23.00	6,50	

DSM Second Scenario

In this algorithm, the same residence tag and EV user profile applies, together with the input given in Figure 3.10. If the EV is connected and not fully charged, the charging power of the EV is added and compared with the demand limit. A "while" loop calculates residence voltages and currents every minute for 24 hours and the algorithm schedules the EV for charging if Equation (3.1), (3.2) and (3.4) are fulfilled. Similar to previous objectives the EV charging is postponed at any constraint contraventions.

Figure 3.12 shows the connection schedule, where the red line represents when the EV is connected and the blue line when charging occur after running the DSM algorithm. The plot shows that the EV is charged at both off peak and peak hours.

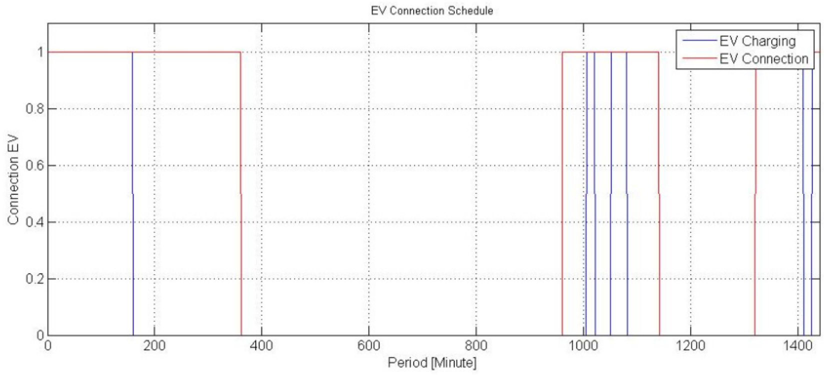

Figure 3.12: *DSM connection schedule, where the red line represents when the EV is connected and the blue line when the charging occur. The x-axis represents the period and y-axis connection and charging.*

3.6 DIgSILENT PowerFactory Description

The DIgSILENT PowerFactory software is applicable to the modelling of generation-, transmission-, distribution- and industrial grids [27]. In this project, the software is used for analysis and verification of the data computed by the SLM and DSM algorithms. The software calculates load flow, short circuit current, harmonics, power quality, reliability among other electric data interesting in the low voltage distribution network.

The low voltage distribution network consists of a transformer, breakers, feeders, overhead lines, underground cables, busbars and loads. The loads are arranged as seen in Figure 3.13, where the residence load is represented with the nine-digit tag and a capital L. The EV load is represented as a secondary load with the same nine-digit tag and a capital EV. Every residence and EV load are connected to a common busbar, which displays the short circuit current at the network subscriber. The lines, cables and feeders are shown with a unique tag. The transformer are connected to two main feeders, which supplies 12 and 8 network subscribers each. The data for the specific cable, line or feeder can be found in Appendix A. The complete network can be seen in Figure 3.14.

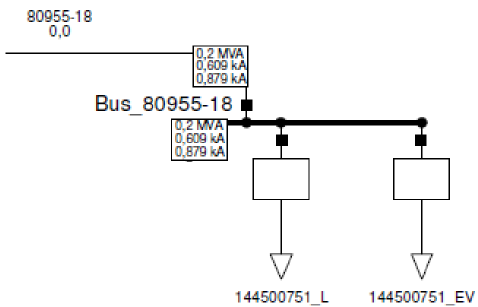

Figure 3.13: *Modeled residence load and EV load in PowerFactory. The information boxes contains short circuit power, initial short circuit current and peak short circuit current.*

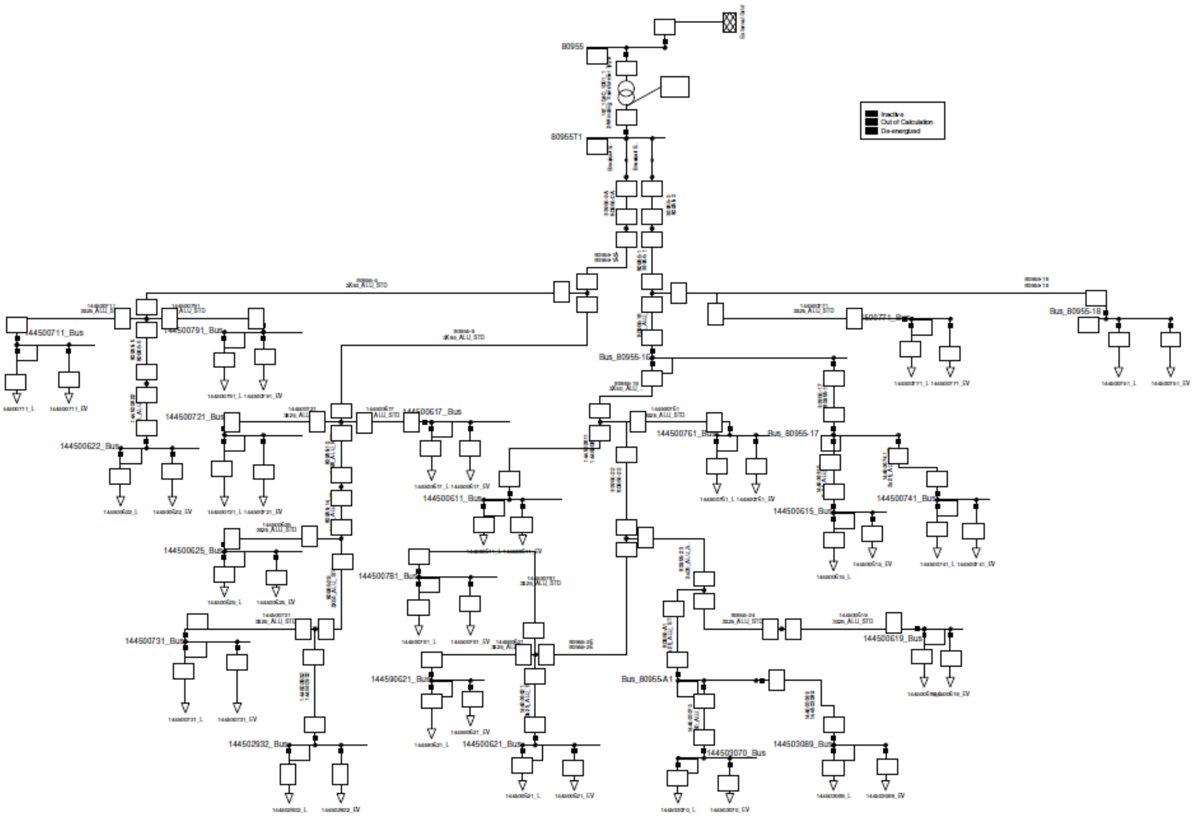

Figure 3.14: *PowerFactory model where the loads are represented as arrows. The EV loads are placed as secondary loads next to the residence loads.*

By using a common busbar for every residence, PowerFactory calculates the short circuit currents listed in Table 3.13 below. These values can also be found using Equation (2.8). A large short circuit value indicates low resistance and therefore a low voltage drop in the distribution lines.

Table 3.13: *Short circuit currents at each network subscriber.*

Network Subscriber	Short Circuit Current [kA]
144503089	0,554
144503070	0,509
144502932	0,534
144500731	0,490
144500771	5,918
114500751	0,609
144500615	1,117
144500741	1,257
144500619	0,524
144500621	0,446
144590621	0,447
144500781	0,521
144500761	0,797
144500611	1,607
144500625	0,620
144500721	1,950
144500617	1,545
144500622	0,461
144500791	1,101
144500711	1,266

4 Results

In this chapter, the results from the algorithms will be validated in DIgSILENT PowerFactory. Simulations of different seasons of the year will be established together with the data determined in Chapter 3. Both tables and MATLAB plots will be presented to show the role of the management programmes.

4.1 Uncoordinated Charging Results

In this section, the uncoordinated charging results for all seasons of the year are presented. The uncoordinated charging results given by objective 1 is validated, as well as uncoordinated charging results from simulating objective 3 without the SLM algorithm.

4.1.1 Uncoordinated Charging Results – Objective 1

By running the MATLAB algorithm for objective 1, load data is computed for the residences with an EV connected and given in Table 4.1 below. The loads are from simulating a winter scenario, where the remaining residence loads can be found in Table 3.3. Note that the simultaneity factor is not taken into account, which means that some of the residences could be unoccupied in real time. The charging power is 3,3 kW for all EVs in the network.

Table 4.1: *Load results for winter scenario uncoordinated charging, objective 1.*

Load Profile of 144500771:				Load Profile of 144500619:			
Hour	Power [kW]	Hour	Power [kW]	Hour	Power [kW]	Hour	Power [kW]
00.00	6,07	12.00	3,24	00.00	6,07	12.00	3,24
01.00	6,07	13.00	3,24	01.00	6,07	13.00	3,24
02.00	6,07	14.00	3,24	02.00	6,07	14.00	3,24
03.00	2,77	15.00	3,24	03.00	2,77	15.00	6,54
04.00	2,77	16.00	10,23	04.00	2,77	16.00	10,23
05.00	2,77	17.00	11,39	05.00	2,77	17.00	11,39
06.00	3,24	18.00	10,23	06.00	3,24	18.00	6,93
07.00	4,04	19.00	6,93	07.00	4,04	19.00	6,93
08.00	3,24	20.00	6,47	08.00	3,24	20.00	6,47
09.00	3,24	21.00	4,85	09.00	3,24	21.00	4,85
10.00	3,24	22.00	4,85	10.00	3,24	22.00	4,85
11.00	3,24	23.00	4,04	11.00	3,24	23.00	4,04

The key results from the DIgSILENT PowerFactory validation are given in Table 4.2. The simulation has been executed for all four seasons, the sixteenth hour of the day. As shown in the table, the transformer and one of the feeders is overloaded in the winter scenario. In the rest of the scenarios, there are no transformer or cable violations, but the PowerFactory software indicates under voltage at some of the residences.

Table 4.2: *DIgSILENT PowerFactory validation results for objective 1.*

Season Hour	Winter 16.00-17.00	Spring 16.00-17.00	Summer 16.00-17.00	Autumn 16.00-17.00
Transformer Loading [%]	112,40	73,80	58,10	94,40
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	1 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	101,10	64,00	46,30	82,90
Lowest voltage at residence [V]	181,00	199,00	210,00	187,00

4.1.2 Uncoordinated Charging Results - Objective 3

To see the impact of the SLM algorithm, objective 3 is simulated without coordinated charging. The load results for a winter scenario, which is the most power-demanding season of the year, are given in Table 4.3. The residences with an EV connected are displayed together with the hour-based load data. Similar to Section 4.1.1 the rest of the residence loads comply with Table 3.3.

Table 4.3: *Load results for winter scenario uncoordinated charging of objective 3.*

	Residence load [kW]							
Hour	144503089	144503070	144502932	144500731	144500771	114500751	144500615	144500741
00.00	6,07	6,07	6,07	6,07	2,77	2,77	6,07	2,77
01.00	2,77	2,77	2,77	2,77	2,77	2,77	6,07	2,77
02.00	2,77	2,77	2,77	2,77	2,77	2,77	6,07	2,77
03.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
04.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
05.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
06.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
07.00	4,04	4,04	4,04	4,04	4,04	4,04	4,04	7,34
08.00	4,04	4,04	4,04	4,04	4,04	4,04	4,04	7,34
09.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	6,54
10.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
11.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
12.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
13.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
14.00	3,24	3,24	3,24	3,24	3,24	3,24	6,54	3,24
15.00	3,24	3,24	3,24	3,24	3,24	3,24	6,54	3,24
16.00	10,23	10,23	10,23	10,23	10,23	10,23	10,23	6,93
17.00	11,39	11,39	11,39	11,39	11,39	11,39	8,09	8,09
18.00	10,23	10,23	10,23	10,23	10,23	10,23	6,93	6,93
19.00	6,93	6,93	6,93	6,93	6,93	6,93	6,93	6,93
20.00	6,47	6,47	6,47	6,47	6,47	6,47	6,47	6,47
21.00	4,85	4,85	4,85	4,85	4,85	4,85	4,85	4,85
22.00	8,15	8,15	8,15	8,15	4,85	4,85	4,85	4,85
23.00	7,34	7,34	7,34	7,34	4,04	4,04	4,04	4,04

A morning scenario, one of the peak hours and an evening scenario are transferred to DIgSILENT PowerFactory for validation. The simulations are executed for all four seasons and the results are given in Table 4.4. The peak hours are the most power demanding hours of the day, which results in under voltage in all of the scenarios. The transformer is overloaded in both winter and autumn, but overloading of one of the feeders only occur in winter season. The morning scenario is the only scenario with no under voltage violations.

***Table 4.4:** DIgSILENT PowerFactory validation results for uncoordinated charging of objective 3. The four season are presented at three different hours of the day.*

Season	Winter	Spring	Summer	Autumn
Hour	07.00-08.00	07.00-08.00	07.00-08.00	07.00-08.00
Transformer Loading [%]	58,60	33,50	45,40	61,50
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	44,90	25,30	36,50	49,10
Lowest voltage at residence [V]	211,00	223,00	216,00	208,00
Hour	16.00-17.00	16.00-17.00	16.00-17.00	16.00-17.00
Transformer Loading [%]	127,90	87,60	67,10	109,00
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	1 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	108,60	70,70	51,80	90,50
Lowest voltage at residence [V]	170,00	194,00	207,00	182,00
Hour	22.00-23.00	22.00-23.00	22.00-23.00	22.00-23.00
Transformer Loading [%]	82,50	83,30	40,10	48,40
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	72,50	71,70	34,20	40,90
Lowest voltage at residence [V]	194,00	199,00	217,00	214,00

4.2 SLM Results

In this section, the results of the suggested SLM algorithms will be presented together with the validation executed in DIgSILENT PowerFactory.

4.2.1 SLM Results Objective 2

With the SLM algorithm given in Section 3.4.2, charging at peak hours are not allowed in the winter scenario. The validation of the results of the sixteenth hour of the day will therefore be carried out with no EVs in the network. For the three other seasons, charging may take place at the peak hours due to the small amount of EVs in the network. Find the charging hours allowed for all seasons in Table 4.5, as well as the constraints that disables the charging.

***Table 4.5:** Charging hours allowed by the first SLM algorithm when the given EVs are connected and the constraints that disables the charging.*

144500771 desired charging hours:	00.00 - 03.00 and 16.00 - 19.00	Constraint:
Winter allowed charging hours:	00.00 - 03.00	Transfomer overload / I_b > I_z (80955-19)
Spring allowed charging hours:	00.00 - 03.00 and 16.00 - 19.00	
Summer allowed charging hours:	00.00 - 03.00 and 16.00 - 19.00	
Autumn allowed charging hours:	00.00 - 03.00 and 16.00 - 19.00	
144500771 desired charging hours:	00.00 - 03.00 and 15.00 - 18.00	Constraint:
Winter allowed charging hours:	00.00 - 03.00 and 15.00	Transfomer overload
Spring allowed charging hours:	00.00 - 03.00 and 15.00 - 18.00	
Summer allowed charging hours:	00.00 - 03.00 and 15.00 - 18.00	
Autumn allowed charging hours:	00.00 - 03.00 and 15.00 - 18.00	

The results of the PowerFactory validation are given in Table 4.6. The simulations are done for all seasons, the most power demanding hour of the day. During winter season, charging at peak hours is not allowed, but there are still a small overload of the transformer due to the household loads. In the autumn charging takes place at peak hours without overloading the transformer.

Table 4.6: *DIgSILENT PowerFactory validation results for the first SLM algorithm.*

Season **Hour**	Winter 16.00-17.00	Spring 16.00-17.00	Summer 16.00-17.00	Autumn 16.00-17.00
Transformer Loading [%]	106,10	73,80	58,10	94,40
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	92,60	64,00	46,30	82,90
Lowest voltage at residence [V]	186,00	199,00	210,00	187,00

4.2.2 SLM Results Objective 3

The output load data of the SLM algorithm presented in Section 3.4.3, are given in Table 4.7. Only the residences with an EV connected are shown since only EV loads are shifted if necessary. The results are from a winter scenario, where remaining residences are simulated with data given in Table 3.3.

Table 4.7: *Load results for a winter scenario of coordinated charging, objective 3.*

	Residence load [kW]							
Hour	144503089	144503070	144502932	144500731	144500771	114500751	144500615	144500741
00.00	6,07	6,07	6,07	6,07	2,77	2,77	6,07	2,77
01.00	2,77	2,77	2,77	2,77	2,77	2,77	6,07	2,77
02.00	2,77	2,77	2,77	2,77	2,77	2,77	6,07	2,77
03.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
04.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
05.00	2,77	2,77	2,77	2,77	2,77	2,77	2,77	2,77
06.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
07.00	4,04	4,04	4,04	4,04	4,04	4,04	4,04	7,34
08.00	4,04	4,04	4,04	4,04	4,04	4,04	4,04	7,34
09.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	6,54
10.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
11.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
12.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
13.00	3,24	3,24	3,24	3,24	3,24	3,24	3,24	3,24
14.00	3,24	3,24	3,24	3,24	3,24	3,24	6,54	3,24
15.00	3,24	3,24	3,24	3,24	3,24	3,24	6,54	3,24
16.00	6,93	6,93	6,93	6,93	6,93	6,93	6,93	6,93
17.00	8,09	8,09	8,09	8,09	8,09	8,09	8,09	8,09
18.00	6,93	6,93	6,93	6,93	6,93	6,93	6,93	6,93
19.00	6,93	6,93	6,93	6,93	6,93	6,93	6,93	6,93
20.00	6,47	6,47	6,47	6,47	9,77	6,47	6,47	6,47
21.00	4,85	4,85	4,85	4,85	8,15	8,15	4,85	4,85
22.00	8,15	8,15	8,15	8,15	8,15	8,15	4,85	4,85
23.00	7,34	7,34	7,34	7,34	4,04	7,34	4,04	4,04

The PowerFactory simulations are carried out at the same periods as in Section 4.1.2 with the uncoordinated charging. The key results are presented in Table 4.8. There are no cable overloading in this scenario and the transformer is only overloaded once at the most power demanding hour of the day. In the winter scenario, the transformer load is larger in the evening than originally due to shifted loads from previous hours.

Table 4.8: *DIgSILENT PowerFactory validation results for the second SLM algorithm with increased number of EVs in the low voltage distribution network.*

Season	Winter	Spring	Summer	Autumn
Hour	07.00-08.00	07.00-08.00	07.00-08.00	07.00-08.00
Transformer Loading [%]	58,60	33,50	45,40	61,50
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	44,90	25,30	36,50	49,10
Lowest voltage at residence [V]	211,00	223,00	216,00	208,00
Hour	16.00-17.00	16.00-17.00	16.00-17.00	16.00-17.00
Transformer Loading [%]	106,10	87,60	67,10	83,50
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	92,60	70,70	51,80	63,90
Lowest voltage at residence [V]	186,00	194,00	207,00	199,00
Hour	22.00-23.00	22.00-23.00	22.00-23.00	22.00-23.00
Transformer Loading [%]	87,30	83,30	40,10	48,40
Amount of overloaded cables	0	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	72,60	71,70	34,20	40,90
Lowest voltage at residence [V]	194,00	199,00	217,00	214,00

Since the SLM algorithm is most effective during winter due to the large power demand, a more detailed simulation is performed of this season. Four of the residences with an EV connected and different user profiles are shown below. The user profiles are described in Section 3.2.3, where the residences 144503089, 144500771, 144500615 and 144500741 have respectively user profiles 1, 2, 3 and 4. The following plots shows the desired charging compared to the charging allowed by the SLM algorithm. Of the residences with an EV connected, 144500741 is charging an EV at 07.00 AM, 144500615 is charging an EV at 16.00 and 144500731, 144502932, 144503070 and 144503089 are charging EVs at 22.00.

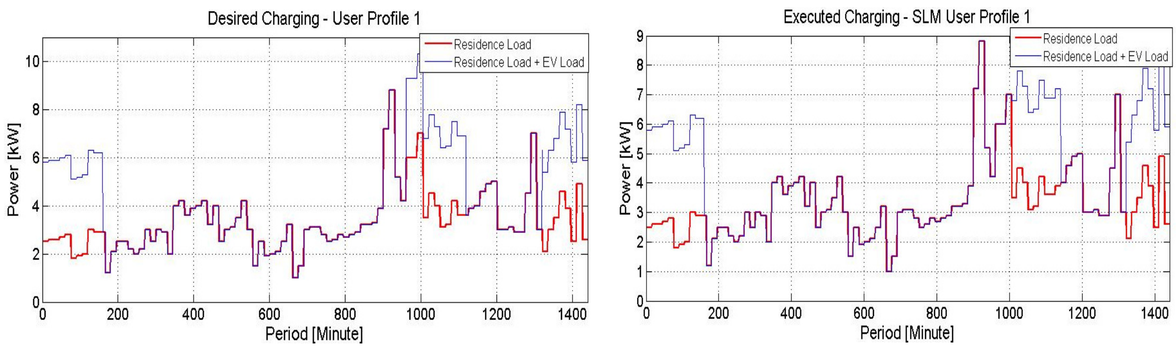

Figure 4.1: *Desired charging compared to executed charging for residence 144503089 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together.*

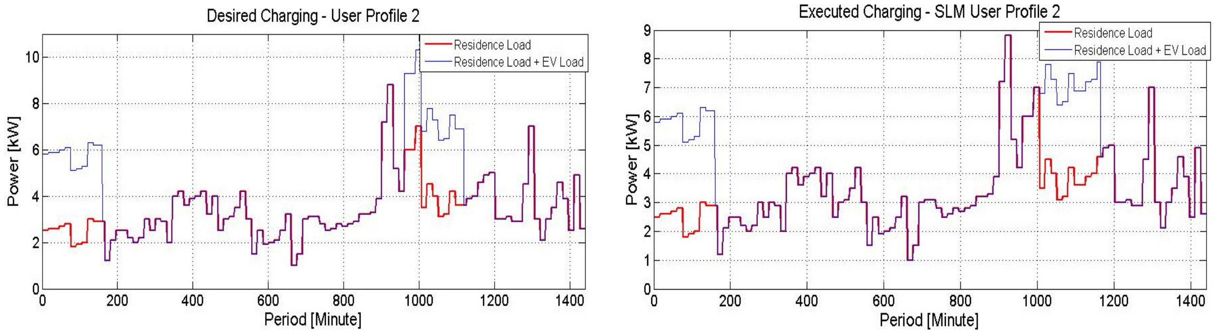

Figure 4.2: *Desired charging compared to executed charging for residence 144500771 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together.*

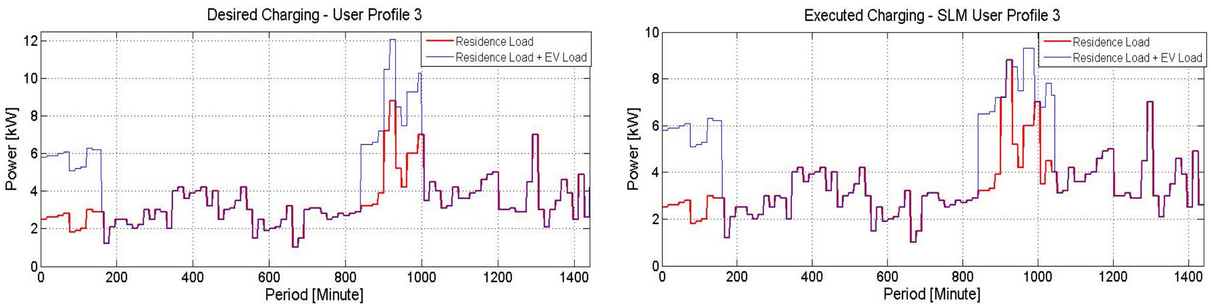

Figure 4.3: *Desired charging compared to executed charging for residence 144500615 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together.*

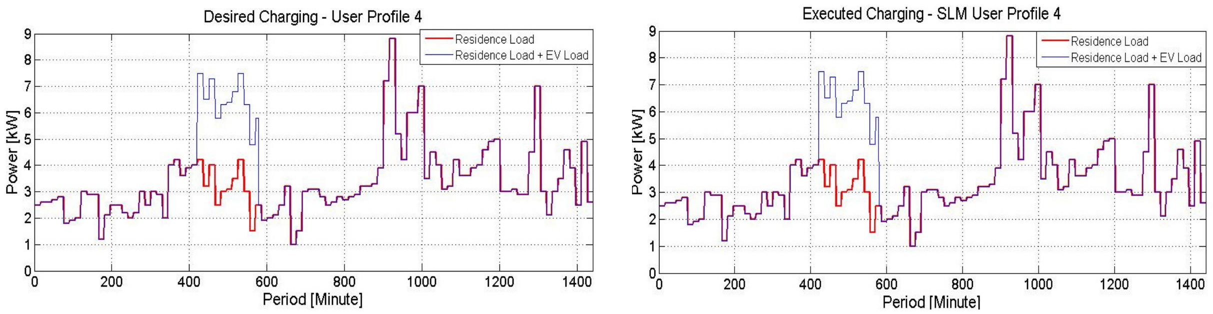

Figure 4.4: *Desired charging compared to executed charging for residence 144500741 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together.*

The DIgSILENT PowerFactory validation of the minute-based simulations are given in Table 4.9. Only the winter scenario is presented at three different periods of the day, the first minute of the same hours as the previous validations. That means 07.00 AM equals 420 minutes, 16.00 PM equals 960 minutes and 22.00 PM equals 1320 minutes. Note that the input household data fluctuates around the original data given in Table 3.3, which means that the minute-based results are not comparable with the hour-based results.

At 07.00 AM, residence 144500741 is charging an EV as predicted. Only residence 144500615 is charging an EV at 16.00 PM, and the original scheduled charging for all residences occur at 22.00.

__Table 4.9:__ DIgSILENT PowerFactory minute-based validation results for the second SLM algorithm with increased number of EVs in the low voltage distribution network.

Season	Winter	Winter	Winter
Hour	07.00-07.01	16.00-16.01	22.00-22.01
Transformer Loading [%]	58,20	92,40	36,80
Amount of overloaded cables	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	46,50	77,60	31,50
Lowest voltage at residence [V]	210,00	189,00	219,00

A scenario with 6,6 kW charging power and the same EVs in the network is also executed. The residence 144503089, 144500771, 144500615 and 144500741 with the user profiles 1,2,3 and 4 are shown to see the impact of the increased charging power. The plots below shows the desired charging compared to the charging allowed by the SLM algorithm utilizing 6,6 kW charging power for the EVs.

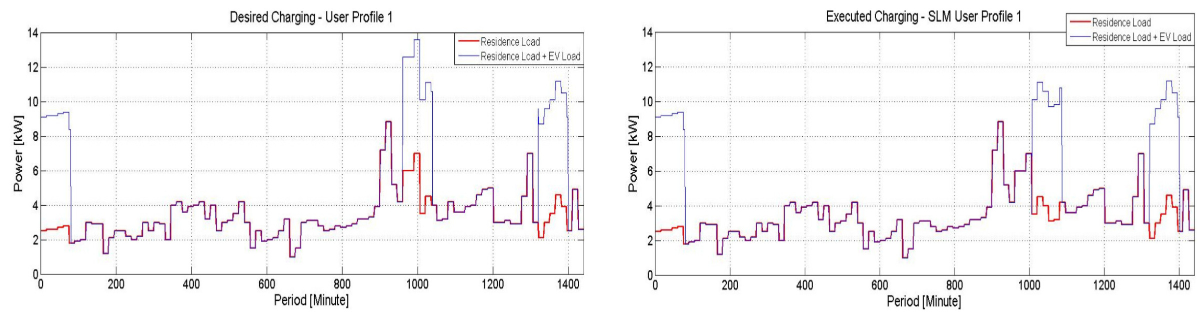

__Figure 4.5:__ Desired charging compared to executed charging for residence 144503089 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together. The simulation is performed with 6,6 kW charging power.

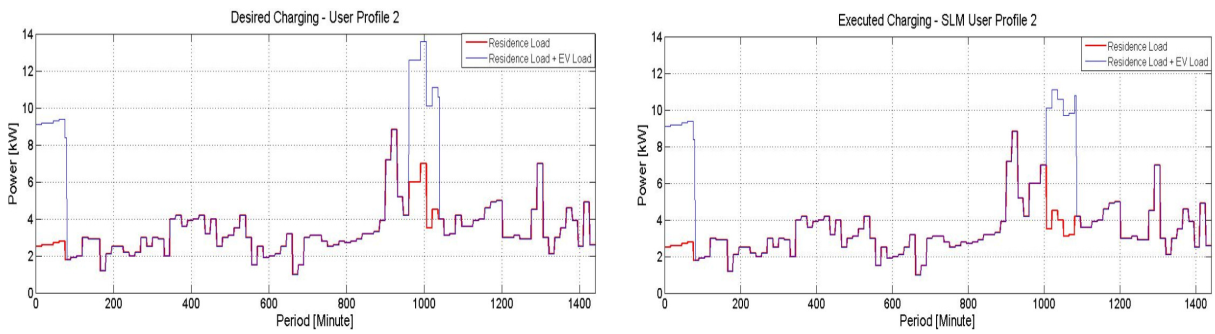

__Figure 4.6:__ Desired charging compared to executed charging for residence 144500771 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together. The simulation is performed with 6,6 kW charging power.

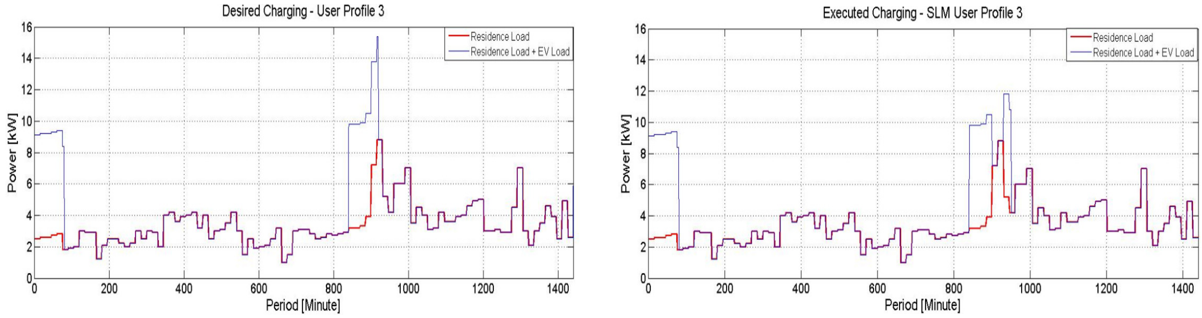

Figure 4.7: *Desired charging compared to executed charging for residence 144500615 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together. The simulation is performed with 6,6 kW charging power.*

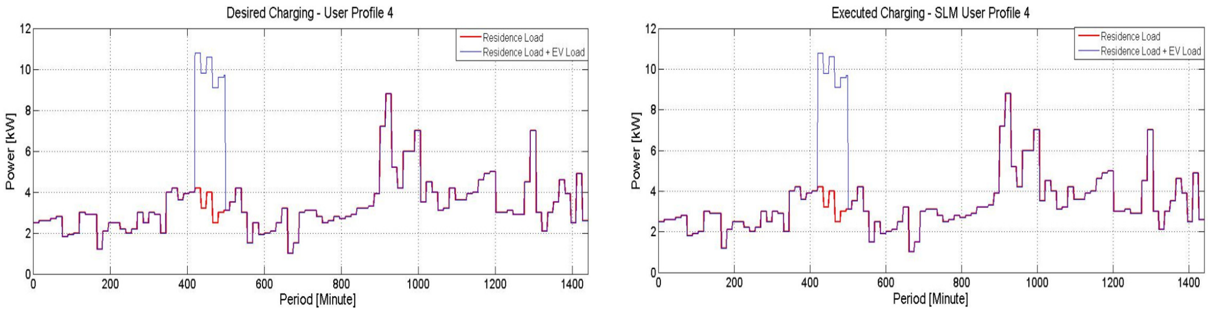

Figure 4.8: *Desired charging compared to executed charging for residence 144500741 with an EV connected. Red line represents the residence loads and blue line the residence and EV loads together. The simulation is performed with 6,6 kW charging power.*

The DIgSILENT PowerFactory validation is performed with the 6,6 kW charging power and the results are given in Table 4.10. At 07.00 AM only residence 144500741 is charging an EV as seen in Figure 4.8. At 16.00 PM none of the connected EVs are charging and at 22.00 PM, the residences 144503089, 144502932, 144503070 and 144500731 are charging one EV each.

Table 4.10: *DIgSILENT PowerFactory validation results for the second SLM algorithm with increased number of EVs and 6,6 kW charging power.*

Season **Hour**	Winter 07.00-07.01	Winter 16.00-16.01	Winter 22.00-22.01
Transformer Loading [%]	60,40	89,70	45,50
Amount of overloaded cables	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	45,30	77,30	40,40
Lowest voltage at residence [V]	209,00	190,00	212,00

4.3 DSM Results

In this section, the DSM results are presented. In the first scenario, the household loads are shifted if necessary, and in the second scenario only the EV loads are shifted.

4.3.1 DSM First scenario

The first DSM algorithm are shifting household loads according to Table 4.11. Which loads that are executed at the specific hour can be found in Table 3.11 and 3.12. Loads are subtracted when the preferred load is larger than the demand limit and loads are added up if the demand limit is larger than the preferred load and there are shifted loads from previous hours. This can be seen in the "Difference" column" in Table 4.11. The loads are as expected within the demand limit at all time.

Table 4.11: *Hour-based results of the first DSM simulation, where the original preferred loads, the final executed loads and the difference between the two are presented.*

Hour	Demand limit [kW]	Connection EV	Load org. [kW]	Final Load [kW]	Difference [kW]
00.00	6,75		6,07	6,07	0,00
01.00	6,75		6,07	6,07	0,00
02.00	6,75		6,07	6,07	0,00
03.00	6,75		2,77	2,77	0,00
04.00	6,75		2,77	2,77	0,00
05.00	5		2,77	2,77	0,00
06.00	5		3,24	3,24	0,00
07.00	4		4,04	4	-0,04
08.00	4		4,04	4	-0,04
09.00	4		3,24	3,32	0,08
10.00	5		3,24	3,24	0,00
11.00	4		3,24	3,24	0,00
12.00	4		3,24	3,24	0,00
13.00	4		3,24	3,24	0,00
14.00	4		3,24	3,24	0,00
15.00	4		3,24	3,24	0,00
16.00	6		10,23	6	-4,23
17.00	6		11,39	6	-5,39
18.00	6		10,23	6	-4,23
19.00	6,5		6,93	6,5	-0,43
20.00	6,5		6,47	6,5	0,03
21.00	6,5		4,85	6,5	1,65
22.00	6,5		8,15	6,5	-1,65
23.00	6,5		7,34	6,5	-0,84

When validating the results in PowerFactory for the same periods as the SLM results, the loads equals the demand limit as long as the original preferred loads are larger than this value. Only winter season was simulated since the demand limit is the same for all seasons. The results are given in Table 4.12.

Table 4.12: *DSM first scenario DIgSILENT PowerFactory validation.*

Season Hour	Winter 07.00-08.00	Winter 16.00-17.00	Winter 22.00-23.00
Transformer Loading [%]	54,50	78,70	98,40
Amount of overloaded cables	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	43,10	62,00	85,30
Lowest voltage at residence [V]	211,00	199,00	184,00

4.3.2 DSM Second Scenario

In this scenario, EV loads are shifted due to household load and charging power exceeding the demand limit given in Figure 3.10. Figure 4.9 shows the periods where EV charging is allowed if utilizing 3,3 kW charging power. User profile number 1, defined in Section 3.2.3, applies and the load data is the same for all residences.

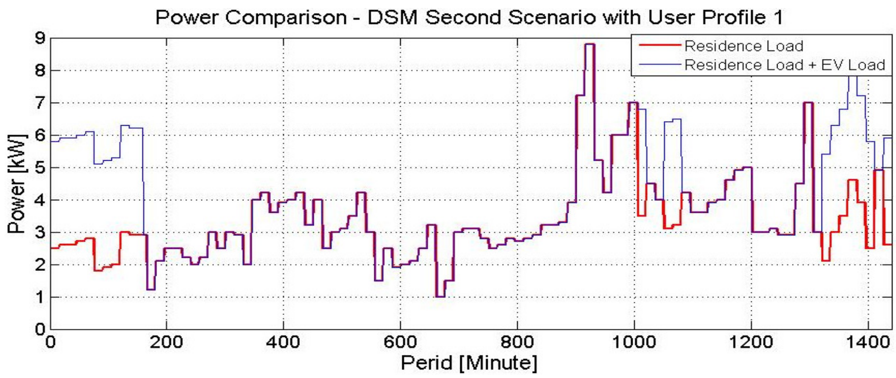

Figure 4.9: *Power comparison of the second DSM scenario with user profile 1. The red line represents the residence load, and the blue line the residence and EV load together.*

The PowerFactory validation are done for the first minute of the same hours as the previous validations. That means 07.00 AM equals 420 minutes, 16.00 PM equals 960 minutes and 22.00 PM equals 1320 minutes. Of the periods chosen, only the last scenario is simulated with EV charging. Since the industrial sector require a large amount of power in the morning, the demand limit for the residences is not large enough for EV charging. In the afternoon the household loads demand large amounts of power, which results in few possibilities for EV charging. The results of the validation are given in Table 4.13.

Table 4.13: *DSM second scenario DIgSILENT PowerFactory validation, with EVs connected to all the residences.*

Season	Winter	Winter	Winter
Hour	07.00-07.01	16.00-16.01	22.00-22.01
Transformer Loading [%]	56,00	89,70	72,00
Amount of overloaded cables	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	45,30	77,30	58,20
Lowest voltage at residence [V]	210,00	190,00	202,00

Since all residences are simulated with the same parameters, a simultaneity factor should be included for the EVs. The same locations for the eight EVs as in objective 3, will be used in the next scenario. That means residence tag 144503089, 144503070, 144502932 and 144500713 will have user profile number 1, 144500771 and 144500751 user connection 2, 144500615 user connection 3 and 144500741 user connection 4. The total load of every residence is still limited by the demand limit given in Figure 3.10. The initially SOC is 66,67 percent for all the vehicles in the network. The residence loads and executed charging loads for user connection 1 is shown in Figure 4.9. The executed charging loads for user profile 2, 3 and 4 are given by Figure 4.10, 4.11 and 4.12 in the following page. The plots shows the possibilities for charging when simulating one minute periods. The desired charging can be seen to the left in Figure 4.1, 4.2, 4.3 and 4.4 for the same residences with the same charging power.

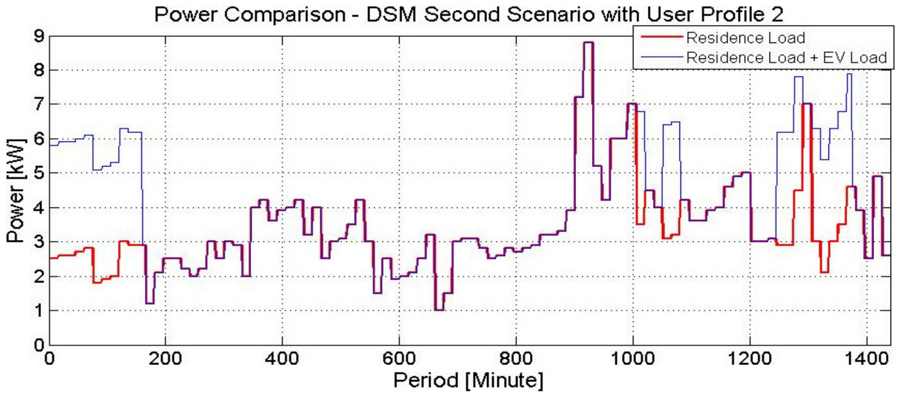

Figure 4.10: *Power comparison of the second DSM scenario with user profile 2, residence 144500751. The red line represents the residence load, and the blue line the residence and EV load together.*

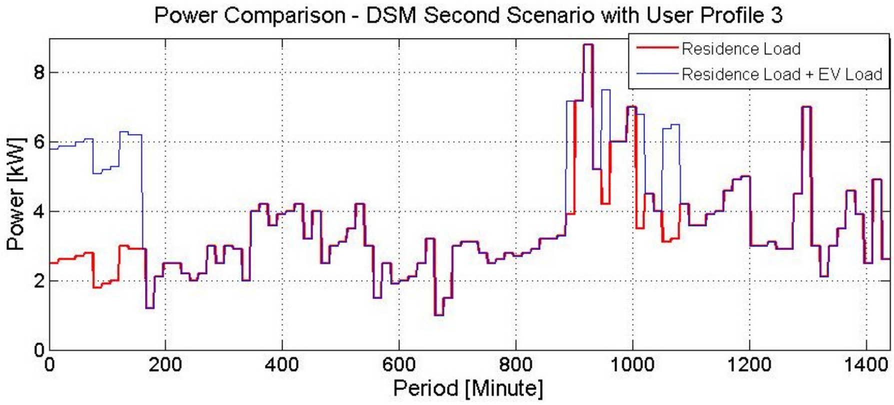

Figure 4.11: *Power comparison of the second DSM scenario with user profile 3, residence 144500615. The red line represents the residence load, and the blue line the residence and EV load together.*

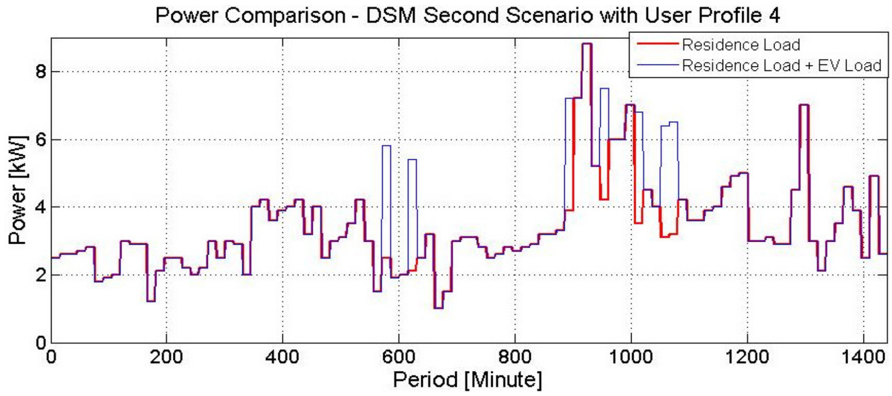

Figure 4.12: *Power comparison of the second DSM scenario with user profile 4, residence 144500741. The red line represents the residence load, and the blue line the residence and EV load together.*

In Table 4.14 the new results are given. Of the chosen periods, the demand limit only allows charging in the evening scenario. The residences 144503089, 144503070, 144502932 and 144500713 have connected EVs at 22.00 which are allowed to charge. 144500771 and 144500751 have had connected EVs since 16.00, but are also charging at 22.00 due to shifted loads.

***Table 4.14**: DSM second scenario DIgSILENT PowerFactory validation. Eight residences are chosen to have an EV connected.*

Season	Winter	Winter	Winter
Hour	07.00-07.01	16.00-16.01	22.00-22.01
Transformer Loading [%]	56,00	92,40	41,20
Amount of overloaded cables	0	0	0
Amount of overloaded main feeders	0 / 2	0 / 2	0 / 2
Highest cable/feeder loading [%]	45,30	77,60	33,30
Lowest voltage at residence [V]	210,00	189,00	218,00

4.4 Power Loss and Voltage Drop

A power loss comparison between the uncoordinated charging, the SLM algorithm and the DSM algorithm are performed with eight EVs in the network and user profiles similar to the ones used in Section 4.2.2. The network is also simulated without EVs for calculation of initial loss. 66,67 SOC, 3,3 kW charging power and winter season applies for all simulations performed in DIgSILENT PowerFactory. The results are given in Table 4.15, where the data is collected 07.00 AM, 16.00 PM and 22.00 PM. The minute-based calculations were used and the loss is presented as a summation of all transmission cable loss from busbar 80955T1 to the residences.

***Table 4.15:** Power loss comparison between no EVs in the network, uncoordinated charging, the SLM programme and the DSM programme.*

Hour	No EVs [kW]	Uncoordinated [kW]	SLM [kW]	DSM [kW]
07.00	4,540	4,756	4,756	4,540
16.00	19,315	28,552	26,484	26,484
22.00	6,430	12,540	12,540	13,806

A voltage comparison from the four objectives simulated at 16.00 PM is given in Figure 4.13 on the next page. The bars shows the voltage at the residences with none EVs in the network, along with uncoordinated charging of two and eight EVs. The impact of the SLM and DSM programmes are shown in the three right bars for each residence.

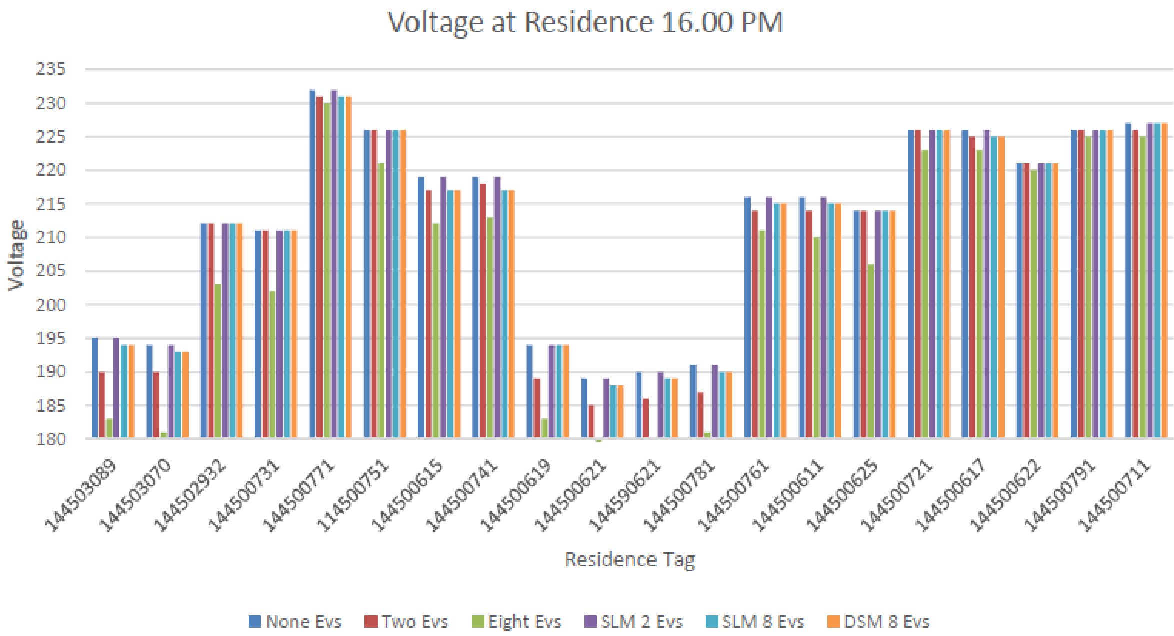

Figure 4.13: Voltage comparison at each residence. *The bars shows voltage at each residence with none, two and eight EVs in the network along with the impact of the management programmes.*

4.5 Economical Aspect

Due to overloading of cables, low short circuit currents or under voltage at the network subscribers, the grid may need reinforcement. An alternative to a management programme that coordinates the loads is exchange or reinforcement of the cables.

In Table 4.16, an estimate of the cost of new cabling is presented. This estimate is prepared by SINTEF and the price is presented in Norwegian kroner per kilometer [15]. The 240 mm^2 cable is used as feeder ground cables, and the 95 mm^2 and 150 mm^2 cables used as both supply overhead lines and ground cables for the residences.

Table 4.16: *Investment cost of new power lines [15].*

Installation	Cost [NOK/km]
EX 3x95 mm²	275 244
EX 3x150 mm²	302 373
TFXP 4x240 AL	165 073
Trench village	311 266
Sum cable installation 4x240 AL	**476 339**

4.6 Summary of Results

This section will give a short summary of the simulation results that were executed for all the objectives defined in Section 1.3. The scenarios were simulated with uncoordinated charging of the EVs, and compared with the effect of the management programmes.

Of the four seasons, winter season is the most power demanding. In all of these scenarios, the SLM algorithm disabled charging in some, or all of the peak hours. Overloading of the transformer was the most critical constraint in the given low voltage distribution network, but also overloading of supply cables occurred. In the uncoordinated winter scenario with only two EVs in the network the transformer load was 112,4 percent at 16.00 PM. In the scenario with eight EVs in the network, the transformer load was 127,9 percent at the same period.

When applying the hour-based SLM programme to the network, charging at peak hours were denied for the winter scenario. This resulted in a transformer load of 106,1 percent and no overloading of cables. In the autumn scenario, the transformer was overloaded in the peak hours when eight EVs were placed in the network. When applying the SLM programme the transformer load resulted in 83.5 percent, and only two of the EVs were charged this period. Spring and summer season scenarios never exceeded the constraints defined in Section 3.3.

Minute-based simulations were carried out to see the possibilities for small periods of charging within the peak hours. The EV connected to residence 144500615 were fully charged within this period, while some EV loads were shifted. In the hour-based simulations the residences 144500771 and 114500751 had shifted EV loads at 22.00, but with the minute-based calculations only the originally scheduled EV loads were executed at 22.00. A scenario with 6,6 kW EV charging power were performed which resulted in denied requested charging of the EV connected to residence 144500615 at 16.00 PM, and larger transformer loading at 07.00 AM and 22.00 PM.

The first DSM algorithm proposed a solution where charging of the EVs effected other household loads. This resulted in some charging of the EVs at peak hours, while the remaining power requirements were shifted. The second DSM algorithm only shifted EV loads if the constraints given by Equation (3.1),(3.2) or (3.4) were violated. This also resulted in some charging of the EVs before disconnection due to the predetermined user profiles. At 07.00 AM residence 144500741 had EV load shifted due to the demand limit. At 16.00, the charging of the EVs were equal to the SLM results. At 22.00 PM, shifted EV loads were executed together with the originally scheduled loads.

Due to a more advanced calculation of the voltage drop in DIgSILENT PowerFactory there were divagations in the voltage at the residences, compared to the MATLAB calculations. In most of the winter scenarios the constraint given by Equation (3.1) were violated due to under voltage calculated by DIgSILENT PowerFactory. This applied for the residences further from the transformer.

5 Discussion

This chapter gives a review of the scenarios simulated in DIgSILENT PowerFactor. The role of the management programmes will be discussed, where the suggested solutions both should satisfy the users and comply with the limitations in the given low voltage distribution network.

5.1 SLM Programmes

In general, the transformer is barely able to supply the given low voltage distribution network in power demanding periods without EVs in the network. In addition to the transformer overload, the maximum current capacity of some of the cables are exceeded when adding EV loads to the residences. The branch with feeder 80955-0 and overhead line 80955-1 supplies 12 residences with cable capacities of 435 A and 280 A. Further, the overhead line 80955-16 supplies 10 residences with a cable capacity of 280 A, while 80955-19 supplies 8 residences alone with a cable capacity of 180 A. Of the cables on this branch, 80955-16 and 80955-1 are close to, or exceeding their maximum current capacity when adding EV loads in the network. The branch that starts with feeder 80955-0A, supplies 8 residences and are not at any time exceeding the maximum cable capacities.

The hour-based SLM algorithm proposed a strict solution during winter where all EV loads were shifted at peak hours from 16.00 PM to 21.00 PM, except the EV load connected to residence 144500771, which was executed at 20.00 PM. Due to the shifted charging there were no cable violations and the overload of the transformer was reduced. At 20.00 PM the transformer load was still above 80 percent, and only the EVs that caused less power loss in the distribution lines were charged. Residence 144500771 is located close to the transformer and caused the smallest power loss, and was therefore scheduled for charging first. In the autumn, transformer overload of 109 percent occurred at 16.00 with eight EVs in the network. With the SLM algorithm the transformer load resulted in 83,5 percent during the same period.

Simulating minute-based periods the MATLAB plots shows periods within the peak hours where the EVs were charging. This is a more realistic scenario with a more user-friendly result. As seen in the plots the worst peaks were avoided, which resulted in a 45 minutes charging delay of the EVs connected to residence 144503089, 144503070, 144502932, 144500731, 144500771 and 114500751. The EV connected to residence 144500615 was charged in two periods of 30 and 15 minutes, and residence 144500741 had no shifted loads. The simulations with 6,6 kW charging power resulted in 45 minutes shifted EV loads for residence 144503089, 144503070, 144502932, 144500731, 144500771, 114500751 and an interruption of 30 minutes of the EV connected to residence 144500615. The EV connected to residence 144500741 was charged in periods where the residence loads were small and are therefore charged as scheduled.

The total power loss in the transmission lines was 48 percent larger with a high penetration of EVs in the network at 16.00 PM, compared to the scenario with no EVs in the network. The difference between uncoordinated charging and the SLM programme were only eight percent.

5.2 DSM Programmes

The first DSM algorithm offered a solution where the user had more liberty to choose his or her own loads. Charging of the connected EV was executed as scheduled during night since the demand limit was large this period. During peak hours, the DSM allowed one hour of charging, before heating was prioritized.

Due to the demand limit, the DIgSILENT PowerFactory validation never exceeded the constraints in the network, but the residence struggled to catch up with the shifted loads. The loading of the transformer was larger in the evening compared to the previous peak hours between 16.00 PM and 19.00 PM. The DSM programme shifted the household peak hours to the evening, while the industry sector is responsible for most of the loads during day. Since most people are at work during the day, the low demand limit usually does not interfere with the household comfort this period.

When applying the different user profiles and only shifted EV loads the DSM programme faced new challenges. Due to the low demand limit during day, charging of EVs was limited. User profile number 4 was connected between 07.00 AM and 20.00 PM, but was only charged 1 hour and 45 minutes divided in six charging periods. The 6,6 kW charging scenario was not simulated for this programme, due to the demand limit which would disable most of the charging.

The minute-based MATLAB plots of the simulations shows the charging periods of the EVs connected to the residences. Residence 144503089, 144503070, 144502932, 144500731 have connected EVs that were charged for 45 minutes before disconnection. The EVs connected to residence 144500771 and 114500751 were finished charging at 23.00 PM and the EV connected to residence 144500615 was charged in three periods, in total one hour, before disconnection.

Simulating the DSM programme resulted in larger share of EV loads that were shifted to the evening. This resulted in a larger transformer loading at 22.00 PM than originally scheduled, and the residences with user profile 3 and 4 had no possibilities to catch up with the shifted loads within a 24-hour period due to the demand limit. The difference of the total power loss in the distribution lines are small between the uncoordinated charging and the DSM programme, and the loss is 1,266 kW larger at 22.00 compared to the SLM programme.

5.3 Possibilities and Relevance

As seen in the DIgSILENT PowerFactory validation tables and Figure 4.13, the voltage at some of the residences are under the boundary set by Equation (3.1). This applies to the eight residences supplied by overhead line 80955-19. In the winter at 16.00 PM, all of these residences have under voltage. By replacing the overhead line to a 3x95 mm^2cable instead of the existing 3x50, mm^2 the voltage would increase with approximately 8 percent at all the applicable residences with EVs connected. The cost of this operation would be 275 244 NOK/km plus work, according to Table 4.16. The line would also be further from the maximum capacity limit, which decreases the thermal stress.

The network operators are interested in keeping the distribution network as effective as possible. Using the reference impedance of 0,08+j0,05 Ω, the minimum bipolar short circuit current at the residences are 1,172 kA. This applies for electrical devices with rated currents below 16 A. As a rule of thumb, the minimum short circuit current should be 0.5 kA when reinforcing an existing network [15]. In the existing network residence 144500731, 144500621, 144590621 and 144500622 have too small short circuit currents, but replacing the overhead line as mentioned above does that residence 144500621 and 144500622 exceeds the recommended value. Residence 144500622 and 144500731 are located on the other branch and it would be cost ineffective to improve these lines.

The SLM programme is an effective solution, which mostly limits the EV charging between 16.00 PM and 20.00 PM. The transformer and overhead line 80955-19 are currently the main limitations, in addition to under voltage detected by the PowerFactory software. The SLM programme offer a solution where all loads are executed within the 24-hour period, while the DSM programme still has shifted loads at 24.00 PM. The strict demand limit combined with the constraints in the network makes the EV charging challenging, while the SLM programme offer a reasonable response time.

6 Conclusion

Charging of EVs will increase the power demand in the low voltage distribution networks. The IT low voltage distribution network, as the one simulated in this project, is the most common system in Norway. These systems are often weak, with small bipolar short circuit currents at the network subscribers, which makes increased penetration of EVs challenging. Load management programmes could be an alternative to replacement or reinforcement of the distribution networks, and the worst power peaks could be avoided.

The aim of this project was to analyze the impact of increased penetration of EVs in the low voltage distribution network and develop a possible solution through load management programmes. This was accomplished by simulating a typical Norwegian network all seasons of the year with different EV load scenarios. The network was simulated using none, two and eight EVs connected. The Nissan Leaf battery specifications were used as a reference, where 3,3 kW and 6,6 kW charging power and a 21 kWh battery were tested. The impact of the original scheduled EV loads were compared with the results from applying the management programmes to the system, and discussed with respect to network stability and user satisfaction.

Hour-based simulations of the SLM programme gave conservative results, where charging of the EVs in the network were disabled from 16.00 to 20.00 PM during winter. Due to that, the transformer overload was reduced with 5.9 percent at 16.00 when simulating two EVs in the network, and 20,5 percent with eight EVs in the network. The reduction of EVs also had impact on overhead line overload and thermal stresses of cables. The minute-based SLM programme gave a more user-friendly result. The household load data was given in 15-minutes periods, which gave charging opportunities within the peak hours. The constraints in the network were also monitored during simulations, such that overloading of the transformer and cables were reduced. All scheduled EV loads were executed within a 24-hour period, with a maximum of 45 minutes charging delay.

The DSM programmes were divided into a scenario where household loads were shifted if necessary and a scenario where only EV loads were shifted. The first scenario resulted in a shift of the peak hours from afternoon to evening. Due to a strict demand limit, the residences never exceeded the power limitations in the network, but the users struggled catching up with the shifted loads. In the second scenario, the charging of the EVs were presented similar to the minute-based SLM programme. Due to the demand limit combined with the network constraints, charging of the EVs were limited. The DSM programme did not manage to fully charge all the vehicles in the network within the 24-hour period, but the overall load of the transformer and thermal stresses on the cables were reduced.

At spring and summer seasons the charging of the EVs were executed as scheduled. The household and EV loads still created a power peak in the afternoon, but not large enough to delay the charging. This applied for both programmes simulating 3,3 kW charging power, and only the SLM programme simulating 6,6 kW charging power. When simulating the autumn scenario, some of the EV loads were shifted during peak hours, which resulted in transformer and cable loads within their limitations.

The transformer was the main bottleneck for the system, but also overloading of overhead lines limited the charging of the EVs. Of the programmes tested, the minute-based SLM programme gave the best results. The constraints were adhered, and the delay of the charging was reasonable. Increased charging power was also possible for this programme during winter with similar results. The voltage divagation between the MATLAB and the DIgSILENT PowerFactory simulations could interfere with the results of the management programmes. A more strict and detailed calculation of the voltage drop in the MATLAB simulations is necessary to decrease this divagation.

Continuing success of EVs is promising if a management programme can be implemented as a feature of the upcoming smart grid technology. The evolution of the EVs is promising, which will result in larger battery capacities and faster charging that demands more power and creates larger power peaks. It may be necessary to investigate the different distribution networks individually for the need of cable reinforcement, load management solutions or both. Network subscribers with the smallest bipolar short circuit currents are especially vulnerable for additional EV loads. It must also be taken into account that overloading of the transformer, thermal stresses on the cables and under voltage mainly occurred during winter. That means with the current share of EVs and with larger share of EVs, only one quarter of the year is critical with respect to the limitations in the network.

Future Work

Some suggestions for future work will be presented. A more detailed simulation of the EV loads could be performed where unbalanced loads should be taken into account. This is important dealing with household management, where the inlet conductor currents should be approximately equal.

More different user profiles together with different types of EVs would make the simulations more credible and interesting. With the implementation of smart metering, real time data from connection and disconnection of EVs could be established. Different types of EVs presents the challenge with larger charging power and battery capacities.

Finally, the harmonic distortion of the chargers should be established together with the impact of harmonic currents on the low voltage distribution network.

Bibliography

[1] Alexander Probst, Martin Siegel, Martin Braun, Stefan Tenbohlen, *"Impacts of electric mobility on distribution grids and possible solution through load management",* 21st International Conference on Electricity Distribution, Paper 0126-. Frankfurt 3-9 June 2011.

[2] Rakesh Sinha, Eloy Rodrìguez Moldes, Arsalan Zaidi, Pukar Mahat, Jayakrishnan R. Pillai, Peter Hansen, "*An Electric Vehicle Charging Management and its Impact on Losses",* 2013 4th IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), October 6-9, Copenhagen 2013.

[3] A.s. Masoum, S. Deilami, P.S. Moses, M.A.S. Masoum, A. Abu.Siada, "*Smart load management of plug-in electric vehicles in distribution and residential networks with charging stations for peak shaving and loss minimisation considering voltage regulation", IET Generation, Transmission & Distribution*, Vol 5, pp. 877-888, 2011.

[4] Alberto Palomar Lozano, "Intelligent Energy Management of Electric Vehicles in Distribution Systems", Short Maser Thesis in Electric Power Systems and High Voltage Engineering, Board of Studies of Energy, Aalborg University, Aalborg, 2012.

[5] Grønn bil. (2014). *"Ladbare biler i Norge des 2014".* [Online]. Available: http://www.gronnbil.no/statistikk/.

[6] Espen Zachariassen, "Smart lading av elbiler skal kutte kapasitetsbehovet med 25 prosent," *Teknisk Ukeblad,* 16. October. 2014, [Online]. Available: http://www.tu.no/it/2014/10/16/smart-lading-av-elbiler-skal-kutte-kapasitetsbehovet-med-25-prosent. [Retrieved 23. January 2015].

[7] Henriette Forsetlund, "Forskningsrådet med millionstøtte til ChargeFlex prosjektet," *Esmart Systems,* 16. October. 2014, [Online]. Available: http://www.esmartsystems.com/news-events/forskningsraadet-med-millionstoette-til-chargeflex-prosjektet-1/. [Retrieved 26. January 2015].

[8] Grønn bil. (2014). *"Bilmodeller".* Available: http://www.gronnbil.no/bilmodeller/

[9] IEC 61851-1, *Electric Vehicle Conductive Charging System - Part 1: General Requirements.* IEC, 1st ed., 2001.

[10] IEC 62196-2, *Plugs, Socket-outlets and Couplers for industrial and similar applications, and for Electric Vehicles - Part 2 Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories.* IEC. 2nd ed., 2011.

[11] D. Sbordona, I. Bertine, B. Di Pietra, M.C. Falvo, A. Genovese, L. Martirano, *"Electric Power Systems Research", Elsevier B.V.,* Vol 120, 96-108, 2015.

[12] Toyota Motor Corporation. (2015). "*Technology File, Plug-in Hybrid Vehicle",* [Online], Available: http://www.toyota-global.com/innovation/environmental_technology/technology_file/plug-in_hybrid.html.

[13] Battery University. “BU-205: Types of Lithium-ion“, 2014. [Online]. Available: http://batteryuniversity.com/learn/article/types_of_lithium_ion.

[14] Battery University. “BU-1204: Batteries for Electric Cars”, 2014. [Online]. Available: http://batteryuniversity.com/learn/article/batteries_for_electric_cars.

[15] Helge Seljeseth, Kjell Sand, Tarjei Solvang, «Håndtering av utfordrende elektriske apparater som tilknyttes elektrisitetsnettet», SINTEF Energi AS Report 2012-12-19, TR A7203, 2012.

[16] Jan H. Sebergsen, Steinar Svarte, *Energiproduksjon og energidistribusjon – Produksjon, nettsystemer og beregninger,* Oslo: Gyldendal Norsk Forlag AS, 2008.

[17] *Forskrift om leveringskvalitet i kraftsystemet,* 2004 part 14. [Online]. Available: https://lovdata.no/dokument/SF/forskrift/2004-11-30-1557.

[18] Judith Rossebø, R&D Group Manager, “The Active House. Background and Technology – Introduction and Overview”, UiA Smart Grid course ENE502, Grimstad, 20.10.2014.

[19] IEEE. “IEEE & Smart Grid”, 2015. [Online]. Available: http://smartgrid.ieee.org/ieee-smart-grid.

[20] PRIME. “Advanced Metering”, 2013. [Online]. Available: http://www.prime-alliance.org/?page_id=310.

[21] *Norsk elektroteknisk norm, NEK 400:2010 4 utg. Elektriske lavspenningsintallasjoner,* 433.1 1), 2010.

[22] Benedicte Langseth, Emilie Nærum Everett, Kjetil Ingeberg. (2006, 11), *Hovedundersøkelse for elektrisitetsbruk i husholdningene [Online].* 2011-06. Retrieved from: www.nve.no/PageFiles/15092/Hovedrapport_Elbrukende_apparter.pdf.

[23] Statistisk Sentralbyrå. (2013). *Kjørelengder, 2013. Available:* https://www.ssb.no/transport-og-reiseliv/statistikker/klreg/aar/2014-05-08?fane=tabell#content.

[24] Statistisk Sentralbyrå. (2013). *Registrerte kjøretøy, 2013. Available:* https://www.ssb.no/transport-og-reiseliv/statistikker/bilreg.

[25] Randi Hjorthol, Liva Vågane, Jens Foller, Bettina Emmerling, “Potensialet for bruk av elektriske biler på de daglige reisene“, *Transportøkonomisk institutt, Stiftelsen Norsk senter for samferdselsforskning,* TØI rapport 1352/2014, 58 pages, 2014.

[26] Statistisk Sentralbyrå. (2012). *Gjennomsnittlig energiforbruk for husholdninger med og uten varmepumpe. kWh tilført energi. Available:* http://www.ssb.no/a/kortnavn/husenergi/tab-2011-04-19-10.html.

[27] “DIgSILENT PowerFactory”, 2015. [Online]. Available: http://www.digsilent.de/index.php/products-powerfactory.html.

Appendices

A – Cable data

B – MATLAB script objective 1

C – MATLAB script objective 2

D – MATLAB script objective 3

E – MATLAB script objective 4, Scenario 1

F – MATLAB script objective 4, Scenario 2

Appendix A - Cable data

Ground cable: (0,23 kV)

Node 1	Node 2	Type	Length [km]	Resistance [Ω]	Reactance [Ω]	Max. Current capacity [A]	Capacity to ground [uF]
80955-A1	144503089	TFXP 4X95 AL	0,135	0,0432	0,0101	220	0,0770
80955-A1	144503070	PFSP 3X50 AL	0,103	0,0661	0,0103	150	0,0690
80955T1	80955-0	TFXP 4X240 Al	0,009	0,0011	0,0006	435	0,0058
80955T1	80955-0A	TFXP 4X240 Al	0,008	0,0010	0,0006	435	0,0051
80955-29	144502932	TFXP 4X50 AL	0,041	0,0263	0,0032	150	0,0217
80955-4	80955-5	PFSP 3X50 AL	0,031	0,0199	0,0024	150	0,0164

Overhead line: (0,23 kV)

Node 1	Node 2	Type	Length [km]	Resistance [Ω]	Reactance [Ω]	Max. Current capacity [A]
80955-1	80955-16	EX 3X95 AL	0,0980	0,0314	0,0074	280
80955-16	80955-17	EX 3X25 AL	0,0500	0,0600	0,0041	115
80955-17	144500615	EX 3X25 AL	0,0130	0,0156	0,0011	115
80955-16	80955-19	EX 3X50 AL	0,0380	0,0244	0,0029	180
80955-19	80955-22	EX 3X50 AL	0,1330	0,0853	0,0102	180
80955-22	80955-23	EX 3X25 AL	0,0370	0,0444	0,0030	115
80955-23	80955-A1	EX 3X25 AL	0,0030	0,0036	0,0002	115
80955-23	80955-24	EX 3X25 AL	0,0460	0,0552	0,0038	115
80955-24	144500619	EX 3X25 AL	0,0060	0,0072	0,0005	115
80955-22	80955-26	EX 3X25 AL	0,0860	0,1032	0,0071	115
80955-26	144500621	EX 3X25 AL	0,0420	0,0504	0,0034	115
80955-26	144590621	EX 3X25 AL	0,0250	0,0300	0,0021	115
80955-19	144500611	EX 3X25 AL	0,0100	0,0120	0,0008	115
80955-1	80955-18	EX 2X25 AL	0,1770	0,2124	0,0145	115
80955-0	80955-1	EX 3X95 AL	0,0300	0,0096	0,0023	280
80955-0A	80955-1A	EX 3X95 AL	0,0300	0,0096	0,0023	280
80955-1A	80955-9	EX 3X95 AL	0,1540	0,0493	0,0117	280
80955-9	80955-10	EX 3X95 AL	0,0380	0,0122	0,0029	280
80955-10	80955-14	EX 3X50 AL	0,2040	0,1308	0,0157	180
80955-14	144500625	EX 3X25 AL	0,0110	0,0132	0,0009	115
80955-14	80955-29	EX 3X50 AL	0,0350	0,0224	0,0027	180
80955-9	144500617	EX 3X25 AL	0,0180	0,0216	0,0015	115
80955-1A	80955-4	EX 3X50 AL	0,1200	0,0769	0,0092	180
80955-5	144500622	EX 3X25 AL	0,1570	0,1884	0,0129	115
80955-29	144500731	EX 3X25 AL	0,0420	0,0266	0,0033	115
80955-1	144500771	EX 3X25 AL	0,0018	0,0216	0,0024	115
80955-18	114500751	EX 3X25 AL	0,0016	0,0192	0,0013	115
80955-17	144500741	EX 3X25 AL	0,0012	0,0144	0,0010	115
80955-26	144500781	EX 3X25 AL	0,0044	0,0528	0,0036	115
80955-19	144500761	EX 3X25 AL	0,0021	0,0300	0,0021	115
80955-9	144500721	EX 3X25 AL	0,0017	0,0227	0,0017	115
80955-4	144500791	EX 3X25 AL	0,0268	0,3216	0,0220	115
80955-5	144500711	EX 3X25 AL	0,0129	0,1884	0,0129	115

Appendix B – MATLAB script objective 1

```
% Smart Load Management for electric vehicles
% ----------------------------------------------------------------------
% SLM selection algorithm objective 1
% Master Thesis Spring 2015
% Author: Stian Namtvedt Gjelsvik
% Email: stian2803@gmail.com
% Version: 1.00
% Last update: 03.03.2015
% ----------------------------------------------------------------------
% Description:
% This script randomly places a given number of electric vehicles (EVs)
% in a low voltage distribution network. Hourly-based connection of the
% EVs and load profiles are selected.
% ----------------------------------------------------------------------
clc;
clear;
% Load distribution and initialization:
% Input:
P_charger = 3.3;   % Charging power (depending on the charger)
SOC=0.6666;           % State of charge. 21 kWh max. 7kWh used: SOC 0,666
Max_bat=21;        % Maximum capacity [kWh] battery
SOC_Recom=0.8;     % Recommended state of charge
Charge_Eff=0.93;   % Charge efficiency
Trans_Eff=0.865;   % Transportation efficiency
Share_of_EV=2;     %How many EVs in the network
% Load profile for residence for the seasons, hourly based 00.00 to 23.00
Load_Summer = [0.97 0.97 0.97 0.97 0.97 0.97 1.62 3.24 3.24 0.81 0.86 0.86
0.86 0.86 0.86 0.86 3.88 3.53 2.16 2.16 2.16 1.94 1.62 1.62];
Load_Autumn = [1.58 1.58 1.58 1.58 1.58 1.58 2.97 4.45 4.45 1.58 1.58 1.58
1.58 1.58 1.58 2.97 5.93 5.93 5.93 5.08 4.45 3.95 2.97 2.97];
Load_Winter = [2.77 2.77 2.77 2.77 2.77 2.77 3.24 4.04 4.04 3.24 3.24 3.24
3.24 3.24 3.24 3.24 6.93 8.09 6.93 6.93 6.47 4.85 4.85 4.04];
Load_Spring = [1.29 1.29 1.29 1.29 1.29 1.29 1.29 2.16 2.16 1.48 1.48 1.48
1.48 1.48 1.48 4.71 4.71 4.31 3.70 3.24 2.88 2.59 2.35 2.16];
%Probability for user profile
User_Profile = [1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 4 ];
% Different probabilities for which EV user that are connected.
% Set to 50 percent for EV_User_1, 35 percent for EV_User_2, 10 percent
% for EV_User_3 and 5 percent for EV_User_4.
User = datasample (User_Profile, 20);  % Choose random User_Profiles
% User is a vector that represents the residences in the network.
% User connection from 00.00 to 23.00, where 0 is disconnected and 1
% connected
User_Connection_1 = [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1];
User_Connection_2 = [1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
User_Connection_3 = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0];
User_Connection_4 = [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0];

% Selection calculations: ---------------------------------------------
% Deside which residence containing EV and what kind of user connection:
EVs_in_use = unidrnd (20,[1,Share_of_EV]);
EV_matrix = [User(1) User(2) User(3) User(4) User(5) User(6) User(7)
User(8) User(9) User(10) User(11) User(12) User(13) User(14) User(15)
User(16) User(17) User(18) User(19) User(20)];

EV_1=EVs_in_use(1);
EV_2=EVs_in_use(2);
```

```
%Which residence have been selected and type of User_Connection Selected
EV_User_1=EV_matrix(EV_1);
EV_User_2=EV_matrix(EV_2);
% Could change name of EV_User_X and use for comparison (shift
% charging hours)
if EV_User_1 == 1
    EV_User_1 = User_Connection_1;
    SOC1=1;
    User_Connection_No1=1    % Display which user connection before SOC
end
if EV_User_1 == 2
    EV_User_1 = User_Connection_2;
    SOC1=2;
    User_Connection_No2=1
end
if EV_User_1 == 3
    EV_User_1 = User_Connection_3;
    SOC1=3;
    User_Connection_No3=1
end
if EV_User_1 == 4
    EV_User_1 = User_Connection_4;
    SOC1=4;
    User_Connection_No4=1
end
if EV_User_2 == 1
    EV_User_2 = User_Connection_1;
    SOC2=1;
    User_Connection_No1=1
end
if EV_User_2 == 2
    EV_User_2 = User_Connection_2;
    SOC2=2;
    User_Connection_No2=1
end
if EV_User_2 == 3
    EV_User_2 = User_Connection_3;
    SOC2=3;
    User_Connection_No3=1
end
if EV_User_2 == 4
    EV_User_2 = User_Connection_4;
    SOC2=4;
    User_Connection_No4=1
end

% SOC calculations:

% Calculate charging time for EVs:
EV_power=(1-SOC)*Max_bat;
Hour_charge=EV_power/(P_charger*Charge_Eff); % Charging time EV

if Hour_charge < 1
    EV_User_1=[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    EV_User_2=[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
end
if Hour_charge < 2 & Hour_charge > 1  % Between 1 and 2 hours -> charge 2
    if SOC1==1
    EV_User_1=[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 ];
    end
```

```
    if SOC1== 2
    EV_User_1=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 1
    EV_User_2=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
end
if Hour_charge < 3 & Hour_charge > 2   % charge 3 hours
    if SOC1==1
    EV_User_1=[ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC1== 2
    EV_User_1=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 1
    EV_User_2=[ 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
end
if Hour_charge < 4 & Hour_charge > 3   % charge 4 hours
    if SOC1==1
    EV_User_1=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC1== 2
    EV_User_1=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
```

```
    if SOC2 == 1
    EV_User_2=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
end
if Hour_charge < 5 & Hour_charge > 4   % Charge 5 hours
    if SOC1==1
    EV_User_1=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC1== 2
    EV_User_1=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 1
    EV_User_2=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 ];
    end
end
if Hour_charge < 6 & Hour_charge > 5   % Charge 6 hours
    if SOC1==1
    EV_User_1=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC1== 2
    EV_User_1=[ 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 1
    EV_User_2=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
```

```
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 ];
    end
end
if Hour_charge < 7 & Hour_charge > 6   % Charge 7 hours
    if SOC1==1
    EV_User_1=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC1== 2
    EV_User_1=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC1== 3
    EV_User_1=[ 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC1== 4
    EV_User_1=[ 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 ];
    end
    if SOC2 == 1
    EV_User_2=[ 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 ];
    end
    if SOC2 == 2
    EV_User_2=[ 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ];
    end
    if SOC2 == 3
    EV_User_2=[ 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 ];
    end
    if SOC2 == 4
    EV_User_2=[ 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 ];
    end
end
% Initial values for further computation:
ONE=0; ONEONE=0; TWO=0; TWOTWO=0; THREE=0; THREETHREE=0;  FOUR=0;
FOURFOUR=0; FIVE=0; FIVEFIVE=0; SIX=0;  SIXSIX=0; SEVEN=0;
SEVENSEVEN=0; EIGHT=0; EIGHTEIGHT=0; NINE=0; NINENINE=0; TEN=0; TENT=0;
ELEVEN=0; ELEVENE=0; TWELVE=0; TWELVET=0; THERTEEN=0; THERTEENT=0;
FOURTEEN=0; FOURTEENF=0; FIFTEEN=0; FIFTEENF=0; SIXTEEN=0; SIXTEENS=0;
SEVENTEEN=0; SEVENTEENS=0; EIGHTEEN=0; EIGHTEENE=0; NINETEEN=0;
NINETEENN=0; TWENTY=0; TWENTYT=0;
% Initial end
% Link residence - Output : EV_residence_id + connection type
if EVs_in_use(1) == 1
EV_144503089 = EV_User_1
ONE=1;
elseif EVs_in_use(2) == 1
EV_144503089 = EV_User_2
ONEONE=1;
end
if EVs_in_use(1) == 2
    EV_144503070 = EV_User_1
    TWO=1;
elseif EVs_in_use(2) == 2
    EV_144503070 = EV_User_2
    TWOTWO=1;
end
if EVs_in_use(1) == 3
    EV_144502932 = EV_User_1
    THREE=1;
elseif EVs_in_use(2) == 3
    EV_144502932 = EV_User_2
    THREETHREE=1;
end
```

```
if EVs_in_use(1) == 4
    EV_144500731 = EV_User_1
    FOUR=1;
elseif EVs_in_use(2) == 4
    EV_144500731 = EV_User_2
    FOURFOUR=1;
end
if EVs_in_use(1) == 5
    EV_144500771 = EV_User_1
    FIVE=1;
elseif EVs_in_use(2) == 5
    EV_144500771 = EV_User_2
    FIVEFIVE=1;
end
if EVs_in_use(1) == 6
    EV_114500751 = EV_User_1
    SIX=1;
elseif EVs_in_use(2) == 6
    EV_114500751 = EV_User_2
    SIXSIX=1;
end
if EVs_in_use(1) == 7
    EV_144500615 = EV_User_1
    SEVEN=1;
elseif EVs_in_use(2) == 7
    EV_144500615 = EV_User_2
    SEVENSEVEN=1;
end
if EVs_in_use(1) == 8
    EV_144500741 = EV_User_1
    EIGTH=1;
elseif EVs_in_use(2) == 8
    EV_144500741 = EV_User_2
    EIGHTEIGHT=1;
end
if EVs_in_use(1) == 9
    EV_144500619 = EV_User_1
    NINE=1;
elseif EVs_in_use(2) == 9
    EV_144500619 = EV_User_2
    NINENINE=1;
end
if EVs_in_use(1) == 10
   EV_144500621 = EV_User_1
   TEN=1;
elseif EVs_in_use(2) == 10
    EV_144500621 = EV_User_2
    TENT=1;
end
if EVs_in_use(1) == 11
   EV_144590621 = EV_User_1
   ELEVEN=1;
elseif EVs_in_use(2) == 11
    EV_144590621 = EV_User_2
    ELEVENE=1;
end
if EVs_in_use(1) == 12
   EV_144500781 = EV_User_1
   TWELVE=1;
elseif EVs_in_use(2) == 12
    EV_144500781 = EV_User_2
```

```
    TWELVET=1;
end
if EVs_in_use(1) == 13
   EV_144500761 = EV_User_1
   THERTEEN=1;
elseif EVs_in_use(2) == 13
    EV_144500761 = EV_User_2
    THERTEENT=1;
end
if EVs_in_use(1) == 14
   EV_144500611 = EV_User_1
   FOURTEEN=1;
elseif EVs_in_use(2) == 14
   EV_144500611 = EV_User_2
   FOURTEENF=1;
end
if EVs_in_use(1) == 15
   EV_144500625 = EV_User_1
   FIFTEEN=1;
elseif EVs_in_use(2) == 15
   EV_144500625 = EV_User_2
   FIFTEENF=1;
end
if EVs_in_use(1) == 16
   EV_144500721 = EV_User_1
   SIXTEEN=1;
elseif EVs_in_use(2) == 16
   EV_144500721 = EV_User_2
   SIXTEENS=1;
end
if EVs_in_use(1) == 17
   EV_144500617 = EV_User_1
   SEVENTEEN=1;
elseif EVs_in_use(2) == 17
   EV_144500617 = EV_User_2
   SEVENTEENS=1;
end
if EVs_in_use(1) == 18
  EV_144500622 = EV_User_1
  EIGHTEEN=1;
elseif EVs_in_use(2) == 18
  EV_144500622 = EV_User_2
  EIGHTEENE=1;
end
if EVs_in_use(1) == 19
  EV_144500791 = EV_User_1
  NINETEEN=1;
elseif EVs_in_use(2) == 19
  EV_144500791 = EV_User_2
  NINETEENN=1;
end
if EVs_in_use(1) == 20
  EV_144500711 = EV_User_1
  TWENTY=1;
elseif EVs_in_use(2) == 20
  EV_144500711 = EV_User_2
  TWENTYT=1;
end

% Load Calculations:  --------------------------------------------------
H_init = 1; %Load hour for power calculations
```

```
H=H_init;
while H < 25;

Load = Load_Winter(H);
%Load=Load_Summer(H);
%Load=Load_Sping(H);
%Load=Load_Autumn(H)

EV_Load_1 = EV_User_1(H)*P_charger;
EV_Load_2 = EV_User_2(H)*P_charger;

%Calculate total load of each residence:
% The numbers in writing represents logic from the "Linked residences"
if ONE == 1
EV_R_144503089 = Load + EV_Load_1 ;
elseif    ONEONE == 1
    EV_R_144503089 = Load + EV_Load_2 ;
else EV_R_144503089 = Load ;
end
if TWO == 1
EV_R_144503070 = Load + EV_Load_1 ;
elseif    TWOTWO == 1
    EV_R_144503070 = Load + EV_Load_2 ;
else EV_R_144503070 = Load ;
end
if THREE==1
EV_R_144502932 = Load + EV_Load_1;
elseif    THREETHREE==1
    EV_R_144502932 = Load + EV_Load_2 ;
else EV_R_144502932 = Load ;
end
if FOUR==1
EV_R_144500731 = Load + EV_Load_1 ;
elseif    FOURFOUR==1
    EV_R_144500731 = Load + EV_Load_2 ;
else EV_R_144500731 = Load ;
end
if FIVE == 1
EV_R_144500771 = Load + EV_Load_1 ;
elseif    FIVEFIVE == 1
    EV_R_144500771 = Load + EV_Load_2 ;
else EV_R_144500771 = Load ;
end
if SIX==1
EV_R_114500751 = Load + EV_Load_1 ;
elseif   SIXSIX==1
    EV_R_114500751 = Load + EV_Load_2 ;
else EV_R_114500751 = Load ;
end
    if SEVEN==1
EV_R_144500615 = Load + EV_Load_1 ;
    elseif    SEVENSEVEN==1
    EV_R_144500615 = Load + EV_Load_2 ;
else EV_R_144500615 = Load ;
    end
        if EIGHT==1
EV_R_144500741 = Load + EV_Load_1 ;
        elseif    EIGHTEIGHT==1
    EV_R_144500741 = Load + EV_Load_2 ;
else EV_R_144500741 = Load ;
        end
```

```
        if NINE==1
EV_R_144500619= Load + EV_Load_1 ;
        elseif   NINENINE==1
    EV_R_144500619 = Load + EV_Load_2 ;
else EV_R_144500619 = Load ;
    end
        if TEN==1
EV_R_144500621 = Load + EV_Load_1 ;
        elseif    TENT==1
    EV_R_144500621 = Load + EV_Load_2 ;
else EV_R_144500621 = Load ;
        end
    if ELEVEN==1
EV_R_144590621 = Load + EV_Load_1 ;
    elseif    ELEVENE==1
   EV_R_144590621 = Load + EV_Load_2 ;
else EV_R_144590621 = Load ;
    end
    if TWELVE==1
EV_R_144500781 = Load + EV_Load_1 ;
    elseif   TWELVET==1
   EV_R_144500781 = Load + EV_Load_2 ;
else EV_R_144500781 = Load ;
    end
    if THERTEEN==1
EV_R_144500761 = Load + EV_Load_1 ;
    elseif    THERTEENT==1
   EV_R_144500761 = Load + EV_Load_2 ;
else EV_R_144500761 = Load ;
    end
    if FOURTEEN==1
EV_R_144500611 = Load + EV_Load_1 ;
    elseif    FOURTEENF==1
   EV_R_144500611 = Load + EV_Load_2 ;
else EV_R_144500611 = Load ;
    end
    if FIFTEEN==1
EV_R_144500625 = Load + EV_Load_1 ;
    elseif    FIFTEENF==1
   EV_R_144500625 = Load + EV_Load_2 ;
else EV_R_144500625 = Load ;
    end
    if SIXTEEN==1
EV_R_144500721 = Load + EV_Load_1 ;
    elseif    SIXTEENS==1
   EV_R_144500721 = Load + EV_Load_2 ;
else EV_R_144500721 = Load ;
    end
    if SEVENTEEN==1
EV_R_144500617 = Load + EV_Load_1 ;
    elseif    SEVENTEENS==1
   EV_R_144500617 = Load + EV_Load_2 ;
else EV_R_144500617 = Load ;
    end
    if EIGHTEEN==1
EV_R_144500622 = Load + EV_Load_1 ;
    elseif    EIGHTEENE==1
   EV_R_144500622 = Load + EV_Load_2 ;
else EV_R_144500622 = Load ;
    end
    if NINETEEN==1
```

```
EV_R_144500791 = Load + EV_Load_1 ;
    elseif    NINETEENN==1
   EV_R_144500791 = Load + EV_Load_2 ;
else EV_R_144500791 = Load ;
    end
    if TWENTY==1
EV_R_144500711 = Load + EV_Load_1;
    elseif    TWENTYT==1
   EV_R_144500711 = Load + EV_Load_2;
else EV_R_144500711 = Load ;
    end
    % Gather the loads in a matrix
    EV_Loads = [EV_R_144503089 EV_R_144503070 EV_R_144502932 EV_R_144500731
EV_R_144500771 EV_R_114500751 EV_R_144500615 EV_R_144500741 EV_R_144500619
EV_R_144500621 EV_R_144590621 EV_R_144500781 EV_R_144500761 EV_R_144500611
EV_R_144500625 EV_R_144500721 EV_R_144500617 EV_R_144500622 EV_R_144500791
EV_R_144500711];
    P_load_tot(H) = sum(EV_Loads); % Totoal load every hour from 00.00 to
23.00

    % Hourly based load matrix for each residence:
    L_144503089(H) =EV_Loads(1);
    L_144503070(H) =EV_Loads(2);
    L_144502932(H) =EV_Loads(3);
    L_144500731(H) =EV_Loads(4);
    L_144500771(H) =EV_Loads(5);
    L_114500751(H) =EV_Loads(6);
    L_144500615(H) =EV_Loads(7);
    L_144500741(H) =EV_Loads(8);
    L_144500619(H) =EV_Loads(9);
    L_144500621(H) =EV_Loads(10);
    L_144590621(H) =EV_Loads(11);
    L_144500781(H) =EV_Loads(12);
    L_144500761(H) =EV_Loads(13);
    L_144500611(H) =EV_Loads(14);
    L_144500625(H) =EV_Loads(15);
    L_144500721(H) =EV_Loads(16);
    L_144500617(H) =EV_Loads(17);
    L_144500622(H) =EV_Loads(18);
    L_144500791(H) =EV_Loads(19);
    L_144500711(H) =EV_Loads(20);

    H=H+1;    %Gives data for all 24 hours
end

% Data from loop: ------------------------------------------------
P_load_tot % Total load for hourly based [1-24]
% Hourly based load from every residence (outside total load loop) :
L_144503089; L_144503070; L_144502932; L_144500731; L_144500771;
L_114500751; L_144500615; L_144500741; L_144500619; L_144500621;
L_144590621; L_144500781; L_144500761; L_144500611; L_144500625;
L_144500721; L_144500617; L_144500622; L_144500791; L_144500711;
```

Appendix C – MATLAB script objective 2

```
% Smart Load Management for electric vehicles
% ----------------------------------------------------------------------
% SLM Algorithm objective 2
% Master Thesis Spring 2015
% Author: Stian Namtvedt Gjelsvik
% Email: stian2803@gmail.com
% Version: 1.00
% Last update: 23.03.2015
% ----------------------------------------------------------------------
% Description:
% This script is a suggested smart load management algorithm, that
% will schedule charging of EVs in a specific order customized for a
% typical Norwegian low voltage distribution network. The algorithm takes
% care of the constraints in the distribution network when scheduling
% charging of EVs.
% ---------------------------------------------------------------------
% Smart load management (SLM) -------------------------------------
% Input:
S_transformer = 150;
CosPhi = 0.9;   %  External value from network
Phi = 25.8419327; % CosPhi transformer lossless = CosPhi network
P_transformer = S_transformer*CosPhi;
U_Transformer = 238;
%Initial voltage condition: (for calculation)
U_init=230;
U_144503089=U_init; U_144503070=U_init; U_144502932=U_init;
U_144500731=U_init; U_144500771=U_init; U_114500751=U_init;
U_144500615=U_init; U_144500741=U_init; U_144500619=U_init;
U_144500621=U_init; U_144590621=U_init; U_144500781=U_init;
U_144500761=U_init; U_144500611=U_init; U_144500625=U_init;
U_144500721=U_init; U_144500617=U_init; U_144500622=U_init;
U_144500791=U_init; U_144500711=U_init;

% Constraints: -----------------------------------------------
% v_min <= v_n <= v_max
v_min=207;
v_max=253;

% I_b <= I_z  (Load current <= cable conductivity)

% P_load_tot <= P_transformer

% Line data: ------------------------------------------------------
% Lines feeding the residence: Z = R + jX
R_144503089= [0.0432 0.0101];
R_144503070= [0.0661 0.0103];
R_144502932= [0.0263 0.0032];
R_144500731= [0.0266 0.0033];
R_144500771= [0.0216 0.0024];
R_114500751= [0.0192 0.0013];
R_144500615= [0.0156 0.0011];
R_144500741= [0.0144 0.0095];
R_144500619= [0.0072 0.0005];
R_144500621= [0.0504 0.0034];
R_144590621= [0.0300 0.0021];
R_144500781= [0.5304 0.0036];
R_144500761= [0.0300 0.0021];
```

```
R_144500611= [0.0120 0.0008];
R_144500625= [0.0132 0.0009];
R_144500721= [0.0227 0.0017];
R_144500617= [0.0216 0.0015];
R_144500622= [0.1884 0.0129];
R_144500791= [0.3216 0.0220];
R_144500711= [0.1884 0.0129];

% Feeding lines: Z = R + jX
F_80955_0= [0.0011 0.0006];
F_80955_0A= [0.0010 0.0006];
F_80955_5= [0.0199 0.0024];
F_80955_16= [0.0314 0.0074];
F_80955_17= [0.0600 0.0041];
F_80955_19= [0.0244 0.0029];
F_80955_22= [0.0853 0.0102];
F_80955_23= [0.0444 0.0030];
F_80955_A1= [0.0036 0.0002];
F_80955_24= [0.0552 0.0038];
F_80955_26= [0.1032 0.0071];
F_80955_18= [0.2124 0.0145];
F_80955_1= [0.0096 0.0023];
F_80955_1A= [0.0096 0.0023];
F_80955_9= [0.0493 0.0117];
F_80955_10= [0.0122 0.0029];
F_80955_14= [0.1308 0.0157];
F_80955_29= [0.0224 0.0027];
F_80955_4= [0.0769 0.0092];

% Total - Total impedance to the residences. Z = R + jX
TOT_144503089= F_80955_0 + F_80955_1 + F_80955_16 + F_80955_19 + F_80955_22
+ F_80955_23 + R_144503089;
TOT_144503070= F_80955_0 + F_80955_1 + F_80955_16 + F_80955_19 + F_80955_22
+ F_80955_23 + R_144503070;
TOT_144502932= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144502932;
TOT_144500731= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144500731;
TOT_144500771= F_80955_1 + F_80955_0 + R_144500771;
TOT_114500751= F_80955_18 + F_80955_1 + F_80955_0 + R_114500751;
TOT_144500615= F_80955_17 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500615;
TOT_144500741= F_80955_17 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500741;
TOT_144500619= F_80955_24 + F_80955_23 + F_80955_22 + F_80955_19 +
F_80955_16 + F_80955_1 + F_80955_0 + R_144500619;
TOT_144500621= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144500621;
TOT_144590621= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144590621;
TOT_144500781= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144500781;
TOT_144500761= F_80955_19 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500761;
TOT_144500611= F_80955_19 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500611;
TOT_144500625= F_80955_14 + F_80955_10 + F_80955_9 + F_80955_1A +
F_80955_0A + R_144500625;
TOT_144500721= F_80955_9 + F_80955_1A + F_80955_0A + R_144500721;
TOT_144500617= F_80955_9 + F_80955_1A + F_80955_0A + R_144500617;
```

```
TOT_144500622= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500622;
TOT_144500791= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500791;
TOT_144500711= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500771;

% Cable conductivity:
I_z_144503089=220;
I_z_144503070=150;
I_z_80955_0=435;
I_z_80955_0A=435;
I_z_144502932=150;
I_z_80955_5=150;
I_z_80955_16=280;
I_z_80955_17=115;
I_z_144500615=115;
I_z_80955_19=180;
I_z_80955_22=180;
I_z_80955_23=115;
I_z_80955_A1=115;
I_z_80955_24=115;
I_z_144500619=115;
I_z_80955_26=115;
I_z_144500621=115;
I_z_144590621=115;
I_z_144500611=115;
I_z_80955_18=115;
I_z_80955_1 =280;
I_z_80955_1A=280;
I_z_80955_9=280;
I_z_80955_10=280;
I_z_80955_14=180;
I_z_144500625=115;
I_z_80955_29=180;
I_z_144500617=115;
I_z_80955_4 =180;
I_z_144500622=115;
I_z_144500731=115;
I_z_144500771=115;
I_z_114500751=115;
I_z_144500741=115;
I_z_144500781=115;
I_z_144500761=115;
I_z_144500721=115;
I_z_144500791=115;
I_z_144500711=115;

% Calculations of constraints : ---------------------------------

%Current:
I_b_144503089 = (1000 * L_144503089) / (U_144503089 * CosPhi* sqrt(3)); %kW
to W
% Delta voltage:
D_144503089 = sqrt(3)*(TOT_144503089(1)*I_b_144503089*cosd(Phi) +
TOT_144503089(2)*I_b_144503089*sind(Phi));
% Voltage at residence:
U_144503089 = U_Transformer - D_144503089;
%Current:
I_b_144503070 = (1000 * L_144503070) / (U_144503070 * CosPhi* sqrt(3));
% Delta voltage:
```

```
D_144503070 = sqrt(3)*(TOT_144503070(1)*I_b_144503070*cosd(Phi) +
TOT_144503070(2)*I_b_144503070*sind(Phi));
% Voltage at residence:
U_144503070 = U_Transformer - D_144503070;
%Current:
I_b_144502932 = (1000 * L_144502932) / (U_144502932 * CosPhi* sqrt(3));
% Delta voltage:
D_144502932 = sqrt(3)*(TOT_144502932(1)*I_b_144502932*cosd(Phi) +
TOT_144502932(2)*I_b_144502932*sind(Phi));
% Voltage at residence:
U_144502932 = U_Transformer - D_144502932;
%Current:
I_b_144500731 = (1000 * L_144500731) / (U_144500731 * CosPhi* sqrt(3));
% Delta voltage:
D_144500731 = sqrt(3)*(TOT_144500731(1)*I_b_144500731*cosd(Phi) +
TOT_144500731(2)*I_b_144500731*sind(Phi));
% Voltage at residence:
U_144500731 = U_Transformer - D_144500731;
%Current:
I_b_144500771 = (1000 * L_144500771) / (U_144500771 * CosPhi* sqrt(3));
% Delta voltage:
D_144500771 = sqrt(3)*(TOT_144500771(1)*I_b_144500771*cosd(Phi) +
TOT_144500771(2)*I_b_144500771*sind(Phi));
% Voltage at residence:
U_144500771 = U_Transformer - D_144500771;
%Current:
I_b_114500751 = (1000 * L_114500751) / (U_114500751 * CosPhi* sqrt(3));
% Delta voltage:
D_114500751 = sqrt(3)*(TOT_114500751(1)*I_b_114500751*cosd(Phi) +
TOT_114500751(2)*I_b_114500751*sind(Phi));
% Voltage at residence:
U_114500751 = U_Transformer - D_114500751;
%Current:
I_b_144500615 = (1000 * L_144500615) / (U_144500615 * CosPhi* sqrt(3));
% Delta voltage:
D_144500615 = sqrt(3)*(TOT_144500615(1)*I_b_144500615*cosd(Phi) +
TOT_144500615(2)*I_b_144500615*sind(Phi));
% Voltage at residence:
U_144500615 = U_Transformer - D_144500615;
%Current:
I_b_144500741 = (1000 * L_144500741) / (U_144500741 * CosPhi* sqrt(3));
% Delta voltage:
D_144500741 = sqrt(3)*(TOT_144500741(1)*I_b_144500741*cosd(Phi) +
TOT_144500741(2)*I_b_144500741*sind(Phi));
% Voltage at residence:
U_144500741 = U_Transformer - D_144500741;
%Current:
I_b_144500619 = (1000 * L_144500619) / (U_144500619 * CosPhi* sqrt(3));
% Delta voltage:
D_144500619 = sqrt(3)*(TOT_144500619(1)*I_b_144500619*cosd(Phi) +
TOT_144500619(2)*I_b_144500619*sind(Phi));
% Voltage at residence:
U_144500619 = U_Transformer - D_144500619;
%Current:
I_b_144500621 = (1000 * L_144500621) / (U_144500621 * CosPhi* sqrt(3));
% Delta voltage:
D_144500621 = sqrt(3)*(TOT_144500621(1)*I_b_144500621*cosd(Phi) +
TOT_144500621(2)*I_b_144500621*sind(Phi));
% Voltage at residence:
U_144500621 = U_Transformer - D_144500621;
%Current:
```

```
I_b_144590621 = (1000 * L_144590621) / (U_144590621 * CosPhi* sqrt(3));
% Delta voltage:
D_144590621 = sqrt(3)*(TOT_144590621(1)*I_b_144590621*cosd(Phi) +
TOT_144590621(2)*I_b_144590621*sind(Phi));
% Voltage at residence:
U_144590621 = U_Transformer - D_144590621;
%Current:
I_b_144500781 = (1000 * L_144500781) / (U_144500781 * CosPhi* sqrt(3));
% Delta voltage:
D_144500781 = sqrt(3)*(TOT_144500781(1)*I_b_144500781*cosd(Phi) +
TOT_144500781(2)*I_b_144500781*sind(Phi));
% Voltage at residence:
U_144500781 = U_Transformer - D_144500781;
%Current:
I_b_144500761 = (1000 * L_144500761) / (U_144500761 * CosPhi* sqrt(3));
% Delta voltage:
D_144500761 = sqrt(3)*(TOT_144500761(1)*I_b_144500761*cosd(Phi) +
TOT_144500761(2)*I_b_144500761*sind(Phi));
% Voltage at residence:
U_144500761 = U_Transformer - D_144500761;
%Current:
I_b_144500611 = (1000 * L_144500611) / (U_144500611 * CosPhi* sqrt(3));
% Delta voltage:
D_144500611 = sqrt(3)*(TOT_144500611(1)*I_b_144500611*cosd(Phi) +
TOT_144500611(2)*I_b_144500611*sind(Phi));
% Voltage at residence:
U_144500611 = U_Transformer - D_144500611;
%Current:
I_b_144500625 = (1000 * L_144500625) / (U_144500625 * CosPhi* sqrt(3));
% Delta voltage:
D_144500625 = sqrt(3)*(TOT_144500625(1)*I_b_144500625*cosd(Phi) +
TOT_144500625(2)*I_b_144500625*sind(Phi));
% Voltage at residence:
U_144500625 = U_Transformer - D_144500625;
%Current:
I_b_144500721 = (1000 * L_144500721) / (U_144500721 * CosPhi* sqrt(3));
% Delta voltage:
D_144500721 = sqrt(3)*(TOT_144500721(1)*I_b_144500721*cosd(Phi) +
TOT_144500721(2)*I_b_144500721*sind(Phi));
% Voltage at residence:
U_144500721 = U_Transformer - D_144500721;
%Current:
I_b_144500617 = (1000 * L_144500617) / (U_144500617 * CosPhi* sqrt(3));
% Delta voltage:
D_144500617 = sqrt(3)*(TOT_144500617(1)*I_b_144500617*cosd(Phi) +
TOT_144500617(2)*I_b_144500617*sind(Phi));
% Voltage at residence:
U_144500617 = U_Transformer - D_144500617;
%Current:
I_b_144500622 = (1000 * L_144500622) / (U_144500622 * CosPhi * sqrt(3));
% Delta voltage:
D_144500622 = sqrt(3)*(TOT_144500622(1)*I_b_144500622*cosd(Phi) +
TOT_144500622(2)*I_b_144500622*sind(Phi));
% Voltage at residence:
U_144500622 = U_Transformer - D_144500622;
%Current:
I_b_144500791 = (1000 * L_144500791) / (U_144500791 * CosPhi);
% Delta voltage:
D_144500791 = sqrt(3)*(TOT_144500791(1)*I_b_144500791*cosd(Phi) +
TOT_144500791(2)*I_b_144500791*sind(Phi));
% Voltage at residence:
```

```
U_144500791 = U_Transformer - D_144500791;
%Current:
I_b_144500711 = (1000 * L_144500711) / (U_144500711 * CosPhi * sqrt(3));
% Delta voltage:
D_144500711 = sqrt(3)*(TOT_144500711(1)*I_b_144500711*cosd(Phi) +
TOT_144500711(2)*I_b_144500711*sind(Phi));
% Voltage at residence:
U_144500711 = U_Transformer - D_144500711;

% Feeder line currents:
I_b_80955_0= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771 ;
I_b_80955_0A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_5= I_b_144500711 + I_b_144500622 + I_b_144500791;
I_b_80955_16= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741;
I_b_80955_17= I_b_144500615 + I_b_144500741 ;
I_b_80955_19= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761;
I_b_80955_22= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781;
I_b_80955_23= I_b_144503070 + I_b_144503089 + I_b_144500619;
I_b_80955_A1= I_b_144503070 + I_b_144503089;
I_b_80955_24= I_b_144500619;
I_b_80955_26= I_b_144590621 + I_b_144500621 + I_b_144500781;
I_b_80955_18= I_b_114500751;
I_b_80955_1= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771 ;
I_b_80955_1A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_9= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617;
I_b_80955_10= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_14= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_29= I_b_144502932 + I_b_144500731;
I_b_80955_4= I_b_144500711 + I_b_144500622 + I_b_144500791;

% While loop for each residence for repsentation reason:
N = 1; % Load hour for constrain calculations
while N<25
    % ONE/ONEONE represents if the residence have an EV connected
    if  ONE==1 & EV_User_1(N)==1
    USE_144503089=1 ;                % Residence with active EV
    elseif  ONEONE==1 & EV_User_2(N)==1
    USE_144503089=1;                 % Residence with active EV
    else USE_144503089=0;            % Residence with no EV or inactive EV
    end
    % Line, feeder, voltage, transformer constraints:
    if I_b_144503089(N) <= I_z_144503089 & I_b_80955_A1(N) <= I_z_80955_A1
& I_b_80955_23(N) <= I_z_80955_23 & I_b_80955_22(N) <= I_z_80955_22 &
I_b_80955_19(N) <= I_z_80955_19 & I_b_80955_16(N) <= I_z_80955_16 &
I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0 &
U_144503089(N) >= v_min & U_144503089(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144503089==1
    EX_144503089 = 1;                % Execute charging this hour
    else
    EX_144503089 = 0;                % No charging due to constraints
```

```
    end
    Ma_144503089(N) = EX_144503089;  % Puts values in a matrix
    N = N+1;
end
N=1;
while N<25
if  TWO==1 & EV_User_1(N)==1
    USE_144503070=1 ;
    elseif  TWOTWO==1 & EV_User_2(N)==1
    USE_144503070=1;
    else USE_144503070=0;
end % Line, feeder, voltage, transformer constraints:
if I_b_144503070(N) <= I_z_144503070 & I_b_80955_A1(N) <= I_z_80955_A1 &
I_b_80955_23(N) <= I_z_80955_23 & I_b_80955_22(N) <= I_z_80955_22 &
I_b_80955_19(N) <= I_z_80955_19 & I_b_80955_16(N) <= I_z_80955_16 &
I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0 &
U_144503070(N) >= v_min & U_144503070(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144503070==1
    EX_144503070 = 1;        % Execute charging this hour
else
    EX_144503070 = 0;
end
Ma_144503070(N) = EX_144503070;
N=N+1;
end
N=1;
while N<25
if  THREE==1 & EV_User_1(N)==1
    USE_144502932=1;
    elseif  THREETHREE==1 && EV_User_2(N)==1
    USE_144502932=1;
    else USE_144502932=0;
end % Line, feeder, voltage, transformer constraints:
if I_b_144502932(N) <= I_z_144502932 & I_b_80955_29(N) <= I_z_80955_29 &
I_b_80955_14(N) <= I_z_80955_14 & I_b_80955_10(N) <= I_z_80955_10 &
I_b_80955_9(N) <= I_z_80955_9 & I_b_80955_1A(N) <= I_z_80955_1A &
I_b_80955_0A(N) <= I_z_80955_0A & U_144502932(N) >= v_min & U_144502932(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144502932==1
    EX_144502932 = 1;       % Execute charging this hour
else
    EX_144502932 = 0;
end
Ma_144502932(N) = EX_144502932;
N=N+1;
end
N=1;
while N<25
if  FOUR==1 & EV_User_1(N)==1
    USE_144500731=1 ;
    elseif  FOURFOUR==1 & EV_User_2(N)==1
    USE_144500731=1;
    else USE_144500731=0;
end  % Line, feeder, voltage, transformer constraints:
if I_b_144500731(N) <= I_z_144500731 & I_b_80955_29(N) <= I_z_80955_29 &
I_b_80955_14(N) <= I_z_80955_14 & I_b_80955_10(N) <= I_z_80955_10 &
I_b_80955_9(N) <= I_z_80955_9 & I_b_80955_1A(N) <= I_z_80955_1A &
I_b_80955_0A(N) <= I_z_80955_0A & U_144500731(N) >= v_min & U_144500731(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500731==1
    EX_144500731 = 1;        % Execute charging this hour
else
    EX_144500731 = 0;
```

```
end
Ma_144500731(N) = EX_144500731;
N=N+1;
end
N=1;
while N<25
if  FIVE==1 & EV_User_1(N)==1
    USE_144500771=1 ;
    elseif  FIVEFIVE==1 & EV_User_2(N)==1
    USE_144500771=1;
    else USE_144500771=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500771(N) <= I_z_144500771 & I_b_80955_18(N) <= I_z_80955_18 &
I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0 &
U_144500771(N) >= v_min & U_144500771(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144500771==1
    EX_144500771 = 1;        % Execute charging this hour
else
    EX_144500771 = 0;
end
Ma_144500771(N) = EX_144500771;
N=N+1;
end
N=1;
while N<25
if  SIX==1 & EV_User_1(N)==1
    USE_114500751=1 ;
    elseif  SIXSIX==1 & EV_User_2(N)==1
    USE_114500751=1;
    else USE_114500751=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_114500751(N) <= I_z_114500751 & I_b_80955_18(N) <= I_z_80955_18 &
I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0 &
U_114500751(N) >= v_min & U_114500751(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_114500751==1
    EX_114500751 = 1;        % Execute charging this hour
else
    EX_114500751 = 0;
end
Ma_114500751(N) = EX_114500751;
N=N+1;
end
N=1;
while N<25
if  SEVEN==1 & EV_User_1(N)==1
    USE_144500615=1 ;
    elseif  SEVENSEVEN==1 & EV_User_2(N)==1
    USE_144500615=1;
    else USE_144500615=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500615(N) <= I_z_144500615 & I_b_80955_17(N) <= I_z_80955_17 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 & U_144500615(N) >= v_min & U_144500615(N) <=
v_max & P_load_tot(N) <= P_transformer & USE_144500615==1
    EX_144500615 = 1;        % Execute charging this hour
else
    EX_144500615 = 0;
end
Ma_144500615(N) = EX_144500615;
N=N+1;
end
```

```
N=1;
while N<25
if  EIGHT==1 & EV_User_1(N)==1
    USE_144500741=1 ;
    elseif  EIGHTEIGHT==1 & EV_User_2(N)==1
    USE_144500741=1;
    else USE_144500741=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500741(N) <= I_z_144500741 & I_b_80955_17(N) <= I_z_80955_17 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 &  U_144500741(N) >= v_min & U_144500741(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500741==1
    EX_144500741 = 1;         % Execute charging this hour
else
    EX_144500741 = 0;
end
Ma_144500741(N) = EX_144500741;
N=N+1;
end
N=1;
while N<25
if  NINE==1 & EV_User_1(N)==1
    USE_144500619=1 ;
    elseif  NINENINE==1 & EV_User_2(N)==1
    USE_144500619=1;
    else USE_144500619=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500619(N) <= I_z_144500619 & I_b_80955_24(N) <= I_z_80955_24 &
I_b_80955_23(N) <= I_z_80955_23 & I_b_80955_22(N) <= I_z_80955_22 &
I_b_80955_19(N) <= I_z_80955_19 & I_b_80955_16(N) <= I_z_80955_16 &
I_b_80955_1(N) <= I_z_80955_1 & I_b_80955_0(N) <= I_z_80955_0 &
U_144500619(N) >= v_min & U_144500619(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144500619==1
    EX_144500619 = 1;         % Execute charging this hour
else
    EX_144500619 = 0;
end
Ma_144500619(N) = EX_144500619;
N=N+1;
end
N=1;
while N<25
if  TEN==1 & EV_User_1(N)==1
    USE_144500621=1 ;
    elseif  TENT==1 & EV_User_2(N)==1
    USE_144500621=1;
    else USE_144500621=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500621(N) <= I_z_144500621 & I_b_80955_26(N) <= I_z_80955_26 &
I_b_80955_22(N) <= I_z_80955_22 & I_b_80955_19(N) <= I_z_80955_19 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 &  U_144500621(N) >= v_min & U_144500621(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500621==1
    EX_144500621 = 1;         % Execute charging this hour
else
    EX_144500621 = 0;
end
Ma_144500621(N) = EX_144500621;
N=N+1;
end
N=1;
```

```
while N<25
if   ELEVEN==1 & EV_User_1(N)==1
     USE_144590621=1 ;
     elseif  ELEVENE==1 & EV_User_2(N)==1
     USE_144590621=1;
     else USE_144590621=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144590621(N) <= I_z_144590621 & I_b_80955_26(N) <= I_z_80955_26 &
I_b_80955_22(N) <= I_z_80955_22 & I_b_80955_19(N) <= I_z_80955_19 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 & U_144590621(N) >= v_min & U_144590621(N) <=
v_max & P_load_tot(N) <= P_transformer & USE_144590621==1
     EX_144590621 = 1;         % Execute charging this hour
else
     EX_144590621 = 0;
end
Ma_144590621(N) = EX_144590621;
N=N+1;
end
N=1;
while N<25
if   TWELVE==1 & EV_User_1(N)==1
     USE_144500781=1 ;
     elseif  TWELVET==1 & EV_User_2(N)==1
     USE_144500781=1;
     else USE_144500781=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500781(N) <= I_z_144500781 & I_b_80955_26(N) <= I_z_80955_26 &
I_b_80955_22(N) <= I_z_80955_22 & I_b_80955_19(N) <= I_z_80955_19 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 & U_144500781(N) >= v_min & U_144500781(N) <=
v_max & P_load_tot(N) <= P_transformer & USE_144500781==1
     EX_144500781 = 1;         % Execute charging this hour
else
     EX_144500781 = 0;
end
Ma_144500781(N) = EX_144500781;
N=N+1;
end
N=1;
while N<25
if   THERTEEN==1 & EV_User_1(N)==1
     USE_144500761=1 ;
     elseif  THERTEENT==1 & EV_User_2(N)==1
     USE_144500761=1;
     else USE_144500761=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500761(N) <= I_z_144500761 & I_b_80955_19(N) <= I_z_80955_19 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 & U_144500761(N) >= v_min & U_144500761(N) <=
v_max & P_load_tot(N) <= P_transformer & USE_144500761==1
     EX_144500761 = 1;         % Execute charging this hour
else
     EX_144500761 = 0;
end
Ma_144500761(N) = EX_144500761;
N=N+1;
end
N=1;
while N<25
if   FOURTEEN==1 & EV_User_1(N)==1
```

```
    USE_144500611=1 ;
    elseif  FOURTEENF==1 & EV_User_2(N)==1
    USE_144500611=1;
    else USE_144500611=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500611(N) <= I_z_144500611 & I_b_80955_19(N) <= I_z_80955_19 &
I_b_80955_16(N) <= I_z_80955_16 & I_b_80955_1(N) <= I_z_80955_1 &
I_b_80955_0(N) <= I_z_80955_0 & U_144500611(N) >= v_min & U_144500611(N) <=
v_max & P_load_tot(N) <= P_transformer & USE_144500611==1
    EX_144500611 = 1;        % Execute charging this hour
else
    EX_144500611 = 0;
end
Ma_144500611(N) = EX_144500611;
N=N+1;
end
N=1;
while N<25
if  FIFTEEN==1 & EV_User_1(N)==1
    USE_144500625=1 ;
    elseif  FIFTEENF==1 & EV_User_2(N)==1
    USE_144500625=1;
    else USE_144500625=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500625(N) <= I_z_144500625 & I_b_80955_14(N) <= I_z_80955_14 &
I_b_80955_10(N) <= I_z_80955_10 & I_b_80955_9(N) <= I_z_80955_9 &
I_b_80955_1A(N) <= I_z_80955_1A & I_b_80955_0A(N) <= I_z_80955_0A &
U_144500625(N) >= v_min & U_144500625(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144500625==1
    EX_144500625 = 1;        % Execute charging this hour
else
    EX_144500625 = 0;
end
Ma_144500625(N) = EX_144500625;
N=N+1;
end
N=1;
while N<25
if  SIXTEEN==1 & EV_User_1(N)==1
    USE_144500721=1 ;
    elseif  SIXTEENS==1 & EV_User_2(N)==1
    USE_144500721=1;
    else USE_144500721=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500721(N) <= I_z_144500721 & I_b_80955_9(N) <= I_z_80955_9 &
I_b_80955_1A(N) <= I_z_80955_1A & I_b_80955_0A(N) <= I_z_80955_0A &
U_144500721(N) >= v_min & U_144500721(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144500721==1
    EX_144500721 = 1;        % Execute charging this hour
else
    EX_144500721 = 0;
end
Ma_144500721(N) = EX_144500721;
N=N+1;
end
N=1;
while N<25
if  SEVENTEEN==1 & EV_User_1(N)==1
    USE_144500617=1 ;
    elseif  SEVENTEENS==1 & EV_User_2(N)==1
    USE_144500617=1;
```

```
    else USE_144500617=0;
end
if I_b_144500617(N) <= I_z_144500617 & I_b_80955_9(N) <= I_z_80955_9 &
I_b_80955_1A(N) <= I_z_80955_1A & I_b_80955_0A(N) <= I_z_80955_0A &
U_144500617(N) >= v_min & U_144500617(N) <= v_max & P_load_tot(N) <=
P_transformer & USE_144500617==1
    EX_144500617 = 1;        % Execute charging this hour
else   % Line, feeder, voltage, transformer constraints:
    EX_144500617 = 0;
end
Ma_144500617(N) = EX_144500617;
N=N+1;
end
N=1;
while N<25
if  EIGHTEEN==1 & EV_User_1(N)==1
    USE_144500622=1 ;
    elseif  EIGHTEENE==1 & EV_User_2(N)==1
    USE_144500622=1;
    else USE_144500622=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500622(N) <= I_z_144500622 & I_b_80955_5(N) <= I_z_80955_5 &
I_b_80955_4(N) <= I_z_80955_4 & I_b_80955_1A(N) <= I_z_80955_1A &
I_b_80955_0A(N) <= I_z_80955_0A & U_144500622(N) >= v_min & U_144500622(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500622==1
    EX_144500622 = 1;        % Execute charging this hour
else
    EX_144500622 = 0;
end
Ma_144500622(N) = EX_144500622;
N=N+1;
end
N=1;
while N<25
if  NINETEEN==1 & EV_User_1(N)==1
    USE_144500791=1 ;
    elseif  NINETEENN==1 & EV_User_2(N)==1
    USE_144500791=1;
    else USE_144500791=0;
end   % Line, feeder, voltage, transformer constraints:
if I_b_144500791(N) <= I_z_144500791 & I_b_80955_5(N) <= I_z_80955_5 &
I_b_80955_4(N) <= I_z_80955_4 & I_b_80955_1A(N) <= I_z_80955_1A &
I_b_80955_0A(N) <= I_z_80955_0A & U_144500791(N) >= v_min & U_144500791(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500791==1
    EX_144500791 = 1;        % Execute charging this hour
else
    EX_144500791 = 0;
end
Ma_144500791(N) = EX_144500791;
N=N+1;
end
N=1;
while N<25
if  TWENTY==1 & EV_User_1(N)==1
    USE_144500711=1 ;
    elseif  TWENTYT==1 & EV_User_2(N)==1
    USE_144500711=1;
    else USE_144500711=0;
end    % Line, feeder, voltage, transformer constraints:
if I_b_144500711(N) <= I_z_144500711 & I_b_80955_5(N) <= I_z_80955_5 &
I_b_80955_4(N) <= I_z_80955_4 & I_b_80955_1A(N) <= I_z_80955_1A &
```

```
I_b_80955_0A(N) <= I_z_80955_0A & U_144500711(N) >= v_min & U_144500711(N)
<= v_max & P_load_tot(N) <= P_transformer & USE_144500711==1
    EX_144500711 = 1;        % Execute charging this hour
else
    EX_144500711 = 0;
end
Ma_144500711(N) = EX_144500711;
N=N+1;
end
% Show matrix with results and compare to initial charge request:
Ma_144503089; Ma_144503070; Ma_144502932; Ma_144500731; Ma_144500771;
Ma_114500751; Ma_144500615; Ma_144500741; Ma_144500619; Ma_144500621;
Ma_144590621; Ma_144500781; Ma_144500761; Ma_144500611; Ma_144500625;
Ma_144500721; Ma_144500617; Ma_144500622; Ma_144500791; Ma_144500711;
```

Appendix D – MATLAB script objective 3

```
% Smart Load Management for electric vehicles
% -----------------------------------------------------------------------
% SLM Algorithm Objective 3
% Master Thesis Spring 2015
% Author: Stian Namtvedt Gjelsvik
% Email: stian2803@gmail.com
% Version: 1.0
% Last update: 29.04.2015
% -----------------------------------------------------------------------
% Description:
% This script is a suggested smart load management algorithm, that
% will schedule charging of EVs in a specific order customized for a
% typical Norwegian low voltage distribution network. The algorithm takes
% care of the constraints in the network and will try to minimize line
% power losses if possible.
% For minute based calculations: Set N=1441 with equivalent load data
% and user profile input.
% ----------------------------------------------------------------------
clc;
clear;
% Input:
P_charger = 3.3;    % Charging power (depending on the charger)
SOC=0.6666;         % State of charge. 21 kWh max. 7kWh used / Trans_Eff
Max_bat=21;         % Maximum capacity [kWh] battery
SOC_Recom=0.8;      % Recommended state of charge
Charge_Eff=0.93;    % Charge efficiency
Trans_Eff=0.865;    % Transportation efficiency
S_transformer = 150;
CosPhi = 0.9;       % External value from DIgSILENT PowerFactory
Phi = 25.8419327;   % CosPhi transformer lossless = CosPhi network
P_transformer = S_transformer*CosPhi;
U_Transformer = 238;
v_min=207;
v_max=253;
Load_max=P_transformer ;  % Max load for minimazation of power loss

% Check for constraints? If 1 = check. if 0 = bypass
Constraints=1;
% Check for loss minimazation? If 1 = check. if 0 = bypass
Minimazation=1;

% Load profile for residence for the seasons, hourly based 00.00 to 00.00
Load_Summer = [0.97 0.97 0.97 0.97 0.97 0.97 1.62 3.24 3.24 0.81 0.86 0.86
0.86 0.86 0.86 0.86 3.88 3.53 2.16 2.16 2.16 1.94 1.62 1.62 0.97];
Load_Autumn = [1.58 1.58 1.58 1.58 1.58 1.58 2.97 4.45 4.45 1.58 1.58 1.58
1.58 1.58 1.58 2.97 5.93 5.93 5.93 5.08 4.45 3.95 2.97 2.97 1.58];
Load_Winter = [2.77 2.77 2.77 2.77 2.77 2.77 3.24 4.04 4.04 3.24 3.24 3.24
3.24 3.24 3.24 3.24 6.93 8.09 6.93 6.93 6.47 4.85 4.85 4.04 2.77];
Load_Spring = [1.29 1.29 1.29 1.29 1.29 1.29 1.29 2.16 2.16 1.48 1.48 1.48
1.48 1.48 1.48 1.48 4.71 4.31 3.70 3.24 2.88 2.59 2.35 2.16 1.29];

% User connection from 00.00 to 00.00, where 0 is disconnected and 1
% connected
User_Connection_1 = [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 ];
User_Connection_2 = [1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 ];
User_Connection_3 = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 ];
User_Connection_4 = [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 ];
```

```
%Load = Load_Winter;
%Load=Load_Summer;
%Load=Load_Spring;
Load=Load_Autumn;

% Eight residences are chosen to have EV connected.
% User_Connection chosen
% Initial load: (Will be changed in demand loop)
L_init_144503089=Load+User_Connection_1*P_charger;
Load_144503089=L_init_144503089;
L_144503089=L_init_144503089;
L_init_144503070=Load+User_Connection_1*P_charger;
L_144503070=L_init_144503070;
Load_144503070=L_init_144503070;
L_init_144502932=Load+User_Connection_1*P_charger;
L_144502932=L_init_144502932;
Load_144502932=L_init_144502932;
L_init_144500731=Load+User_Connection_1*P_charger;
L_144500731=L_init_144500731;
Load_144500731=L_init_144500731;
L_init_144500771=Load+User_Connection_2*P_charger;
L_144500771=L_init_144500771;
Load_144500771=L_init_144500771;
L_init_114500751=Load+User_Connection_2*P_charger;
L_114500751=L_init_114500751;
Load_114500751=L_init_114500751;
L_init_144500615=Load+User_Connection_3*P_charger;
L_144500615=L_init_144500615;
Load_144500615=L_init_144500615;
L_init_144500741=Load+User_Connection_4*P_charger;
L_144500741=L_init_144500741;
Load_144500741=L_init_144500741;
% Residence with NO EV:
L_init_144500619=Load;
L_144500619=L_init_144500619;
Load_144500619=L_init_144500619;
L_init_144500621=Load;
L_144500621=L_init_144500621;
Load_144500621=L_init_144500621;
L_init_144590621=Load;
L_144590621=L_init_144590621;
Load_144590621=L_init_144590621;
L_init_144500781=Load;
L_144500781=L_init_144500781;
Load_144500781=L_init_144500781;
L_init_144500761=Load;
L_144500761=L_init_144500761;
Load_144500761=L_init_144500761;
L_init_144500611=Load;
L_144500611=L_init_144500611;
Load_144500611=L_init_144500611;
L_init_144500625=Load;
L_144500625=L_init_144500625;
Load_144500625=L_init_144500625;
L_init_144500721=Load;
L_144500721=L_init_144500721;
Load_144500721=L_init_144500721;
L_init_144500617=Load;
L_144500617=L_init_144500617;
Load_144500617=L_init_144500617;
```

```
L_init_144500622=Load;
L_144500622=L_init_144500622;
Load_144500622=L_init_144500622;
L_init_144500791=Load;
L_144500791=L_init_144500791;
Load_144500791=L_init_144500791;
L_init_144500711=Load;
L_144500711=L_init_144500711;
Load_144500711=L_init_144500711;
% Load_Tot_init : EV user connected + possible full load at given hour
Load_Tot_init=L_init_144503089+L_init_144503070+L_init_144502932+L_init_144
500731+L_init_144500771+L_init_114500751+L_init_144500615+L_init_144500741+
L_init_144500619+L_init_144500621+L_init_144590621+L_init_144500781+L_init_
144500761+L_init_144500611+L_init_144500625+L_init_144500721+L_init_1445006
17+L_init_144500622+L_init_144500791+L_init_144500711;
Load_Tot=Load_Tot_init(1);
%Initial voltage condition: (for calculation)
U_init=230;
U_144503089=U_init; U_144503070=U_init; U_144502932=U_init;
U_144500731=U_init; U_144500771=U_init; U_114500751=U_init;
U_144500615=U_init; U_144500741=U_init; U_144500619=U_init;
U_144500621=U_init; U_144590621=U_init; U_144500781=U_init;
U_144500761=U_init; U_144500611=U_init; U_144500625=U_init;
U_144500721=U_init; U_144500617=U_init; U_144500622=U_init;
U_144500791=U_init; U_144500711=U_init;
% Constraints: -----------------------------------------
% v_min <= v_n <= v_max
% v_min=207;
% v_max=253;
% I_b <= I_z  (Load current <= cable conductivity)
% P_load_tot <= P_transformer
%-----------------------------------------------------
% Set Pri_XXXXXXXX = 1 ; IF consumer have priority
Pri_144503089=0; Pri_144503070=0;  Pri_144502932=0;
Pri_144500731=0; Pri_144500771=0;  Pri_114500751=0;
Pri_144500615=0; Pri_144500741=0;  Pri_144500619=0;
Pri_144500621=0; Pri_144590621=0;  Pri_144500781=0;
Pri_144500761=0; Pri_144500611=0;  Pri_144500625=0;
Pri_144500721=0; Pri_144500617=0;  Pri_144500622=0;
Pri_144500791=0; Pri_144500711=0;
% Line data:
% Lines feeding the residence: Z = R + jX
R_144503089= [0.0432 0.0101];
R_144503070= [0.0661 0.0103];
R_144502932= [0.0263 0.0032];
R_144500731= [0.0266 0.0033];
R_144500771= [0.0216 0.0024];
R_114500751= [0.0192 0.0013];
R_144500615= [0.0156 0.0011];
R_144500741= [0.0144 0.0095];
R_144500619= [0.0072 0.0005];
R_144500621= [0.0504 0.0034];
R_144590621= [0.0300 0.0021];
R_144500781= [0.5304 0.0036];
R_144500761= [0.0300 0.0021];
R_144500611= [0.0120 0.0008];
R_144500625= [0.0132 0.0009];
R_144500721= [0.0227 0.0017];
R_144500617= [0.0216 0.0015];
R_144500622= [0.1884 0.0129];
R_144500791= [0.3216 0.0220];
```

```
R_144500711= [0.1884 0.0129];
% Feeding lines: Z = R + jX
F_80955_0= [0.0011 0.0006];
F_80955_0A= [0.0010 0.0006];
F_80955_5= [0.0199 0.0024];
F_80955_16= [0.0314 0.0074];
F_80955_17= [0.0600 0.0041];
F_80955_19= [0.0244 0.0029];
F_80955_22= [0.0853 0.0102];
F_80955_23= [0.0444 0.0030];
F_80955_A1= [0.0036 0.0002];
F_80955_24= [0.0552 0.0038];
F_80955_26= [0.1032 0.0071];
F_80955_18= [0.2124 0.0145];
F_80955_1= [0.0096 0.0023];
F_80955_1A= [0.0096 0.0023];
F_80955_9= [0.0493 0.0117];
F_80955_10= [0.0122 0.0029];
F_80955_14= [0.1308 0.0157];
F_80955_29= [0.0224 0.0027];
F_80955_4= [0.0769 0.0092];
% Total - Total impedance to the residences. Z = R + jX
TOT_144503089= F_80955_0 + F_80955_1 + F_80955_16 + F_80955_19 + F_80955_22
+ F_80955_23 + R_144503089;
TOT_144503070= F_80955_0 + F_80955_1 + F_80955_16 + F_80955_19 + F_80955_22
+ F_80955_23 + R_144503070;
TOT_144502932= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144502932;
TOT_144500731= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144500731;
TOT_144500771= F_80955_1 + F_80955_0 + R_144500771;  % Very close to
transformer = low resistance/loss
TOT_114500751= F_80955_18 + F_80955_1 + F_80955_0 + R_114500751;
TOT_144500615= F_80955_17 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500615;
TOT_144500741= F_80955_17 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500741;
TOT_144500619= F_80955_24 + F_80955_23 + F_80955_22 + F_80955_19 +
F_80955_16 + F_80955_1 + F_80955_0 + R_144500619;
TOT_144500621= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144500621;
TOT_144590621= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144590621;
TOT_144500781= F_80955_26 + F_80955_22 + F_80955_19 + F_80955_16 +
F_80955_1 + F_80955_0 + R_144500781;
TOT_144500761= F_80955_19 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500761;
TOT_144500611= F_80955_19 + F_80955_16 + F_80955_1 + F_80955_0 +
R_144500611;
TOT_144500625= F_80955_14 + F_80955_10 + F_80955_9 + F_80955_1A +
F_80955_0A + R_144500625;
TOT_144500721= F_80955_9 + F_80955_1A + F_80955_0A + R_144500721;
TOT_144500617= F_80955_9 + F_80955_1A + F_80955_0A + R_144500617;
TOT_144500622= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500622;
TOT_144500791= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500791;
TOT_144500711= F_80955_5 + F_80955_4 + F_80955_1A + F_80955_0A +
R_144500771;
% Cable conductivity:
I_z_144503089=220;
```

```
I_z_144503070=150;
I_z_80955_0=435;
I_z_80955_0A=435;
I_z_144502932=150;
I_z_80955_5=150;
I_z_80955_16=280;
I_z_80955_17=115;
I_z_144500615=115;
I_z_80955_19=180;
I_z_80955_22=180;
I_z_80955_23=115;
I_z_80955_A1=115;
I_z_80955_24=115;
I_z_144500619=115;
I_z_80955_26=115;
I_z_144500621=115;
I_z_144590621=115;
I_z_144500611=115;
I_z_80955_18=115;
I_z_80955_1 =280;
I_z_80955_1A=280;
I_z_80955_9=280;
I_z_80955_10=280;
I_z_80955_14=180;
I_z_144500625=115;
I_z_80955_29=180;
I_z_144500617=115;
I_z_80955_4 =180;
I_z_144500622=115;
I_z_144500731=115;
I_z_144500771=115;
I_z_114500751=115;
I_z_144500741=115;
I_z_144500781=115;
I_z_144500761=115;
I_z_144500721=115;
I_z_144500791=115;
I_z_144500711=115;
% Initial feeder currents: (First hour with preffered charging of EVs)
I_b_144503089 = (1000 * L_144503089(1)) / (U_144503089 * CosPhi* sqrt(3));
I_b_144503070 = (1000 * L_144503070(1)) / (U_144503070 * CosPhi* sqrt(3));
I_b_144502932 = (1000 * L_144502932(1)) / (U_144502932 * CosPhi* sqrt(3));
I_b_144500731 = (1000 * L_144500731(1)) / (U_144500731 * CosPhi* sqrt(3));
I_b_144500771 = (1000 * L_144500771(1)) / (U_144500771 * CosPhi* sqrt(3));
I_b_114500751 = (1000 * L_114500751(1)) / (U_114500751 * CosPhi* sqrt(3));
I_b_144500615 = (1000 * L_144500615(1)) / (U_144500615 * CosPhi* sqrt(3));
I_b_144500741 = (1000 * L_144500741(1)) / (U_144500741 * CosPhi* sqrt(3));
% Without EV:(I_b_144500741 have initially NO EV. (less loss))
I_b_144500619 = (1000 * L_144500619(1)) / (U_144500619 * CosPhi* sqrt(3));
I_b_144500621 = (1000 * L_144500621(1)) / (U_144500621 * CosPhi* sqrt(3));
I_b_144590621 = (1000 * L_144590621(1)) / (U_144590621 * CosPhi* sqrt(3));
I_b_144500781 = (1000 * L_144500781(1)) / (U_144500781 * CosPhi* sqrt(3));
I_b_144500761 = (1000 * L_144500761(1)) / (U_144500761 * CosPhi* sqrt(3));
I_b_144500611 = (1000 * L_144500611(1)) / (U_144500611 * CosPhi* sqrt(3));
I_b_144500625 = (1000 * L_144500625(1)) / (U_144500625 * CosPhi* sqrt(3));
I_b_144500721 = (1000 * L_144500721(1)) / (U_144500721 * CosPhi* sqrt(3));
I_b_144500617 = (1000 * L_144500617(1)) / (U_144500617 * CosPhi* sqrt(3));
I_b_144500622 = (1000 * L_144500622(1)) / (U_144500622 * CosPhi* sqrt(3));
I_b_144500791 = (1000 * L_144500791(1)) / (U_144500791 * CosPhi* sqrt(3));
I_b_144500711 = (1000 * L_144500711(1)) / (U_144500711 * CosPhi* sqrt(3));
```

```
I_b_80955_0= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771 ;
I_b_80955_0A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_5= I_b_144500711 + I_b_144500622 + I_b_144500791;
I_b_80955_16= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741;
I_b_80955_17= I_b_144500615 + I_b_144500741 ;
I_b_80955_19= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761;
I_b_80955_22= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781;
I_b_80955_23= I_b_144503070 + I_b_144503089 + I_b_144500619;
I_b_80955_A1= I_b_144503070 + I_b_144503089;
I_b_80955_24= I_b_144500619;
I_b_80955_26= I_b_144590621 + I_b_144500621 + I_b_144500781;
I_b_80955_18= I_b_114500751;
I_b_80955_1= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771 ;
I_b_80955_1A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_9= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617;
I_b_80955_10= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_14= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_29= I_b_144502932 + I_b_144500731;
I_b_80955_4= I_b_144500711 + I_b_144500622 + I_b_144500791;
% Initial loss:
Loss_144503089=I_b_144503089^2*TOT_144503089(1)+I_b_144503089^2*TOT_1445030
89(2);
Loss_144503070=I_b_144503070^2*TOT_144503070(1)+I_b_144503070^2*TOT_1445030
70(2);
Loss_144502932=I_b_144502932^2*TOT_144502932(1)+I_b_144502932^2*TOT_1445029
32(2);
Loss_144500731=I_b_144500731^2*TOT_144500731(1)+I_b_144500731^2*TOT_1445007
31(2);
Loss_144500771=I_b_144500771^2*TOT_144500771(1)+I_b_144500771^2*TOT_1445007
71(2);
Loss_114500751=I_b_114500751^2*TOT_114500751(1)+I_b_114500751^2*TOT_1145007
51(2);
Loss_144500615=I_b_144500615^2*TOT_144500615(1)+I_b_144500615^2*TOT_1445006
15(2);
Loss_144500741=I_b_144500741^2*TOT_144500741(1)+I_b_144500741^2*TOT_1445007
41(2);
Loss_144500619=I_b_144500619^2*TOT_144500619(1)+I_b_144500619^2*TOT_1445006
19(2);
Loss_144500621=I_b_144500621^2*TOT_144500621(1)+I_b_144500621^2*TOT_1445006
21(2);
Loss_144590621=I_b_144590621^2*TOT_144590621(1)+I_b_144590621^2*TOT_1445906
21(2);
Loss_144500781=I_b_144500781^2*TOT_144500781(1)+I_b_144500781^2*TOT_1445007
81(2);
Loss_144500761=I_b_144500761^2*TOT_144500761(1)+I_b_144500761^2*TOT_1445007
61(2);
Loss_144500611=I_b_144500611^2*TOT_144500611(1)+I_b_144500611^2*TOT_1445006
11(2);
Loss_144500625=I_b_144500625^2*TOT_144500625(1)+I_b_144500625^2*TOT_1445006
25(2);
```

```
Loss_144500721=I_b_144500721^2*TOT_144500721(1)+I_b_144500721^2*TOT_1445007
21(2);
Loss_144500617=I_b_144500617^2*TOT_144500617(1)+I_b_144500617^2*TOT_1445006
17(2);
Loss_144500622=I_b_144500622^2*TOT_144500622(1)+I_b_144500622^2*TOT_1445006
22(2);
Loss_144500791=I_b_144500791^2*TOT_144500791(1)+I_b_144500791^2*TOT_1445007
91(2);
Loss_144500711=I_b_144500711^2*TOT_144500711(1)+I_b_144500711^2*TOT_1445007
11(2);
% Initial EV loss matrix (Total loss to the residence):
EV_R_Loss_Matrix_Init= [Loss_144503089 Loss_144503070 Loss_144502932
Loss_144500731 Loss_144500771 Loss_114500751 Loss_144500615 Loss_144500741
];
EV_loss=EV_R_Loss_Matrix_Init;
% Initial load:
L_144503089= L_144503089(1);
L_144503089= L_144503089(1);
L_144503070= L_144503070(1);
L_144502932= L_144502932(1);
L_144500731= L_144500731(1);
L_144500771= L_144500771(1);
L_114500751= L_114500751(1);
L_144500615= L_144500615(1);
L_144500741= L_144500741(1);
L_144500619= L_144500619(1);
L_144500621= L_144500621(1);
L_144590621= L_144590621(1);
L_144500781= L_144500781(1);
L_144500761= L_144500761(1);
L_144500611= L_144500611(1);
L_144500625= L_144500625(1);
L_144500721= L_144500721(1);
L_144500617= L_144500617(1);
L_144500622= L_144500622(1);
L_144500791= L_144500791(1);
L_144500711= L_144500711(1);
% Calculations : -----------------------------------------------------
% Calculate charging time for EVs:
EV_power=(1-SOC)*Max_bat/Trans_Eff;
% Initial values for residences with EV:
% ( Script starts at 00.00 )
Hour_charge_init=EV_power/(P_charger*Charge_Eff);
if SOC < 0.5
Hour_charge_144503089=Hour_charge_init;  % (Have been connected X hours)
Hour_charge_144503070=Hour_charge_init;
Hour_charge_144502932=Hour_charge_init;
Hour_charge_144500731=Hour_charge_init;
Hour_charge_144500771=Hour_charge_init-2; % (Have been connected X hours)
Hour_charge_114500751=Hour_charge_init-2; % (Have been connected X hours
Hour_charge_144500615=Hour_charge_init; % (First connection initally)
Hour_charge_144500741=Hour_charge_init; % (First connection initally)
end
if SOC > 0.5
Hour_charge_144503089=Hour_charge_init-2;  % (Have been connected X hours)
Hour_charge_144503070=Hour_charge_init-2;
Hour_charge_144502932=Hour_charge_init-2;
Hour_charge_144500731=Hour_charge_init-2;
Hour_charge_144500771=Hour_charge_init-3; % (Have been connected X hours)
Hour_charge_114500751=Hour_charge_init-3; % (Have been connected X hours
Hour_charge_144500615=Hour_charge_init; % (First connection initally)
```

```
Hour_charge_144500741=Hour_charge_init; % (First connection initally)
end
N=1;    % Initial hour
% Calculation while loop
while N<25
% 1. Residence with EV:
% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144503089 = EV_power/(P_charger*Charge_Eff);
else
  Hour_charge_144503089 = Hour_charge_144503089;
  end

    % Priority:
    if Pri_144503089==1 & Hour_charge_144503089 > 0
        L_144503089=Load_144503089(N)
    end
    % No Priority
        if Minimazation == 1
    if Pri_144503089==0 & Hour_charge_144503089 > 0
        L_144503089=Load_144503089(N);
    elseif Pri_144503089==0 & Hour_charge_144503089 > 0
        L_144503089=Load_144503089(N);
    else
        L_144503089=Load(N);
    end
        elseif Pri_144503089==0 & Hour_charge_144503089 > 0
         L_144503089=Load_144503089(N);
        else
         L_144503089=Load(N);
        end
% Hourly based calculations:
I_b_144503089 = (1000 * L_144503089) / (U_144503089 * CosPhi* sqrt(3));
% Delta voltage:
D_144503089 = sqrt(3)*(TOT_144503089(1)*I_b_144503089*cosd(Phi) +
TOT_144503089(2)*I_b_144503089*sind(Phi));
% Voltage at residence:
U_144503089 = U_Transformer - D_144503089;
% Loss caused by residence + EV
Loss_144503089=I_b_144503089^2*TOT_144503089(1)+I_b_144503089^2*TOT_1445030
89(2);
% 2. Residence with EV:
% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144503070 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144503070 = Hour_charge_144503070;
end
    % Priority:
    if Pri_144503070==1 & Hour_charge_144503070 > 0
        L_144503070=Load_144503070(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_144503070==0 & Hour_charge_144503070 > 0
        L_144503070=Load_144503070(N);
    elseif Pri_144503070==0 & Hour_charge_144503070 > 0
        L_144503070=Load_144503070(N);
    else
        L_144503070=Load(N);
    end
```

```
        elseif Pri_144503070==0 & Hour_charge_144503070 > 0
         L_144503070=Load_144503070(N);
        else
         L_144503070=Load(N);
        end
% Hourly based calculations:
I_b_144503070 = (1000 * L_144503070) / (U_144503070 * CosPhi* sqrt(3));
% Delta voltage:
D_144503070 = sqrt(3)*(TOT_144503070(1)*I_b_144503070*cosd(Phi) +
TOT_144503070(2)*I_b_144503070*sind(Phi));
% Voltage at residence:
U_144503070 = U_Transformer - D_144503089;
% Loss caused by residence + EV
Loss_144503070=I_b_144503070^2*TOT_144503070(1)+I_b_144503070^2*TOT_1445030
70(2);
% 3. Residence with EV:
% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144502932 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144502932 = Hour_charge_144502932;
end
    % Priority:
    if Pri_144502932==1 & Hour_charge_144502932 > 0
        L_144502932=Load_144502932(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_144502932==0 & Hour_charge_144502932 > 0
        L_144502932=Load_144502932(N);
    elseif Pri_144502932==0 & Hour_charge_144502932 > 0
        L_144502932=Load_144502932(N);
    else
        L_144502932=Load(N);
    end
        elseif Pri_144502932==0 & Hour_charge_144502932 > 0
         L_144502932=Load_144502932(N);
        else
         L_144502932=Load(N);
        end
% Hourly based calculations:
I_b_144502932 = (1000 * L_144502932) / (U_144502932 * CosPhi* sqrt(3));
% Delta voltage:
D_144502932 = sqrt(3)*(TOT_144502932(1)*I_b_144502932*cosd(Phi) +
TOT_144502932(2)*I_b_144502932*sind(Phi));
% Voltage at residence:
U_144502932 = U_Transformer - D_144502932;
% Loss caused by residence + EV
Loss_144502932=I_b_144502932^2*TOT_144502932(1)+I_b_144502932^2*TOT_1445029
32(2);
% 4. Residence with EV:
% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144500731 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144500731 = Hour_charge_144500731;
end
    % Priority:
    if Pri_144500731==1 & Hour_charge_144500731 > 0
        L_144503089=Load_144503089(N);
    end
```

```
    % No Priority
        if Minimazation == 1
    if Pri_144500731==0 & Hour_charge_144500731 > 0
        L_144500731=Load_144500731(N);
    elseif Pri_144500731==0 & Hour_charge_144500731 > 0
        L_144500731=Load_144500731(N);
    else
        L_144500731=Load(N);
    end
        elseif Pri_144500731==0 & Hour_charge_144500731 > 0
         L_144500731=Load_144500731(N);
        else
         L_144500731=Load(N);
        end
% Hourly based calculations:
I_b_144500731 = (1000 * L_144500731) / (U_144500731 * CosPhi* sqrt(3));
% Delta voltage:
D_144500731 = sqrt(3)*(TOT_144500731(1)*I_b_144500731*cosd(Phi) +
TOT_144500731(2)*I_b_144500731*sind(Phi));
% Voltage at residence:
U_144500731 = U_Transformer - D_144500731;
% Loss caused by residence + EV
Loss_144500731=I_b_144500731^2*TOT_144500731(1)+I_b_144500731^2*TOT_1445007
31(2);

% 5. Residence with EV:
% SOC - Connection logic
if User_Connection_2(N+1) == 1 & User_Connection_2(N) == 0
    Hour_charge_144500771 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144500771 = Hour_charge_144500771;
end
    % Priority:
    if Pri_144500771==1 & Hour_charge_144500771 > 0
        L_144500771=Load_144500771(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_144500771==0 & Hour_charge_144500771 > 0
        L_144500771=Load_144500771(N);
    elseif Pri_144500771==0 & Hour_charge_144500771 > 0
        L_144500771=Load_144500771(N);
    else
        L_144500771=Load(N);
    end
        elseif Pri_144500771==0 & Hour_charge_144500771 > 0
         L_144500771=Load_144500771(N);
        else
         L_144500771=Load(N);
        end
% Hourly based calculations:
I_b_144500771 = (1000 * L_144500771) / (U_144500771 * CosPhi* sqrt(3));
% Delta voltage:
D_144500771 = sqrt(3)*(TOT_144500771(1)*I_b_144500771*cosd(Phi) +
TOT_144500771(2)*I_b_144500771*sind(Phi));
% Voltage at residence:
U_144500771 = U_Transformer - D_144500771;
% Loss caused by residence + EV
Loss_144500771=I_b_144500771^2*TOT_144500771(1)+I_b_144500771^2*TOT_1445007
71(2);
% 6. Residence with EV:
```

```
% SOC - Connection logic
if User_Connection_2(N+1) == 1 & User_Connection_2(N) == 0
    Hour_charge_114500751 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_114500751 = Hour_charge_114500751;
end
    % Priority:
    if Pri_114500751==1 & Hour_charge_114500751 > 0
        L_114500751=Load_114500751(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_114500751==0 & Hour_charge_114500751 > 0
        L_114500751=Load_114500751(N);
    elseif Pri_114500751==0 & Hour_charge_114500751 > 0
        L_114500751=Load_114500751(N);
    else
        L_114500751=Load(N);
    end
        elseif Pri_114500751==0 & Hour_charge_114500751 > 0
         L_114500751=Load_114500751(N);
        else
         L_114500751=Load(N);
        end
% Hourly based calculations:
I_b_114500751 = (1000 * L_114500751) / (U_114500751 * CosPhi* sqrt(3));
% Delta voltage:
D_114500751 = sqrt(3)*(TOT_114500751(1)*I_b_114500751*cosd(Phi) +
TOT_114500751(2)*I_b_114500751*sind(Phi));
% Voltage at residence:
U_114500751 = U_Transformer - D_114500751;
% Loss caused by residence + EV
Loss_114500751=I_b_114500751^2*TOT_114500751(1)+I_b_114500751^2*TOT_1145007
51(2);
% 7. Residence with EV:
% SOC - Connection logic
if User_Connection_3(N+1) == 1 & User_Connection_3(N) == 0
    Hour_charge_144500615 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144500615 = Hour_charge_144500615;
end
    % Priority:
    if Pri_144500615==1 & Hour_charge_144500615 > 0
        L_144500615=Load_144500615(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_144500615==0 & Hour_charge_144500615 > 0
        L_144500615=Load_144500615(N);
    elseif Pri_144500615==0 & Hour_charge_144500615 > 0
        L_144500615=Load_144500615(N);
    else
        L_144500615=Load(N);
    end
        elseif Pri_144500615==0 & Hour_charge_144500615 > 0
         L_144500615=Load_144500615(N);
        else
         L_144500615=Load(N);
        end
% Hourly based calculations:
I_b_144500615 = (1000 * L_144500615) / (U_144500615 * CosPhi* sqrt(3));
```

```
% Delta voltage:
D_144500615 = sqrt(3)*(TOT_144500615(1)*I_b_144500615*cosd(Phi) +
TOT_144500615(2)*I_b_144500615*sind(Phi));
% Voltage at residence:
U_144500615 = U_Transformer - D_144500615;
% Loss caused by residence + EV
Loss_144500615=I_b_144500615^2*TOT_144500615(1)+I_b_144500615^2*TOT_1445006
15(2);
% 8. Residence with EV:
% SOC - Connection logic
if User_Connection_4(N+1) == 1 & User_Connection_4(N) == 0
    Hour_charge_144500741 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144500741 = Hour_charge_144500741;
end
    % Priority:
    if Pri_144500741==1 & Hour_charge_144500741 > 0
        L_144500741=Load_144500741(N);
    end
    % No Priority
        if Minimazation == 1
    if Pri_144500741==0 & Hour_charge_144500741 > 0
        L_144500741=Load_144500741(N);
    elseif Pri_144500741==0 & Hour_charge_144500741 > 0
        L_144500741=Load_144500741(N);
    else
        L_144500741=Load(N);
    end
        elseif Pri_144500741==0 & Hour_charge_144500741 > 0
         L_144500741=Load_144500741(N);
        else
         L_144500741=Load(N);
        end
% Hourly based calculations:
I_b_144500741 = (1000 * L_144500741) / (U_144500741 * CosPhi* sqrt(3));
% Delta voltage:
D_144500741 = sqrt(3)*(TOT_144500741(1)*I_b_144500741*cosd(Phi) +
TOT_144500741(2)*I_b_144500741*sind(Phi));
% Voltage at residence:
U_144500741 = U_Transformer - D_144500741;
% Loss caused by residence + EV
Loss_144500741=I_b_144500741^2*TOT_144500741(1)+I_b_144500741^2*TOT_1445007
41(2);
% Residences without EV:
%Current:
I_b_144500619 = (1000 * L_144500619) / (U_144500619 * CosPhi);
% Delta voltage:
D_144500619 = sqrt(3)*(TOT_144500619(1)*I_b_144500619*cosd(Phi) +
TOT_144500619(2)*I_b_144500619*sind(Phi));
% Voltage at residence:
U_144500619 = U_Transformer - D_144500619;
Loss_144500619=I_b_144500619^2*TOT_144500619(1)+I_b_144500619^2*TOT_1445006
19(2);
L_144500619=Load(N);
%Current:
I_b_144500621 = (1000 * L_144500621) / (U_144500621 * CosPhi* sqrt(3));
% Delta voltage:
D_144500621 = sqrt(3)*(TOT_144500621(1)*I_b_144500621*cosd(Phi) +
TOT_144500621(2)*I_b_144500621*sind(Phi));
% Voltage at residence:
U_144500621 = U_Transformer - D_144500621;
```

```
Loss_144500621=I_b_144500621^2*TOT_144500621(1)+I_b_144500621^2*TOT_1445006
21(2);
L_144500621=Load(N);
%Current:
I_b_144590621 = (1000 * L_144590621) / (U_144590621 * CosPhi* sqrt(3));
% Delta voltage:
D_144590621 = sqrt(3)*(TOT_144590621(1)*I_b_144590621*cosd(Phi) +
TOT_144590621(2)*I_b_144590621*sind(Phi));
% Voltage at residence:
U_144590621 = U_Transformer - D_144590621;
Loss_144590621=I_b_144590621^2*TOT_144590621(1)+I_b_144590621^2*TOT_1445906
21(2);
L_144590621=Load(N);
%Current:
I_b_144500781 = (1000 * L_144500781) / (U_144500781 * CosPhi* sqrt(3));
% Delta voltage:
D_144500781 = sqrt(3)*(TOT_144500781(1)*I_b_144500781*cosd(Phi) +
TOT_144500781(2)*I_b_144500781*sind(Phi));
% Voltage at residence:
U_144500781 = U_Transformer - D_144500781;
Loss_144500781=I_b_144500781^2*TOT_144500781(1)+I_b_144500781^2*TOT_1445007
81(2);
L_144500781=Load(N);
%Current:
I_b_144500761 = (1000 * L_144500761) / (U_144500761 * CosPhi* sqrt(3));
% Delta voltage:
D_144500761 = sqrt(3)*(TOT_144500761(1)*I_b_144500761*cosd(Phi) +
TOT_144500761(2)*I_b_144500761*sind(Phi));
% Voltage at residence:
U_144500761 = U_Transformer - D_144500761;
Loss_144500761=I_b_144500761^2*TOT_144500761(1)+I_b_144500761^2*TOT_1445007
61(2);
L_144500761=Load(N);
%Current:
I_b_144500611 = (1000 * L_144500611) / (U_144500611 * CosPhi* sqrt(3));
% Delta voltage:
D_144500611 = sqrt(3)*(TOT_144500611(1)*I_b_144500611*cosd(Phi) +
TOT_144500611(2)*I_b_144500611*sind(Phi));
% Voltage at residence:
U_144500611 = U_Transformer - D_144500611;
Loss_144500611=I_b_144500611^2*TOT_144500611(1)+I_b_144500611^2*TOT_1445006
11(2);
L_144500611=Load(N);
%Current:
I_b_144500625 = (1000 * L_144500625) / (U_144500625 * CosPhi* sqrt(3));
% Delta voltage:
D_144500625 = sqrt(3)*(TOT_144500625(1)*I_b_144500625*cosd(Phi) +
TOT_144500625(2)*I_b_144500625*sind(Phi));
% Voltage at residence:
U_144500625 = U_Transformer - D_144500625;
Loss_144500625=I_b_144500625^2*TOT_144500625(1)+I_b_144500625^2*TOT_1445006
25(2);
L_144500625=Load(N);
%Current:
I_b_144500721 = (1000 * L_144500721) / (U_144500721 * CosPhi* sqrt(3));
% Delta voltage:
D_144500721 = sqrt(3)*(TOT_144500721(1)*I_b_144500721*cosd(Phi) +
TOT_144500721(2)*I_b_144500721*sind(Phi));
% Voltage at residence:
U_144500721 = U_Transformer - D_144500721;
```

```
Loss_144500721=I_b_144500721^2*TOT_144500721(1)+I_b_144500721^2*TOT_1445007
21(2);
L_144500721=Load(N);
%Current:
I_b_144500617 = (1000 * L_144500617) / (U_144500617 * CosPhi* sqrt(3));
% Delta voltage:
D_144500617 = sqrt(3)*(TOT_144500617(1)*I_b_144500617*cosd(Phi) +
TOT_144500617(2)*I_b_144500617*sind(Phi));
% Voltage at residence:
U_144500617 = U_Transformer - D_144500617;
Loss_144500617=I_b_144500617^2*TOT_144500617(1)+I_b_144500617^2*TOT_1445006
17(2);
L_144500617=Load(N);
%Current:
I_b_144500622 = (1000 * L_144500622) / (U_144500622 * CosPhi* sqrt(3));
% Delta voltage:
D_144500622 = sqrt(3)*(TOT_144500622(1)*I_b_144500622*cosd(Phi) +
TOT_144500622(2)*I_b_144500622*sind(Phi));
% Voltage at residence:
U_144500622 = U_Transformer - D_144500622;
Loss_144500622=I_b_144500622^2*TOT_144500622(1)+I_b_144500622^2*TOT_1445006
22(2);
L_144500622=Load(N);
%Current:
I_b_144500791 = (1000 * L_144500791) / (U_144500791 * CosPhi* sqrt(3));
% Delta voltage:
D_144500791 = sqrt(3)*(TOT_144500791(1)*I_b_144500791*cosd(Phi) +
TOT_144500791(2)*I_b_144500791*sind(Phi));
% Voltage at residence:
U_144500791 = U_Transformer - D_144500791;
Loss_144500791=I_b_144500791^2*TOT_144500791(1)+I_b_144500791^2*TOT_1445007
91(2);
L_144500791=Load(N);
%Current:
I_b_144500711 = (1000 * L_144500711) / (U_144500711 * CosPhi* sqrt(3));
% Delta voltage:
D_144500711 = sqrt(3)*(TOT_144500711(1)*I_b_144500711*cosd(Phi) +
TOT_144500711(2)*I_b_144500711*sind(Phi));
% Voltage at residence:
U_144500711 = U_Transformer - D_144500711;
Loss_144500711=I_b_144500711^2*TOT_144500711(1)+I_b_144500711^2*TOT_1445007
11(2);
L_144500711=Load(N);
% Total power and loss of the system:
Load_Tot =
L_144503089+L_144503070+L_144502932+L_144500731+L_144500771+L_114500751+L_1
44500615+L_144500741+L_144500619+L_144500621+L_144590621+L_144500781+L_1445
00761+L_144500611+L_144500625+L_144500721+L_144500617+L_144500622+L_1445007
91+L_144500711;
EV_loss = [Loss_144503089 Loss_144503070 Loss_144502932 Loss_144500731
Loss_144500771 Loss_114500751 Loss_144500615 Loss_144500741];
Sort_EV=sort(EV_loss);
% Feeder line currents:
I_b_80955_0= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771;
I_b_80955_0A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_5= I_b_144500711 + I_b_144500622 + I_b_144500791;
```

```
I_b_80955_16= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741;
I_b_80955_17= I_b_144500615 + I_b_144500741 ;
I_b_80955_19= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761;
I_b_80955_22= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781;
I_b_80955_23= I_b_144503070 + I_b_144503089 + I_b_144500619;
I_b_80955_A1= I_b_144503070 + I_b_144503089;
I_b_80955_24= I_b_144500619;
I_b_80955_26= I_b_144590621 + I_b_144500621 + I_b_144500781;
I_b_80955_18= I_b_114500751;
I_b_80955_1= I_b_144503070 + I_b_144503089 + I_b_144500619 + I_b_144590621
+ I_b_144500621 + I_b_144500781 + I_b_144500611 + I_b_144500761 +
I_b_144500615 + I_b_144500741 + I_b_114500751 + I_b_144500771 ;
I_b_80955_1A= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617 + I_b_144500622 + I_b_144500711 + I_b_144500791;
I_b_80955_9= I_b_144502932 + I_b_144500731 + I_b_144500625 + I_b_144500721
+ I_b_144500617;
I_b_80955_10= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_14= I_b_144502932 + I_b_144500731 + I_b_144500625;
I_b_80955_29= I_b_144502932 + I_b_144500731;
I_b_80955_4= I_b_144500711 + I_b_144500622 + I_b_144500791;
%EVs in the network
EV_in= User_Connection_1(N)*4 + User_Connection_2(N)*2 + User_Connection_3
(N) + User_Connection_4(N);
% Constraints concerning residences with EV:
% 1. Residence with EV:
if Constraints==1  % Checking for bypass constraint
    % Note: Load_Tot-P_charger (To give the first EV a chance to connect)
    if I_b_144503089 <= I_z_144503089 & I_b_80955_A1 <= I_z_80955_A1 &
I_b_80955_23 <= I_z_80955_23 & I_b_80955_22 <= I_z_80955_22 & I_b_80955_19
<= I_z_80955_19 & I_b_80955_16 <= I_z_80955_16 & I_b_80955_1 <= I_z_80955_1
& I_b_80955_0 <= I_z_80955_0 & U_144503089 >= v_min & U_144503089 <= v_max
& Load_Tot <= P_transformer
C_144503089=1;  % Constraints OK
else
C_144503089=0;   % One or more constraints NOT ok
    end
else
    C_144503089=1;
end
if C_144503089==0 & User_Connection_1 == 1
end
    % Priority:
    if Pri_144503089==1 & C_144503089==1 & Hour_charge_144503089 > 0 &
User_Connection_1(N)==1
    L_144503089=Load_144503089(N);
    Hour_charge=Hour_charge-1;
     elseif Pri_144503089==1 & C_144503089==0 & Hour_charge_144503089 > 0
    L_144503070=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144503089==0 & Hour_charge_144503089 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144503089 |
Sort_EV(2)== Loss_144503089  | Sort_EV(3) == Loss_144503089 ) &
User_Connection_1(N)==1
    L_144503089=Load_144503089(N);
    Hour_charge_144503089=Hour_charge_144503089-1;
```

```
     C_144503089=1 ;
%(EV_in*P_charger) =?? spring scenario
    elseif Pri_144503089==0 & Hour_charge_144503089 > 0 & C_144503089==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_1(N)==1;
    L_144503089=Load_144503089(N);
    Hour_charge_144503089=Hour_charge_144503089-1;
    else
    L_144503089=Load(N);
    Hour_charge_144503089=Hour_charge_144503089;
    end
        elseif Pri_144503089==0 & Hour_charge_144503089 > 0 &
C_144503089==1 & User_Connection_1(N)==1
            L_144503089=Load_144503089(N);
            Hour_charge_144503089=Hour_charge_144503089-1;
        else
         L_144503089=Load(N);
         Hour_charge_144503089=Hour_charge_144503089;
        end
% 2. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_144503070 <= I_z_144503070 & I_b_80955_A1 <= I_z_80955_A1 &
I_b_80955_23 <= I_z_80955_23 & I_b_80955_22 <= I_z_80955_22 & I_b_80955_19
<= I_z_80955_19 & I_b_80955_16 <= I_z_80955_16 & I_b_80955_1 <= I_z_80955_1
& I_b_80955_0 <= I_z_80955_0 & U_144503070 >= v_min & U_144503070 <= v_max
& Load_Tot <= P_transformer
C_144503070=1;  % Constraints OK
else
C_144503070=0;   % One or more constraints NOT ok
    end
else
    C_144503070=1;
end
if C_144503070==0 & User_Connection_1 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144503070==1 & C_144503070==1 & Hour_charge_144503070 > 0 &
User_Connection_1(N)==1
    L_144503070=Load_144503070(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144503070==1 & C_144503070==0 & Hour_charge_144503070 > 0
    L_144503070=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144503070==0 & Hour_charge_144503070 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144503070  |
Sort_EV(2)== Loss_144503070  | Sort_EV(3) == Loss_144503070 ) &
User_Connection_1(N)==1
    L_144503070=Load_144503070(N);
    Hour_charge_144503070=Hour_charge_144503070-1;
     C_144503070=1;
    elseif Pri_144503070==0 & Hour_charge_144503070 > 0 & C_144503070==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_1(N)==1;
    L_144503070=Load_144503070(N);
    Hour_charge_144503070=Hour_charge_144503070-1;
    else
    L_144503070=Load(N);
    Hour_charge_144503070=Hour_charge_144503070;
    end
```

```
        elseif Pri_144503070==0 & Hour_charge_144503070 > 0 &
C_144503070==1 & User_Connection_1(N)==1
            L_144503070=Load_144503070(N);
            Hour_charge_144503070=Hour_charge_144503070-1;
        else
         L_144503070=Load(N);
            Hour_charge_144503070=Hour_charge_144503070;
        end
% 3. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_144502932 <= I_z_144502932 & I_b_80955_29 <= I_z_80955_29 &
I_b_80955_14 <= I_z_80955_14 & I_b_80955_10 <= I_z_80955_10 & I_b_80955_9
<= I_z_80955_9 & I_b_80955_1A <= I_z_80955_1A & I_b_80955_0A <=
I_z_80955_0A & U_144502932 >= v_min & U_144502932 <= v_max & Load_Tot <=
P_transformer
C_144502932=1;  % Constraints OK
else
C_144502932=0;   % One or more constraints NOT ok
    end
else
    C_144502932=1;
end
if C_144502932==0 & User_Connection_1 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144502932==1 & C_144502932==1 & Hour_charge_144502932 > 0 &
User_Connection_1(N)==1
    L_144502932=Load_144502932(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144502932==1 & C_144502932==0 & Hour_charge_144502932 > 0
    L_144502932=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144502932==0 & Hour_charge_144502932 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144502932  |
Sort_EV(2)== Loss_144502932  | Sort_EV(3) == Loss_144502932 ) &
User_Connection_1(N)==1
    L_144502932=Load_144502932(N);
    Hour_charge_144502932=Hour_charge_144502932-1;
    C_144502932==1;
    elseif Pri_144502932==0 & Hour_charge_144502932 > 0 & C_144502932==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_1(N)==1;
    L_144502932=Load_144502932(N);
    Hour_charge_144502932=Hour_charge_144502932-1;
    else
    L_144502932=Load(N);
    Hour_charge_144502932=Hour_charge_144502932;
    end
        elseif Pri_144502932==0 & Hour_charge_144502932 > 0 &
C_144502932==1 & User_Connection_1(N)==1
            L_144502932=Load_144502932(N);
            Hour_charge_144502932=Hour_charge_144502932-1;
        else
         L_144502932=Load(N);
            Hour_charge_144502932=Hour_charge_144502932;
        end
% 4. Residence with EV:
if Constraints==1  % Checking for bypass constraint
```

```
if I_b_144500731 <= I_z_144500731 & I_b_80955_29 <= I_z_80955_29 &
I_b_80955_14 <= I_z_80955_14 & I_b_80955_10 <= I_z_80955_10 & I_b_80955_9
<= I_z_80955_9 & I_b_80955_1A <= I_z_80955_1A & I_b_80955_0A <=
I_z_80955_0A & U_144500731 >= v_min & U_144500731 <= v_max & Load_Tot <=
P_transformer
C_144500731=1;  % Constraints OK
else
C_144500731=0;   % One or more constraints NOT ok
    end
else
    C_144500731=1;
end
if C_144500731==0 & User_Connection_1 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144500731==1 & C_144500731==1 & Hour_charge_144500731 > 0 &
User_Connection_1(N)==1
    L_144500731=Load_144500731(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144500731==1 & C_144500731==0 & Hour_charge_144500731 > 0
    L_144500731=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144500731==0 & Hour_charge_144500731 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144500731  |
Sort_EV(2)== Loss_144500731  | Sort_EV(3) == Loss_144500731 )  &
User_Connection_1(N)==1
    L_144500731=Load_144500731(N);
    Hour_charge_144500731=Hour_charge_144500731-1;
    C_144500731=1;
    elseif Pri_144500731==0 & Hour_charge_144500731 > 0 & C_144500731==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_1(N)==1;
    L_144500731=Load_144500731(N);
    Hour_charge_144500731=Hour_charge_144500731-1;
    else
    L_144500731=Load(N);
    Hour_charge_144500731=Hour_charge_144500731;
    end
        elseif Pri_144500731==0 & Hour_charge_144500731 > 0 &
C_144500731==1 & User_Connection_1(N)==1
            L_144500731=Load_144500731(N);
            Hour_charge_144500731=Hour_charge_144500731-1;
        else
         L_144500731=Load(N);
            Hour_charge_144500731=Hour_charge_144500731;
        end
% 5. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_144500771 <= I_z_144500771 & I_b_80955_18 <= I_z_80955_18 &
I_b_80955_1 <= I_z_80955_1 & I_b_80955_0 <= I_z_80955_0 & U_144500771 >=
v_min & U_144500771 <= v_max & Load_Tot <= P_transformer
C_144500771=1;  % Constraints OK
else
C_144500771=0;   % One or more constraints NOT ok
    end
else
    C_144500771=1;
end
if C_144500771==0 & User_Connection_2 == 1
```

```
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144500771==1 & C_144500771==1 & Hour_charge_144500771 > 0 &
User_Connection_2(N)==1
    L_144500771=Load_144500771(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144500771==1 & C_144500771==0 & Hour_charge_144500771 > 0
    L_144500771=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144500771==0 & Hour_charge_144500771 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144500771  |
Sort_EV(2)== Loss_144500771  | Sort_EV(3) == Loss_144500771) &
User_Connection_2(N)==1
    L_144500771=Load_144500771(N);
    Hour_charge_144500771=Hour_charge_144500771-1;
    C_144500771=1 ;
    elseif Pri_144500771==0 & Hour_charge_144500771 > 0 & C_144500771==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_2(N)==1;
    L_144500771=Load_144500771(N);
    Hour_charge_144500771=Hour_charge_144500771-1;
    else
    L_144500771=Load(N);
    Hour_charge_144500771=Hour_charge_144500771;
    end
        elseif Pri_144500771==0 & Hour_charge_144500771 > 0 &
C_144500771==1 & User_Connection_2(N)==1
            L_144500771=Load_144500771(N);
            Hour_charge_144500771=Hour_charge_144500771-1;
        else
         L_144500771=Load(N);
            Hour_charge_144500771=Hour_charge_144500771;
        end
% 6. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_114500751 <= I_z_114500751 & I_b_80955_18 <= I_z_80955_18 &
I_b_80955_1 <= I_z_80955_1 & I_b_80955_0 <= I_z_80955_0 & U_114500751 >=
v_min & U_114500751 <= v_max & Load_Tot <= P_transformer
C_114500751=1;  % Constraints OK
else
C_114500751=0;   % One or more constraints NOT ok
    end
else
    C_114500751=1;
end
if C_114500751==0 & User_Connection_2 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_114500751==1 & C_114500751==1 & Hour_charge_114500751 > 0 &
User_Connection_2(N)==1
    L_114500751=Load_114500751(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_114500751==1 & C_114500751==0 & Hour_charge_114500751 > 0
    L_114500751=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
```

```
    if Pri_114500751==0 & Hour_charge_114500751 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max &( Sort_EV(1) == Loss_114500751  |
Sort_EV(2)== Loss_114500751 | Sort_EV(3) == Loss_114500751) &
User_Connection_2(N)==1
    L_114500751=Load_114500751(N);
    Hour_charge_114500751=Hour_charge_114500751-1;
    C_114500751=1;
    elseif Pri_114500751==0 & Hour_charge_114500751 > 0 & C_114500751==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_2(N)==1  ;
    L_114500751=Load_114500751(N);
    Hour_charge_114500751=Hour_charge_114500751-1;
    else
    L_114500751=Load(N);
    Hour_charge_114500751=Hour_charge_114500751;
    end
        elseif Pri_114500751==0 & Hour_charge_114500751 > 0 &
C_114500751==1 & User_Connection_2(N)==1
            L_114500751=Load_114500751(N);
            Hour_charge_114500751=Hour_charge_114500751-1;
        else
         L_114500751=Load(N);
            Hour_charge_114500751=Hour_charge_114500751;
        end
% 7. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_144500615 <= I_z_144500615 & I_b_80955_17 <= I_z_80955_17 &
I_b_80955_16 <= I_z_80955_16 & I_b_80955_1 <= I_z_80955_1 & I_b_80955_0 <=
I_z_80955_0 & U_144500615 >= v_min & U_144500615 <= v_max & Load_Tot <=
P_transformer
C_144500615=1;  % Constraints OK
else
C_144500615=0;   % One or more constraints NOT ok
    end
else
    C_144500615=1;
end
if C_144500615==0 & User_Connection_3 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144500615==1 & C_144500615==1 & Hour_charge_144500615 > 0 &
User_Connection_3(N)==1
    L_144500615=Load_144500615(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144500615==1 & C_144500615==0 & Hour_charge_144500615 > 0
    L_144500615=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144500615==0 & Hour_charge_144500615 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.08*Load_max & ( Sort_EV(1) == Loss_144500615  |
Sort_EV(2)== Loss_144500615| Sort_EV(3) == Loss_144500615) &
User_Connection_3(N)==1
    L_144500615=Load_144500615(N);
    Hour_charge_144500615=Hour_charge_144500615-1;
    C_144500615=1 ;
    elseif Pri_144500615==0 & Hour_charge_144500615 > 0 & C_144500615==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_3(N)==1 ;
    L_144500615=Load_144500615(N);
    Hour_charge_144500615=Hour_charge_144500615-1;
    else
```

```
    L_144500615=Load(N);
    Hour_charge_144500615=Hour_charge_144500615;
    end
        elseif Pri_144500615==0 & Hour_charge_144500615 > 0 &
C_144500615==1 & User_Connection_3(N)==1
            L_144500615=Load_144500615(N);
            Hour_charge_144500615=Hour_charge_144500615-1;
        else
         L_144500615=Load(N);
            Hour_charge_144500615=Hour_charge_144500615;
        end
% 8. Residence with EV:
if Constraints==1  % Checking for bypass constraint
if I_b_144500741 <= I_z_144500741 & I_b_80955_17 <= I_z_80955_17 &
I_b_80955_16 <= I_z_80955_16 & I_b_80955_1 <= I_z_80955_1 & I_b_80955_0 <=
I_z_80955_0 &  U_144500741 >= v_min & U_144500741 <= v_max & Load_Tot <=
P_transformer
C_144500741=1;  % Constraints OK
else
C_144500741=0;   % One or more constraints NOT ok
    end
else
    C_144500741=1;
end
if C_144500741==0 & User_Connection_4 == 1
    Load_Tot=Load_Tot-P_charger;
end
    % Priority:
    if Pri_144500741==1 & C_144500741==1 & Hour_charge_144500741 > 0 &
User_Connection_4(N)==1
    L_144500741=Load_144500741(N);
    Hour_charge=Hour_charge-1;
    elseif Pri_144500741==1 & C_144500741==0 & Hour_charge_144500741 > 0
    L_144500741=Load(N);
    end
    % Power loss minimazation:
        if Minimazation == 1
    if Pri_144500741==0 & Hour_charge_144500741 > 0 & Load_Tot >=
0.8*Load_max & Load_Tot < 1.05*Load_max & ( Sort_EV(1) == Loss_144500741  |
Sort_EV(2)== Loss_144500741  | Sort_EV(3) == Loss_144500741 ) &
User_Connection_4(N)==1
    L_144500741=Load_144500741(N);
    Hour_charge_144500741=Hour_charge_144500741-1;
     C_144500741=1;
    elseif Pri_144500741==0 & Hour_charge_144500741 > 0 & C_144500741==1 &
Load_Tot < 0.8*Load_max+(EV_in*P_charger) & User_Connection_4(N)==1 ;
    L_144500741=Load_144500741(N);
    Hour_charge_144500741=Hour_charge_144500741-1;
    else
    L_144500741=Load(N);
    Hour_charge_144500741=Hour_charge_144500741;
    end
        elseif Pri_144500741==0 & Hour_charge_144500741 > 0 &
C_144500741==1 & User_Connection_4(N)==1
            L_144500741=Load_144500741(N);
            Hour_charge_144500741=Hour_charge_144500741-1;
        else
         L_144500741=Load(N);
            Hour_charge_144500741=Hour_charge_144500741;
        end
% New total power:
```

```
Load_Tot_Final =
L_144503089+L_144503070+L_144502932+L_144500731+L_144500771+L_114500751+L_1
44500615+L_144500741+L_144500619+L_144500621+L_144590621+L_144500781+L_1445
00761+L_144500611+L_144500625+L_144500721+L_144500617+L_144500622+L_1445007
91+L_144500711;
% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144503089 = EV_power/(P_charger*Charge_Eff);
else
    Hour_charge_144503089 = Hour_charge_144503089;
end
% For presentation:
C_144503089_Plot(N)=C_144503089;
C_144503070_Plot(N)=C_144503070;
C_144502932_Plot(N)=C_144502932;
C_144500731_Plot(N)=C_144500731;
C_144500771_Plot(N)=C_144500771;
C_114500751_Plot(N)=C_114500751;
C_144500615_Plot(N)=C_144500615;
C_144500741_Plot(N)=C_144500741;

L_144503089_Plot(N)=L_144503089;
L_144503070_Plot(N)=L_144503070;
L_144502932_Plot(N)=L_144502932;
L_144500731_Plot(N)=L_144500731;
L_144500771_Plot(N)=L_144500771;
L_114500751_Plot(N)=L_114500751;
L_144500615_Plot(N)=L_144500615;
L_144500741_Plot(N)=L_144500741;

Load_Plot(N)=Load(N)*20;
Load_Tot_Plot(N)=Load_Tot;
Load_Tot_Final_Plot(N)=Load_Tot_Final;
Time_Plot(N)=N;

N=N+1;
end
plot(Time_Plot, L_144500741_Plot)
grid
```

Appendix E – MATLAB script objective 4, Scenario 1

```
% Demand Side Management for electric vehicles
% ------------------------------------------------------------------------
% DSM Algorithm Objective 4
% Master Thesis Spring 2015
% Author: Stian Namtvedt Gjelsvik
% Email: stian2803@gmail.com
% Version: 1.00
% Last update: 04.05.2015
% ------------------------------------------------------------------------
% Description:
% Smart charging algorithm based on demand side management (DSM).
% A demand limit provided by the distribution network operator is given
% and applies for the residential power consumer of the low voltage
% distribution network. In this case one residence is analysed.
% -----------------------------------------------------------------------
clc;
clear;
% Input:
P_charger = 3.3;   % Charging power (depending on the charger)
SOC=0.6666;        % State of charge. 21 kWh max. 7kWh used / Trans_Eff
Max_bat=21;        % Maximum capacity [kWh] battery
SOC_Recom=0.8;     % Recommended state of charge
Charge_Eff=0.93;   % Charge efficiency
Trans_Eff=0.865;   % Transportation efficiency
S_transformer = 150;
CosPhi = 0.9;      % External value from DIgSILENT PowerFactory
Phi = 25.8419327;  % CosPhi transformer lossless = CosPhi network
P_transformer = S_transformer*CosPhi;
U_Transformer = 238;
v_min=207;
v_max=253;

% Load profile for residence for the seasons, hourly based 00.00 to 00.00
Load_Summer = [0.97 0.97 0.97 0.97 0.97 0.97 1.62 3.24 3.24 0.81 0.86 0.86
0.86 0.86 0.86 0.86 3.88 3.53 2.16 2.16 2.16 1.94 1.62 1.62 0.97];
Load_Autumn = [1.58 1.58 1.58 1.58 1.58 1.58 2.97 4.45 4.45 1.58 1.58 1.58
1.58 1.58 1.58 2.97 5.93 5.93 5.93 5.08 4.45 3.95 2.97 2.97 1.58];
Load_Winter = [2.77 2.77 2.77 2.77 2.77 2.77 3.24 4.04 4.04 3.24 3.24 3.24
3.24 3.24 3.24 3.24 6.93 8.09 6.93 6.93 6.47 4.85 4.85 4.04 2.77];
Load_Spring = [1.29 1.29 1.29 1.29 1.29 1.29 1.29 2.16 2.16 1.48 1.48 1.48
1.48 1.48 1.48 1.48 4.71 4.31 3.70 3.24 2.88 2.59 2.35 2.16 1.29];

% User connection from 00.00 to 00.00, where 0 is disconnected and 1
% connected
User_Connection_1 = [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 ];
User_Connection_2 = [1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 ];
User_Connection_3 = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 1 ];
User_Connection_4 = [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 ];

Load = Load_Winter;
%Load=Load_Summer;
%Load=Load_Spring;
%Load=Load_Autumn;

% Demand limit 00.00 to 00.00
Demand_limit = [6.75 6.75 6.75 6.75 6.75 5 5 4 4 4 4 5 4 4 4 4 6 6 6 6.5
6.5 6.5 6.5 6.5 6.5 6.75];
```

```
% Loads (critical, heating and miscellaneous): 00.00 to 00.00
Load_Crit = Load*0.35;
Load_Heat = Load*0.50;
Load_Misc = Load*0.15;

% Chosen residence tag with EV:
% ********* 144502932 *********

% Constraints: --------------------------------------------------
% v_min <= v_n <= v_max
% v_min=207;
% v_max=253;
% I_b <= I_z  (Load current <= cable conductivity)
% --------------------------------------------------------------

% Cable data:
R_144502932= [0.0263 0.0032];
I_z_144502932=150;
% Feeding lines: Z = R + jX
F_80955_0= [0.0011 0.0006];
F_80955_0A= [0.0010 0.0006];
F_80955_5= [0.0199 0.0024];
F_80955_16= [0.0314 0.0074];
F_80955_17= [0.0600 0.0041];
F_80955_19= [0.0244 0.0029];
F_80955_22= [0.0853 0.0102];
F_80955_23= [0.0444 0.0030];
F_80955_A1= [0.0036 0.0002];
F_80955_24= [0.0552 0.0038];
F_80955_26= [0.1032 0.0071];
F_80955_18= [0.2124 0.0145];
F_80955_1= [0.0096 0.0023];
F_80955_1A= [0.0096 0.0023];
F_80955_9= [0.0493 0.0117];
F_80955_10= [0.0122 0.0029];
F_80955_14= [0.1308 0.0157];
F_80955_29= [0.0224 0.0027];
F_80955_4= [0.0769 0.0092];
TOT_144502932= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144502932;

% Initial values:
L_init_144502932=Load+(User_Connection_1*P_charger);
L_144502932=L_init_144502932;
Load_144502932=L_init_144502932;
U_init=230;
U_144502932=U_init;

EV_power=(1-SOC)*Max_bat/Trans_Eff;
Hour_charge_init=EV_power/(P_charger*Charge_Eff);
Hour_charge_144502932=Hour_charge_init;

I_b_144502932_init = (1000 * L_144502932(1)) / (U_144502932 * CosPhi*
sqrt(3));
I_b_144502932=I_b_144502932_init;

% Connection logic
Add_Misc_init=0;
Add_Misc=Add_Misc_init;
Tot_Misc_init=0;
```

```
Tot_Misc=Tot_Misc_init;

Add_Heat_init=0;
Add_Heat=Add_Heat_init;
Tot_Heat_init=0;
Tot_Heat=Tot_Heat_init;

Load_EV_init=0;
Load_EV=Load_EV_init;

%  Calculate constraints:
%  If maximum value of the demand limit creates overcurrent or voltage
%  the while loop will not run.
M=max(Demand_limit);
% Hourly based calculations constraints:
I_b_144502932 = (1000 * M) / (U_144502932 * CosPhi* sqrt(3));
% Delta voltage:
D_144502932 = sqrt(3)*(TOT_144502932(1)*I_b_144502932*cosd(Phi) +
TOT_144502932(2)*I_b_144502932*sind(Phi));
% Voltage at residence:
U_144502932 = U_Transformer - D_144502932;
% Loss caused by residence + EV
Loss_144502932=I_b_144502932.^2*TOT_144502932(1)+I_b_144502932.^2*TOT_14450
2932(2);
% 3. Residence with EV:
if I_b_144502932 <= I_z_144502932 & U_144502932 >= v_min & U_144502932 <=
v_max
C_144502932=1;  % Constraints OK
else
C_144502932=0;  % One or more constraints NOT ok
end
C_144502932=1;

if C_144502932==1
N=1;
elseif C_144502932==0
N=0; % Loop will not run!
end

while N<25
% SLM_EV_Objective4.m <---- update
% Use maximum desired loads within the demand limit.

% SOC - Connection logic (at the bottom)
%if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
%    Hour_charge_144502932 = EV_power/(P_charger*Charge_Eff);
%else Hour_charge_144502932 = Hour_charge_144502932;
%end

% Shifted connection logic
if Tot_Misc > 0
    Load_Misc_new=Tot_Misc+Load_Misc(N);
else Load_Misc_new=Load_Misc(N);
end

if Tot_Heat > 0
    Load_Heat_new=Tot_Heat+Load_Heat(N);
else Load_Heat_new=Load_Heat(N);
end
```

```
% Due to comfort postponed heating will be prioritized:
% Criteria: EV will have one hour of charging before heating
if Load_Heat_new > Demand_limit(N) & Load_EV==1;
        Load_Heat_new=Demand_limit(N)-Load_Crit(N);
        L_144502932 = Load_Heat_new + Load_Crit(N) ;
        Add_Heat=Load_Crit(N) ;
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Load_EX=1; % Specified load executed
        NULL=0;
elseif  Load_Misc_new > Demand_limit(N) & Load_EV==1;
        Load_Misc_new=Demand_limit(N) - Load_Crit(N);
        L_144502932=Load_Crit(N)+Load_Misc_new;
        Add_Misc=0;
        Load_EX=1; % Specified load executed
        NULL=1;

 % All ordinary + all shifted loads + EV charging:
elseif Hour_charge_144502932 > 0 & Load_Misc_new + Load_Heat_new +
Load_Crit(N) + P_charger < Demand_limit(N) & User_Connection_1(N) == 1
        L_144502932=Load_Misc_new + Load_Heat_new + Load_Crit(N) +
P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Misc=0;
        Add_Heat=0;
        Load_EX=1; % Specified load executed
        Load_EV=1;
        EN=1;
    % Ordinary loads + only shifted heat + EV
    elseif Hour_charge_144502932 > 0 & Load_Misc(N) + Load_Heat_new +
Load_Crit(N) + P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932=Load_Misc(N) + Load_Heat_new + Load_Crit(N) +
P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Heat=0;
        Load_EX=1; % Specified load executed
        Load_EV=1;
        TO=2;
     % All ordinary loads (no shifted) + EV
     elseif Hour_charge_144502932 > 0 & Load_Misc(N) + Load_Heat(N) +
Load_Crit(N) + P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932=Load_Misc(N) + Load_Heat(N) + Load_Crit(N) + P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Load_EX=1; % Specified load executed
        Load_EV=1;
        TRE=3;
 % Crit, heat + EV charging: (2/3 ordinary loads)
        % Shifted + ordinary heat + EV
        elseif Hour_charge_144502932 > 0 & Load_Crit(N)+ Load_Heat_new +
P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932= Load_Crit(N)+Load_Heat_new+ P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Heat=0;
        EV_144502932=1;
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Load_EX=1; % Specified load executed
        Load_EV=1;
        FIRE=4;
        % Ordinary heat + crit + EV
        elseif Hour_charge_144502932 > 0 & Load_Crit(N)+ Load_Heat(N) +
P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932= Load_Crit(N)+Load_Heat(N)+ P_charger;
```

```
        Load_EX=1; % Specified load executed
        Load_EV=1;
        FEM=5;
        % Crit + Misc + shifted misc + EV
        elseif Hour_charge_144502932 > 0 & Load_Crit(N) + Load_Misc_new +
P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932= Load_Crit(N)+ Load_Misc_new + P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Misc=0;
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EV=1;
        Load_EX=1; % Specified load executed
        SEKS=6;
    % Crit + Misc + EV (to maximize load if possible)
    elseif Hour_charge_144502932 > 0 & Load_Crit(N) + Load_Misc(N) +
P_charger < Demand_limit(N) & User_Connection_1(N)==1
        L_144502932= Load_Crit(N)+ Load_Misc(N) + P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EV=1;
        Load_EX=1; % Specified load executed
        SYV=7;
  % Crit + EV charging
  elseif Hour_charge_144502932 > 0 & Load_Crit(N) + P_charger <
Demand_limit(N) & User_Connection_1(N)==1
        L_144502932= Load_Crit(N)+ P_charger;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EV=1;
        Load_EX=1; % Specified load executed
        OTTE=8;
  % ------------ Without EV --------------
   % All ordinary + all shifted loads
  elseif Hour_charge_144502932 < 0 & Load_Misc_new + Load_Heat_new +
Load_Crit(N) < Demand_limit(N)
        L_144502932=Load_Misc_new + Load_Heat_new + Load_Crit(N);
        Add_Misc=0;
        Add_Heat=0;
        Load_EX=1; % Specified load executed
        Load_EV=0;
        NI=9;
     % Ordinary loads + only shifted heat
     elseif Hour_charge_144502932 < 0 & Load_Misc(N) + Load_Heat_new +
Load_Crit(N) < Demand_limit(N)
        L_144502932=Load_Misc(N) + Load_Heat_new + Load_Crit(N);
        Add_Heat=0;
        Load_EX=1; % Specified load executed
        Load_EV=0;
        TI=10;
     % All ordinary loads (no shifted)
     elseif Hour_charge_144502932 < 0 & Load_Misc(N) + Load_Heat(N) +
Load_Crit(N) < Demand_limit(N)
        L_144502932=Load_Misc(N) + Load_Heat(N) + Load_Crit(N);
        Load_EX=1; % Specified load executed
        Load_EV=0;
        ELLEVE=11;
 % Crit, heat
        % shifted + ordinary heat
elseif Hour_charge_144502932 < 0 & Load_Crit(N)+ Load_Heat_new  <
Demand_limit(N)
```

```
        L_144502932= Load_Crit(N)+Load_Heat_new;
        Add_Heat=0;
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Load_EX=1; % Specified load executed
        Load_EV=0;
        TOLV=12;
        % Only ordinary crit + heat
elseif Hour_charge_144502932 < 0 & Load_Crit(N)+ Load_Heat(N) <
Demand_limit(N)
        L_144502932= Load_Crit(N)+Load_Heat(N);
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Load_EX=1; % Specified load executed
        Load_EV=0;
        TRETTEN=13;
   % Crit + Misc + shifted misc + EV
 elseif Hour_charge_144502932 < 0 & Load_Crit(N) + Load_Misc_new <
Demand_limit(N)
        L_144502932= Load_Crit(N)+ Load_Misc_new;
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Misc=0;
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EX=1; % Specified load executed
        Load_EV=0;
        FJORTEN=14;
    % Crit + Misc + EV
    elseif Hour_charge_144502932 < 0 & Load_Crit(N) + Load_Misc(N) <
Demand_limit(N)
        L_144502932= Load_Crit(N)+ Load_Misc(N);
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EX=1; % Specified load executed
        Load_EV=0;
        FEMTEN=15;
  % Crit
  elseif Hour_charge_144502932 > 0 & Load_Crit(N)< Demand_limit(N)
        L_144502932= Load_Crit(N);
        Hour_charge_144502932=Hour_charge_144502932-1;
        Add_Misc=[Tot_Misc Load_Misc(N)];
        Add_Heat=[Tot_Heat Load_Heat(N)];
        Load_EX=1; % Specified load executed
        Load_EV=0;
        SEKSTEN=16;
end

  % Sum misc and heat
        Tot_Misc=sum(Add_Misc);
        Tot_Heat=sum(Add_Heat);

  % Maximize to reach demand limit:
   if L_144502932<Demand_limit(N) & Tot_Heat > Demand_limit(N)-L_144502932
& Tot_Heat > Tot_Misc
        Load_Heat_max=Demand_limit(N)- L_144502932;
        Add_Heat=Tot_Heat-Load_Heat_max;
        L_144502932=Demand_limit(N);
        SYTTEN=17;
   elseif L_144502932<Demand_limit(N) & Tot_Misc > Demand_limit(N)-
L_144502932
        Load_Misc_new=Demand_limit(N)-L_144502932;
        Add_Misc=Tot_Misc-Load_Misc_new;
        L_144502932 = Demand_limit(N) ;
        ATTEN=18;
```

```
   end

  % Sum misc and heat to the next hour
        Tot_Misc=sum(Add_Misc);
        Tot_Heat=sum(Add_Heat);

% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144502932 = EV_power/(P_charger*Charge_Eff);
else Hour_charge_144502932 = Hour_charge_144502932;
end

N=N+1;

end
```

Appendix F – MATLAB script objective 4, Scenario 2

```
% Demand Side Management for electric vehicles
% -----------------------------------------------------------------------
% DSM Algorithm Objective 4 One User Minute based
% Master Thesis Spring 2015
% Author: Stian Namtvedt Gjelsvik
% Email: stian2803@gmail.com
% Version: 1.00
% Last update: 17.03.2015
% -----------------------------------------------------------------------
% Description:
% Smart charging algorithm based on demand side management (DSM).
% A demand limit provided by the distribution network operator is given
% and applies for the residential power consumer of the low voltage
% distribution network. In this case one residence is analysed.
% ----------------------------------------------------------------------
clc;
clear;
close all;
% Input:
P_charger = 3.3;    % Charging power (depending on the charger)
SOC=0.6666;         % State of charge. 21 kWh max. 7kWh used / Trans_Eff
Max_bat=21;         % Maximum capacity [kWh] battery
SOC_Recom=0.8;      % Recommended state of charge
Charge_Eff=0.93;    % Charge efficiency
Trans_Eff=0.865;    % Transportation efficiency
S_transformer = 150;
CosPhi = 0.9;       % External value from DIgSILENT PowerFactory
Phi = 25.8419327;   % CosPhi transformer lossless = CosPhi network
P_transformer = S_transformer*CosPhi;
U_Transformer = 238;
v_min=207;
v_max=253;

% Load profile for residence for the seasons, hourly based 00.00 to 00.00
Load_Summer = [0.97 0.97 0.97 0.97 0.97 0.97 1.62 3.24 3.24 0.81 0.86 0.86
0.86 0.86 0.86 0.86 3.88 3.53 2.16 2.16 2.16 1.94 1.62 1.62 0.97];
Load_Autumn = [1.58 1.58 1.58 1.58 1.58 1.58 2.97 4.45 4.45 1.58 1.58 1.58
1.58 1.58 1.58 2.97 5.93 5.93 5.93 5.08 4.45 3.95 2.97 2.97 1.58];
Load_Winter = [2.77 2.77 2.77 2.77 2.77 2.77 3.24 4.04 4.04 3.24 3.24 3.24
3.24 3.24 3.24 3.24 6.93 8.09 6.93 6.93 6.47 4.85 4.85 4.04 2.77];
Load_Spring = [1.29 1.29 1.29 1.29 1.29 1.29 1.29 2.16 2.16 1.48 1.48 1.48
1.48 1.48 1.48 1.48 4.71 4.31 3.70 3.24 2.88 2.59 2.35 2.16 1.29];

% User connection from 00.00 to 00.00, where 0 is disconnected and 1
% connected
% User_Connection_1_HOUR = [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1
1 ];
% User_Connection_1_Minute:
User_Connection_1 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
%Load = Load_Winter;
%Load=Load_Summer;
%Load=Load_Spring;
%Load=Load_Autumn;

% Demand limit and loads -  00.00 to 00.00  - 15 minute periods
Demand_Limit = [8.0 7.9 7.9 7.8 7.5 7.9 7.5 7.6 7.4 8.0 8.1 7.3 7.6 7.4 7.8
7.0 7.0 7.7 7.7 6.5 5.0 4.2 5.0 4.2 4.8 4.3 4.2 4.8 4.3 4.3 5.2 4.2 4.9 4.2
4.4 4.5 4.7 4.2 6.0 4.9 5.3 5.6 4.0 4.0 4.2 4.5 4.2 4.1 4.0 5.0 4.9 4.8 4.5
5.0 3.9 4.2 4.5 4.9 5.9 7.9 8.8 9.0 8.5 8.4 6.5 7.0 7.5 7.9 6.7 6.5 6.6 7.0
7.0 6.9 5.7 5.8 5.4 6.0 5.9 5.0 5.4 5.0 5.2 6.6 6.7 7.9 6.9 7.5 8.0 8.3 7.9
7.9 8.0 8.9 7.9 7.8 7.5];
Load=          [2.5 2.6 2.6 2.7 2.8 1.8 1.9 2.0 3.0 2.9 2.9 1.2 2.1 2.5 2.5
2.2 2.0 2.2 3.0 2.5 3.0 2.9 2.0 4.0 4.2 3.6 3.9 4.0 4.2 3.2 4.0 2.5 3.0 3.1
3.5 4.2 3.0 1.5 2.5 1.9 2.0 2.1 2.5 3.2 1.0 1.5 3.0 3.1 3.1 2.8 2.5 2.6 2.8
2.7 2.8 2.9 3.2 3.2 3.3 3.9 7.2 8.8 5.2 4.2 6.0 6.0 7.0 3.5 4.5 4.0 3.1 3.2
4.2 3.6 3.6 3.9 4.0 4.6 4.9 5.0 3.0 3.0 3.1 2.9 2.9 4.5 7.0 3.0 2.1 3.0 3.5
4.6 3.9 2.5 4.9 2.6 2.7 ];

% Chosen residence tag with EV:
% ********* 144502932 *********

% Constraints: ---------------------------------------------
% v_min <= v_n <= v_max
% v_min=207;
% v_max=253;
% I_b <= I_z  (Load current <= cable conductivity)
% ---------------------------------------------------------

% Cable data:
```

```
R_144502932= [0.0263 0.0032];
I_z_144502932=150;
% Feeding lines: Z = R + jX
F_80955_0= [0.0011 0.0006];
F_80955_0A= [0.0010 0.0006];
F_80955_5= [0.0199 0.0024];
F_80955_16= [0.0314 0.0074];
F_80955_17= [0.0600 0.0041];
F_80955_19= [0.0244 0.0029];
F_80955_22= [0.0853 0.0102];
F_80955_23= [0.0444 0.0030];
F_80955_A1= [0.0036 0.0002];
F_80955_24= [0.0552 0.0038];
F_80955_26= [0.1032 0.0071];
F_80955_18= [0.2124 0.0145];
F_80955_1= [0.0096 0.0023];
F_80955_1A= [0.0096 0.0023];
F_80955_9= [0.0493 0.0117];
F_80955_10= [0.0122 0.0029];
F_80955_14= [0.1308 0.0157];
F_80955_29= [0.0224 0.0027];
F_80955_4= [0.0769 0.0092];
TOT_144502932= F_80955_29 + F_80955_14 + F_80955_10 + F_80955_9 +
F_80955_1A + F_80955_0A + R_144502932;

% Initial values:
L_init_144502932=Load+(User_Connection_1(1)*P_charger);
L_144502932=L_init_144502932;
Load_144502932=L_init_144502932;
U_init=230;
U_144502932=U_init;

% Minute based charging time:
EV_power=(1-SOC)*Max_bat/Trans_Eff;
Hour_charge_init=EV_power/(P_charger*Charge_Eff)*60;
Hour_charge_144502932=Hour_charge_init;

I_b_144502932_init = (1000 * L_144502932) / (U_144502932 * CosPhi*
sqrt(3));
I_b_144502932=I_b_144502932_init;

N=1;
while N<1441  % 24 Hours * 60 minutes
%  Only shift EV
% --------------------------
%  Logic 15 minute periods into 1 minute:
if N<16
    Load_Residence=Load(1);
    Demand_limit=Demand_Limit(1);
elseif N<31
    Load_Residence=Load(2);
    Demand_limit=Demand_Limit(2);
elseif N<46
    Load_Residence=Load(3);
    Demand_limit=Demand_Limit(3);
elseif N<61
    Load_Residence=Load(4);
    Demand_limit=Demand_Limit(4);
    elseif N<76
    Load_Residence=Load(5);
    Demand_limit=Demand_Limit(5);
```

```
elseif N<91
Load_Residence=Load(6);
Demand_limit=Demand_Limit(6);
elseif N<106
Load_Residence=Load(7);
Demand_limit=Demand_Limit(7);
elseif N<121
Load_Residence=Load(8);
Demand_limit=Demand_Limit(8);
elseif N<136
Load_Residence=Load(9);
Demand_limit=Demand_Limit(9);
elseif N<151
Load_Residence=Load(10);
Demand_limit=Demand_Limit(10);
elseif N<166
Load_Residence=Load(11);
Demand_limit=Demand_Limit(11);
elseif N<181
Load_Residence=Load(12);
Demand_limit=Demand_Limit(12);
elseif N<196
Load_Residence=Load(13);
Demand_limit=Demand_Limit(13);
elseif N<211
Load_Residence=Load(14);
Demand_limit=Demand_Limit(14);
elseif N<226
Load_Residence=Load(15);
Demand_limit=Demand_Limit(15);
elseif N<241
Load_Residence=Load(16);
Demand_limit=Demand_Limit(16);
elseif N<256
Load_Residence=Load(17);
Demand_limit=Demand_Limit(17);
elseif N<271
Load_Residence=Load(18);
Demand_limit=Demand_Limit(18);
elseif N<286
Load_Residence=Load(19);
Demand_limit=Demand_Limit(19);
elseif N<301
Load_Residence=Load(20);
Demand_limit=Demand_Limit(20);
elseif N<316
Load_Residence=Load(21);
Demand_limit=Demand_Limit(21);
elseif N<331
Load_Residence=Load(22);
Demand_limit=Demand_Limit(22);
elseif N<346
Load_Residence=Load(23);
Demand_limit=Demand_Limit(23);
elseif N<361
Load_Residence=Load(24);
Demand_limit=Demand_Limit(24);
elseif N<376
Load_Residence=Load(25);
Demand_limit=Demand_Limit(25);
elseif N<391
```

```
Load_Residence=Load(26);
Demand_limit=Demand_Limit(26);
elseif N<406
Load_Residence=Load(27);
Demand_limit=Demand_Limit(27);
elseif N<421
Load_Residence=Load(28);
Demand_limit=Demand_Limit(28);
elseif N<436
Load_Residence=Load(29);
Demand_limit=Demand_Limit(29);
elseif N<451
Load_Residence=Load(30);
Demand_limit=Demand_Limit(30);
elseif N<466
Load_Residence=Load(31);
Demand_limit=Demand_Limit(31);
elseif N<481
Load_Residence=Load(32);
Demand_limit=Demand_Limit(32);
elseif N<496
Load_Residence=Load(33);
Demand_limit=Demand_Limit(33);
elseif N<511
Load_Residence=Load(34);
Demand_limit=Demand_Limit(34);
elseif N<526
Load_Residence=Load(35);
Demand_limit=Demand_Limit(35);
elseif N<541
Load_Residence=Load(36);
Demand_limit=Demand_Limit(36);
elseif N<556
Load_Residence=Load(37);
Demand_limit=Demand_Limit(37);
elseif N<571
Load_Residence=Load(38);
Demand_limit=Demand_Limit(38);
elseif N<586
Load_Residence=Load(39);
Demand_limit=Demand_Limit(39);
elseif N<601
Load_Residence=Load(40);
Demand_limit=Demand_Limit(40);
elseif N<616
Load_Residence=Load(41);
Demand_limit=Demand_Limit(41);
elseif N<631
Load_Residence=Load(42);
Demand_limit=Demand_Limit(42);
elseif N<646
Load_Residence=Load(43);
Demand_limit=Demand_Limit(43);
   elseif N<661
Load_Residence=Load(44);
Demand_limit=Demand_Limit(44);
elseif N<676
Load_Residence=Load(45);
Demand_limit=Demand_Limit(45);
elseif N<691
Load_Residence=Load(46);
```

```
Demand_limit=Demand_Limit(46);
elseif N<706
Load_Residence=Load(47);
Demand_limit=Demand_Limit(47);
elseif N<721
Load_Residence=Load(48);
Demand_limit=Demand_Limit(48);
elseif N<736
Load_Residence=Load(49);
Demand_limit=Demand_Limit(49);
elseif N<751
Load_Residence=Load(50);
Demand_limit=Demand_Limit(50);
elseif N<766
Load_Residence=Load(51);
Demand_limit=Demand_Limit(51);
elseif N<781
Load_Residence=Load(52);
Demand_limit=Demand_Limit(52);
    elseif N<796
Load_Residence=Load(53);
Demand_limit=Demand_Limit(53);
elseif N<811
Load_Residence=Load(54);
Demand_limit=Demand_Limit(54);
elseif N<826
Load_Residence=Load(55);
Demand_limit=Demand_Limit(55);
elseif N<841
Load_Residence=Load(56);
Demand_limit=Demand_Limit(56);
elseif N<856
Load_Residence=Load(57);
Demand_limit=Demand_Limit(57);
elseif N<871
Load_Residence=Load(58);
Demand_limit=Demand_Limit(58);
elseif N<886
Load_Residence=Load(59);
Demand_limit=Demand_Limit(59);
   elseif N<901
Load_Residence=Load(60);
Demand_limit=Demand_Limit(60);
elseif N<916
Load_Residence=Load(61);
Demand_limit=Demand_Limit(61);
elseif N<931
Load_Residence=Load(62);
Demand_limit=Demand_Limit(62);
elseif N<946
Load_Residence=Load(63);
Demand_limit=Demand_Limit(63);
elseif N<961
Load_Residence=Load(64);
Demand_limit=Demand_Limit(64);
elseif N<976
Load_Residence=Load(65);
Demand_limit=Demand_Limit(65);
elseif N<991
Load_Residence=Load(66);
Demand_limit=Demand_Limit(66);
```

```
elseif N<1006
Load_Residence=Load(67);
Demand_limit=Demand_Limit(67);
elseif N<1021
Load_Residence=Load(68);
Demand_limit=Demand_Limit(68);
    elseif N<1036
Load_Residence=Load(69);
Demand_limit=Demand_Limit(69);
elseif N<1051
Load_Residence=Load(70);
Demand_limit=Demand_Limit(70);
elseif N<1066
Load_Residence=Load(71);
Demand_limit=Demand_Limit(71);
elseif N<1081
Load_Residence=Load(72);
Demand_limit=Demand_Limit(72);
elseif N<1096
Load_Residence=Load(73);
Demand_limit=Demand_Limit(73);
elseif N<1111
Load_Residence=Load(74);
Demand_limit=Demand_Limit(74);
elseif N<1126
Load_Residence=Load(75);
Demand_limit=Demand_Limit(75);
   elseif N<1141
Load_Residence=Load(76);
Demand_limit=Demand_Limit(76);
elseif N<1156
Load_Residence=Load(77);
Demand_limit=Demand_Limit(77);
elseif N<1171
Load_Residence=Load(78);
Demand_limit=Demand_Limit(78);
elseif N<1186
Load_Residence=Load(79);
Demand_limit=Demand_Limit(79);
elseif N<1201
Load_Residence=Load(80);
Demand_limit=Demand_Limit(80);
elseif N<1216
Load_Residence=Load(81);
Demand_limit=Demand_Limit(81);
elseif N<1231
Load_Residence=Load(82);
Demand_limit=Demand_Limit(82);
elseif N<1246
Load_Residence=Load(83);
Demand_limit=Demand_Limit(83);
elseif N<1261
Load_Residence=Load(84);
Demand_limit=Demand_Limit(84);
elseif N<1276
Load_Residence=Load(85);
Demand_limit=Demand_Limit(85);
elseif N<1291
Load_Residence=Load(86);
Demand_limit=Demand_Limit(86);
elseif N<1306
```

```
    Load_Residence=Load(87);
    Demand_limit=Demand_Limit(87);
    elseif N<1321
    Load_Residence=Load(88);
    Demand_limit=Demand_Limit(88);
    elseif N<1336
    Load_Residence=Load(89);
    Demand_limit=Demand_Limit(89);
    elseif N<1351
    Load_Residence=Load(90);
    Demand_limit=Demand_Limit(90);
    elseif N<1366
    Load_Residence=Load(91);
    Demand_limit=Demand_Limit(91);
    elseif N<1381
    Load_Residence=Load(92);
    Demand_limit=Demand_Limit(92);
    elseif N<1396
    Load_Residence=Load(93);
    Demand_limit=Demand_Limit(93);
    elseif N<1411
    Load_Residence=Load(94);
    Demand_limit=Demand_Limit(94);
    elseif N<1426
    Load_Residence=Load(95);
    Demand_limit=Demand_Limit(95);
    elseif N<1441
    Load_Residence=Load(96);
    Demand_limit=Demand_Limit(96);
end

% Minute base calculations constraints:
if User_Connection_1(N)==1
    Load_tot=Load_Residence+P_charger;
elseif User_Connection_1(N)==0
    Load_tot=Load_Residence;
end
I_b_144502932 = (1000 * Load_tot) / (U_144502932 * CosPhi* sqrt(3));
% Delta voltage:
D_144502932 = sqrt(3)*(TOT_144502932(1)*I_b_144502932*cosd(Phi) +
TOT_144502932(2)*I_b_144502932*sind(Phi));
% Voltage at residence:
U_144502932 = U_Transformer - D_144502932;
% Loss caused by residence + EV
Loss_144502932=I_b_144502932.^2*TOT_144502932(1)+I_b_144502932.^2*TOT_14450
2932(2);

if User_Connection_1(N)==1 & Hour_charge_144502932 > 0 & I_b_144502932 <=
I_z_144502932 & U_144502932 >= v_min & U_144502932 <= v_max &
Demand_limit>Load_tot
    Hour_charge_144502932=Hour_charge_144502932-1;
    L_144502932=Load_tot;
    EV_charge=1;
else L_144502932=Load_Residence;
    EV_charge=0;
end

if User_Connection_1(N)==1 &  Hour_charge_144502932 > 0
    EV_charge_wish=Load_Residence+P_charger;
    EV_charge_org=1;
else
```

```
    EV_charge_org=0;
end

% SOC - Connection logic
if User_Connection_1(N+1) == 1 & User_Connection_1(N) == 0
    Hour_charge_144502932 = EV_power/(P_charger*Charge_Eff)*60;
else Hour_charge_144502932 = Hour_charge_144502932;
end

Demand_limit;
Time_Plot(N)=N;
Load_Plot(N)=Load_Residence;
EV_charge_Plot(N)=EV_charge;
EV_charge_org_Plot(N)=EV_charge_org;
L_144502932_Plot(N)=L_144502932;
Demand_limit_Plot(N)=Demand_limit;
EV_charge_wish_Plot(N)=EV_charge_wish;
User_Connection_1_Plot(N)=User_Connection_1(N);

N=N+1;

end

plot(Time_Plot, Load_Plot,'r','LineWidth',2)
hold on
plot(Time_Plot, L_144502932_Plot)
hold on
plot(Time_Plot, Demand_limit_Plot,'g')
grid
%plot(Time_Plot, EV_charge_Plot,'r')
%hold on
%plot(Time_Plot, User_Connection_1_Plot)
grid
```