Open Source Module for the Investigation of the Impact of Electric Vehicles in a Low Voltage Grid

Luca Giorgi, Artjoms Obushevs, Petr Korba
Institute of Energy Systems and Fluid Engineering
Zurich University of Applied Science
Winterthur, Switzerland
obus@zhaw.ch, korb@zhaw.ch

***Abstract*—This paper provides an open-source Python-based module overview focused on simulating the integration of electric vehicles in a low voltage grid. This module aims to investigate the possible effects that the integration of electric vehicles could have on the operability of the power network. The electric grid conditions are estimated by analysing line loading, voltage values at the final customer, and transformers' loading. The following tool enables modelling electric grids composed of basic grid elements such as lines, two-winding transformers, predefined load profiles and generated electric vehicle load profiles based on a statistical approach. The module performs time-series simulations through a secondary software, OpenDSS, with result exporting functionality for further analysis, and a graphical user interface.**

***Keywords*—distribution system impacts, electric vehicles, power distribution, power system modelling, quasi-dynamic time-series analysis**

I. Introduction

The actual climate conditions, which result in many phenomena such as global warming, require effective and sometimes significant actions to limit these negatives outcomes [1]. According to the Swiss Federal Office for Environment [2], the main action is the reduction of greenhouse gas emissions, particularly in the sectors with the highest reduction potential, such as transports, electrical production/consumption, and heating systems. Developing and improving new or existing eco-friendly technologies, such as power generators based on renewable energies (RES) and Electric Vehicles (EVs), is constantly growing. Countries are showing great interest and effort in regards to this problem, by applying support and investment subsidies for the construction of new power generators based on RES as well as by promoting electromobility [3], or by planning a phase-out of the sales of Internal Combustion Engine (ICE) cars within the next twenty years [4]. In Switzerland, the Swiss Federal Government (SFG) is active at various stages to improve the framework conditions for alternative propulsion systems by applying legal measures such as the introduction in 2012 of the Swiss CO_2 law [5], and to develop pilot projects in cooperation with municipalities [2].

The growth of the share of EVs is a part of the solution against global warming [6]. However, their improvement is not without consequences. With the increase of EVs, electric energy demand will also increase, leading to a possible increase in daily energy consumption. Moreover, an increase in power peak can appear. These incrementations can harm the functionality of power systems by affecting the quality and safety of the grid. Thus, also considering the integration of decentralised power generators, the low-voltage distribution grids (LVDG) will be pushed to their limits [7],[8],[9]. Existing open-source and commercial software don't have modules that allow considering EV load profiles and combining them with existing standard load profiles (SLP) or smart meter data. Therefore, the main goal of this paper is to present an open-source module (OSM) able to simulate and estimate the impact that can have the integration of EVs in an LVDG by analysing the overload of lines, voltage level at the Point of Common Coupling (PCC), an overload of transformers. Different open-source software and libraries, which allow the power flow calculation, are tested and compared. The best solution is integrated with Electric Vehicle Load Profiles Generator (EVLPG)[10]. With final software, studies and tests are performed to define the simplifications that can be implemented in the software. These simplifications make possible the achievement of plausible results by reducing the number of simulations and time.

The work structure is following. The second chapter introduces the approach used to realise the OSM, a comparison of several power analysis tools, the structure of the OSM, and a simple algorithm that allows reducing the time needed to simulate. The third chapter provides results concerning the empirical analysis of possible simplifications, a demonstration of its functionality, and limitations. The final chapter includes an interpretation of the results and a discussion of the functionality of the OSM.

II. Approach and Methodology

The following chapter presents the approach and methodologies used in the proposed module.

A. Global Approach

The first phase is about the research and selection of alternative software for power system modelling. In particular, two available libraries for Python – which are PandaPower [11] and PyPSA [12] – and an open-source simulator called OpenDSS [13]. These tools were tested in different conditions (grid model and timeframe of the simulation), and results have been compared with outcomes obtained using

The published version of this author accepted manuscript can be found via DOI: 10.1109/RTUCON53541.2021.9711593

PowerFactory (PF) [14], which acts as a reference. The criteria considered for the decision of the final tool are the time needed to simulate, the exactitude of results compared to PF, and the facility to use and implement the software. Once defined the tool for power system modelling, the second phase of the approach takes place. Here, the EVLPG is extended, improved, and modified to communicate with the new tool – OpenDSS – by giving a form to the final module. This module is then improved and adapted to be simple (user-friendly), easy to understand, and versatile. Based on the complexity of the grid and the percentage of EV integration, simulations need time (minutes or even days). For this reason, studies about simplifications on the simulation method have been done to obtain almost the same results by reducing the simulation time. Once obtained a completed module by applying these simplifications, different integration scenarios in a complex grid have been simulated using the new module and PF. Results are then compared to prove the functionality and the effectiveness of this new powerful tool.

B. Power Flow Calculation tools

This subchapter presents various power grid modelling tools selected for this work as well as their functionality and properties. After that, a complete comparison between software is presented, and the best option is chosen.

Nowadays, different open-source tools regarding power systems analysis are available, distinguished by different characteristics, domains of application, and calculation methods. Software should manage power flow and time-series calculations and the capability to model grid elements. Another fundamental point that makes the integration of the EVLPG easy is that software should be based, or at least compatible, with Python since the EVLPG has been developed using this programming language. The selected tools are Pandapower, PyPSA and OpenDSS.

This comparison aims to select the most appropriate alternative software that can replace the actual PF. The final choice has been made by considering several aspects regarding all involved software. These criteria are listed below:

- The facility to use the tool (Programming): how grid's elements and simulation parameters are defined (parameters of elements, time step, time frame, method, etc.), and the tool's flexibility to be adapted for a versatile functionality.
- The exactitude of results (time-series simulation): the chosen software should compete with commercial tools by achieving the same, or almost the same, results.
- Time of simulation: the time needed to simulate plays an important role in selecting the tool. The alternative software should be in the same range, or even low, compared to commercial software. This characteristic allows realising an efficient and attractive OSM.
- Result analysis: the facility to export results of specific variables. The variables used to estimate the impact on a LVDG are loading lines, voltage magnitude at the PCCs, and loading of transformers.
- Future implementations: interesting functionalities that can be exploited to improve the module developed. An example is the possibility to perform unbalanced power flow calculations.

Classification from the best to the worst was defined, and a more suitable tool was chosen according to the final rating shown in TABLE I.

TABLE I. SUMMARY OF CLASSIFICATIONS OF ALTERNATIVE TOOLS.

	Programming	Precision of results	Time	Results analysis	Improvements
1	OpenDSS	PyPSA	OpenDSS	Pandapower	OpenDSS
2	Pandapower	OpenDSS	Pandapower	OpenDSS	Pandapower
3	PyPSA	Pandapower	PyPSA	PyPSA	PyPSA

All five criteria play a role in the choice of the final tool. However, the most important parameters are the time of simulations and the exactitude of results compared with PF. In terms of precision of results (Simulations), PyPSA is the leading software. However, the distinction between the three software is not noticeable. In contrast, simulation time indicates a significant distinction between software, with OpenDSS being significantly faster. Therefore, OpenDSS provides the best compromise between the precision of results and the time of the simulation. Besides, according to the classification "Improvements", OpenDSS stands out as it also offers the possibility to introduce unbalanced power flow calculation. In addition, OpenDSS is the most user-friendly software. However, once the necessary programming experience is acquired, all alternative software can be employed appropriately and thus, this criterion is less relevant. Following this discussion, OpenDSS has been chosen as alternative software.

Note that this subselection is based on specific criteria and application. Consequently, Pandapower and PyPSA may be better options for other kinds of applications.

C. Open-Source Module

In this section, the OSM is presented. Firstly, a general insight about its components and functionalities. Secondly, an in-depth explanation focused on each stage of the OSM is provided. The OSM is available at the following link [15].

1) General Overview

The module is divided into four stages as depicted in Fig. 1. Results analysis is an additional part that is not included in the module. Hence, the user should analyse results manually through an analysis tool (e.g. Matlab [16]).

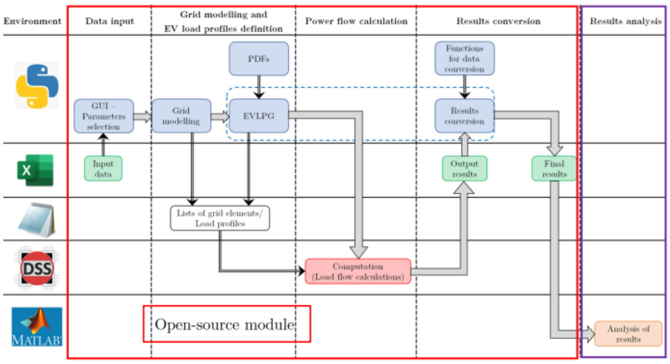

Fig. 1. Flowchart diagram of the functionality of the OSM.

Initially, the user defines the input parameters necessary to determine the conditions of simulation (grid elements, SLPs, timestep, timeframe, etc.). This selection is realised through a graphical interface. Once established the input data, the second step takes place. During this part, the power grid is modelled by defining the elements of the grid with their relative parameters. Subsequently, the EVLPG provides – following the input parameters and probability density functions – load profiles to allocate within specific points in the grid. All these parameters are then exported in external text files. These external text files are imported in OpenDSS, and a time-series simulation is performed. The results calculated during the time-series simulation are then exported as a CSV file. The fourth step provides the conversion of results from currents of lines and transformers into loading. Afterwards, results are exported in CSV form.

The whole module comprises several files (scripts), as shown in Fig. 2, where each script is designed for a specific procedure. This division offers the possibility to realise a global module easy to understand and modify. However, the main script is the "*Master.py*" script. Therefore, simulations are performed through this main script, and the user does not need to have access to the other scripts.

2) Grid Modelling

This module functionality is defined in the script "Grid_Modelling.py" and is responsible for the transaction of grid elements and SLPs from the excel file to the specific format for OpenDSS. Moreover, additional elements are created, such as monitors and energy meters. Monitors allow the measurement of power, voltage, and current, allocated to each element. Otherwise, without monitors, OpenDSS will not provide results. Energy meters allow extracting the distances of the PCC from the point where these energy meters are installed (from the main busbar). These values of distances are used to select potential PCC where to connect charging stations. The structure of this part can be classified into three steps (Fig. 3). The first one concerns importing Excel files. Successively, variables and matrices are defined. Each topology of an element has its matrix. N rows and one column form these matrices. The number of rows depends on the total number of elements of a specific topology. In the case of the Stadtwerk Winterthur (SW) grid [8], the total number of loads is 63, thus, the matrix dedicated to the loads (named "*loads*") would be composed of one column and 63 rows, as shown in Fig. 4.

The main part of this script is step three. Here, elements are converted from the excel file to a python variable (into the matrices). Afterwards, the matrices are saved in text files. A specific matrix defines each topology of the element, also text files are dived into these topologies.

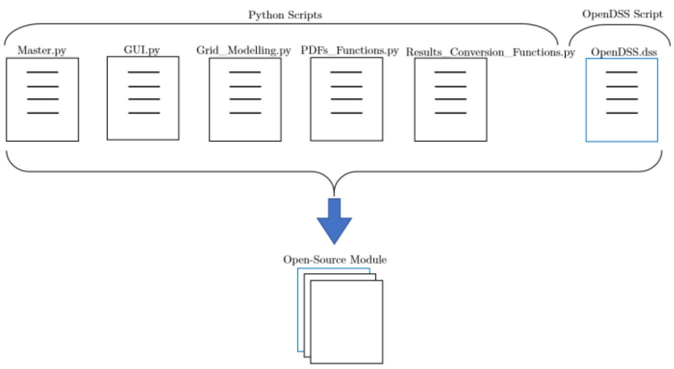

Fig. 2. Structure of the OSM.

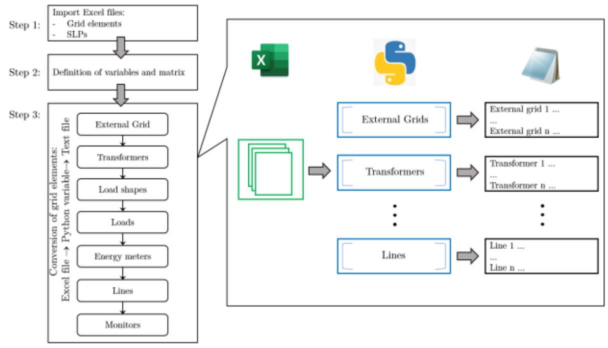

Fig. 3. Structure of the script "*Grid_Modelling.py*".

```
In [13]: loads
Out[13]:
['New Load.Load1 phases=3 bus1=A_001 conn=wye kv=0.4 kw=2.2721301400000002 kvar=0 daily=H0 Vmaxpu=1.2 Vminpu=0.8',
 'New Load.Load2 phases=3 bus1=A_002 conn=wye kv=0.4 kw=0.45648878800000003 kvar=0 daily=H0 Vmaxpu=1.2 Vminpu=0.8',
 'New Load.Load3 phases=3 bus1=A_003 conn=wye kv=0.4 kw=2.4443706 kvar=0 daily=H0 Vmaxpu=1.2 Vminpu=0.8',
```

Fig. 4. Example of the composition of the matrix dedicated to the loads.

3) EV Load Profiles Generator

This is the main part that distinguishes this OSM from the other power analysis software by providing a module purely focused on an advanced generation of quasi-realistic EVs load profiles. The particularity of the generator of load profiles is that load profiles are created autonomously according to several fundamental parameters that characterise the charge of an EV. The EVLPG [10] is implemented in the script "*Master.py*".

The distribution of charging fleets (CF) across the grid is composed of two steps. The first one is related to the definition of all potential PCC where CF could be installed. The second step defines exactly in which PCC CF will be installed according to the parameter of integration. This module provides three methods for defining potential PCC:

- Based on the grid (Grid): all PCC in the grid are considered as potential points where CF could be installed.
- Defined by the user (External File): potential PCCs are defined exclusively according to an external file containing the desired PCC list. This method allows realising a focused simulation by applying CF in specific zones of the grid.
- Considering both previous methods (Mix): in this case, it is possible to define already existing CF through an external file. Besides, the module will consider all PCC as potential points, except for the PCC already defined by the user.

Once defined, the potential PCC, CF are distributed across the grid, and the configuration of this distribution will influence the results of the simulation. Hence, three approaches that define the methodology of distribution are available:

- Considering all possible distributions (All combinations): The exact distribution of a defined number of CF across the grid cannot be predicted. Therefore, all possible distribution scenarios are simulated to obtain an overall impact that a defined percentage of integration of EVs could have on the grid.

- Randomly distributed (Random): CFs are distributed randomly across the grid. It is possible to simulate n different distributions, where *the user defines n*. This approach reduces the number of distributions and thus the number of simulations and the time needed to simulate. However, if the number of distributions is not sufficiently elevated, the results cannot be associated with an average impact as in the previous method. Thus, there is the advantage to reduce the simulation time, but as a consequence, the results may be unreliable.

- Following a logical approach (Algorithm): differently from the previous cases, the approach distributes CFs in PCC where there is a higher risk of problems (overload or under voltage). Therefore, an algorithm that classified PCC is introduced. This classification is done by considering the distance of the PCC from the main busbar, the annual energy consumption, and the power peak of the customer connected to the PCC.

Additional improvements are the possibility of changing the time step and timeframe of simulation and selecting the number of repetitions by holding the same distribution of CF.

Fig. 5 illustrates the diagram of EVLPG functionality on the top, CF's definition, and the calculation of load profiles on the bottom. The first step involves the definition of the total number of distributions to investigate. This number is strictly dependent on the methodology of distribution that the user defines and the size of the grid. For instance, if the number of potential PCC is ten, the number of CF is one, and the chosen approach of distribution is "All combinations", the number of distributions will be ten. Secondarily, CF are defined. This definition includes the allocations of CF, the number of customers (or charging stations) per CF, and the power level for each charging station.

Once defined the configuration of distribution of CF, load profiles for each CF are calculated. Afterwards, load flow calculation over a defined timeframe is performed, and results are converted to suitable variables. The whole simulation can be repeated n times (Loop number of repetitions) where CF's configuration stays fixed while load profiles change. This approach of simulation is then applied for each possible distribution (Loop number of CF distributions). The process to calculate load profiles is the following: we admit that the timeframe is based on n days, and the integration involves a total of m CF. The algorithm calculates the load profile over one day (from 00:00 to 23:59). It starts with the first CF and calculates the daily load profile starting from day one. Once this daily load profile is defined, it moves forward to the second day, and another daily load profile is calculated. This step is repeated up to the day number n. These load profiles are merged by forming the complete load profile from day one to day n of the first CF. Afterwards, the load profile of the second CF is calculated in the same way. This approach of calculation is applied up to the CF number m. These load profiles are summarised (Fig. 6) and export as a text file.

4) *Power Flow Calculations*

Power flow calculations are performed in OpenDSS and are directly managed through the script "*Master.py*". This communication is possible through a library named "*win32com.client*", which access the OpenDSS COM module. All the files that define the grid model and load profiles created in previous steps are imported in OpenDSS, and time-series simulation is performed by applying the current injection method.

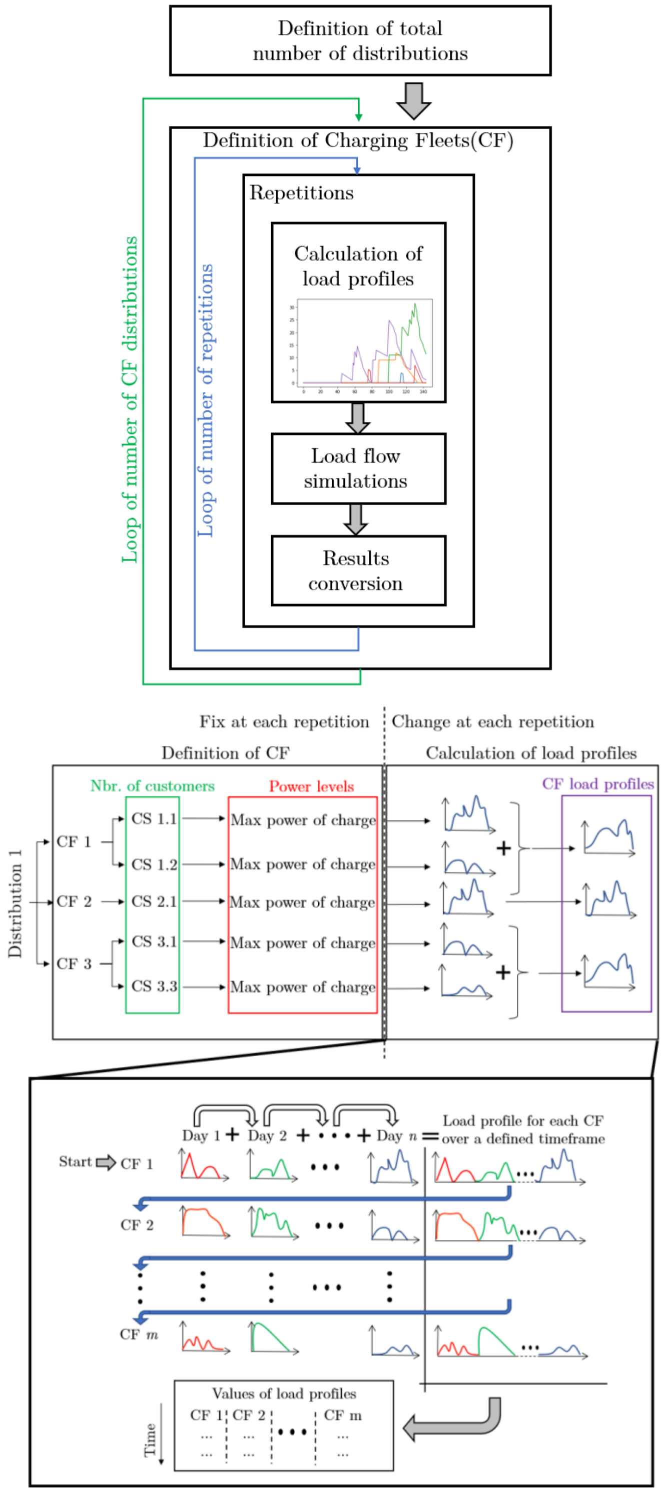

Fig. 5. Structure of the functionality of the EVLPG (Top). Definition of one distribution (above). The process to calculate load profiles (below).

5) *Results Conversion*

OpenDSS provides results for each element where a monitor is installed (lines, PCC, and transformers). The case of lines and transformers provides the current (in Amps) behaviour over time for each phase (L1, L2, and L3). While regarding the PCC, it provides voltage values (in Volt) over time for each phase.

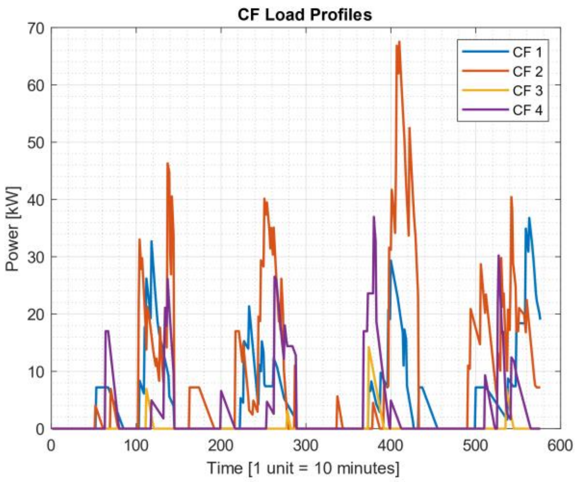

Fig. 6. The behaviour of four CF over four days

This OSM processes the results directly in the "*Master.py*" script through the additional module "*Results_Conversion_Functions.py*". This process consists of converting from current values to loading in percentage, and from the voltage in Volt to voltage expressed per unit (p.u.). These results are then exported as CSV files. This procedure is actuated after each simulation. Furthermore, line loading, loading of the transformer, and PCC voltage are exported in different files. Let's assume a situation of *m* different combinations, and each combination is repeated *n* times. For each variable topology, a total of n(m files will be saved as presented in Fig. 7.

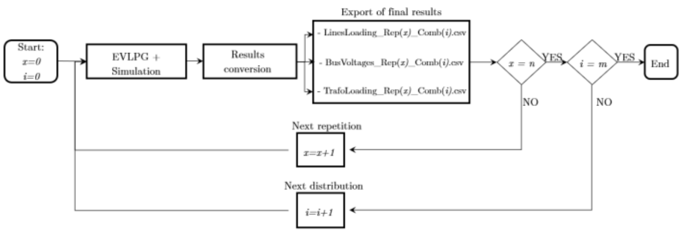

Fig. 7. The diagram shows the approach used to save results.

Moreover, the OSM provides three options to save results:

- Automatically: the process of conversion is actuated, and final results are saved. Once finished with this process, results provided directly by OpenDSS are deleted.
- Manually: The conversion is not applied. Only results export from OpenDSS are available.
- Mix: The conversion is applied, and results provided by OpenDSS are not eliminated.

6) Graphical User Interface

Graphical User Interface (GUI) has been implemented, where the user can manage the tool and define all input data without needing to go inside the program for setting or changing variables. The GUI is managed by the python script "*GUI.py*". Five main sectors can be distinguished, as shown in Fig. 8.

1. External files are defined from here. The user should select the file that contains a list with all elements in the grid and their parameters, the list that contains the SLPs or smart meter data, the OpenDSS script, and the folder where results will be saved.
2. In this part, time-series characteristics are defined. In particular: the timeframe of simulation in days, the time step in minutes, and the number of repetitions that define how many times the simulation – by keeping the same distribution of CF – should be repeated.
3. Define if results should be converted automatically by the module, manually by the user, or a mix of these options.
4. This part defines the methods of distribution of CF.
5. Once defined all parameters, the simulation can start by pressing "*Simulate*". It is also possible to simulate the grid without CF to have a reference situation of the grid.

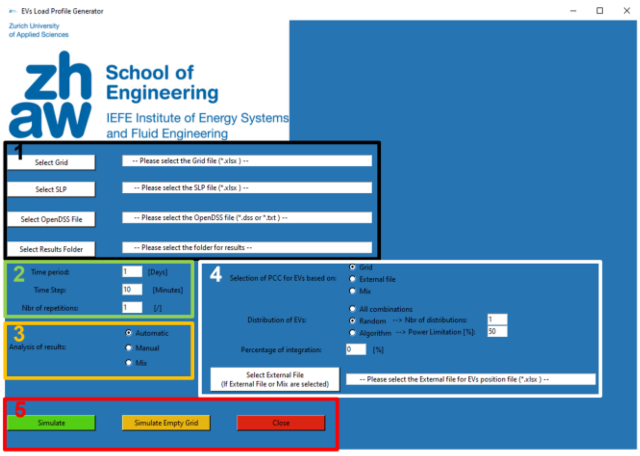

Fig. 8. Graphical User Interface.

D. *Algorithm for Simplifications*

CF could be distributed in different places across the grid. Thus, the term "combination" is introduced. This number of possible combinations depends on the size of the grid and the desired number of CF, and it can be estimated by means of equation [17]:

$$C_r^n = \frac{n!}{r!(n-r)!} \tag{1}$$

Where: C_r^n is the number of combinations, n is a set of items (in this case: potential PCC). r is a selected number of items from the set *n* (in this case: the number of CF).

Fig. 9 illustrates a comparison of the total number of combinations according to the percentage of integration for three different grids (24, 38, and 63 PCC). The number of combinations, according to the size of the grid, could achieve large values. For instance, considering the case of the 63 PCC grid and a percentage of integration of 50% (≈ 32 PCC), the number of possible distributions is calculated by applying equation (1):

$$C_{32}^{63} = \frac{63!}{32!(63-32)!} = 9.16 \cdot 10^{17}$$

The OSM employs approximately ten seconds (the time of one simulation is affected by several parameters. Thus, in the certain case could be higher or lower.) to simulate one combination. Therefore, the time needed to simulate would be around $1.06 \cdot 10^{14}$ days. This example shows that simulate all possible combinations will require an unsustainable time of the simulation. Hence, simplifications are required.

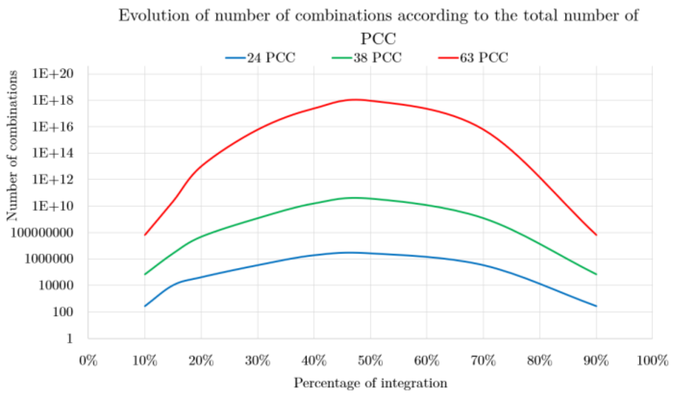

Fig. 9. Combinations for percentages of integration for three different grids.

A simple algorithm that allows reducing the number of combinations is introduced to decrease significantly the time needed to simulate and still achieve plausible results. This algorithm is used to distribute CF at strategical points where problems of overloading or undervoltage could arise. The algorithm works in three steps. The first step allows realising the first reduction of potential PCC by excluding PCC according to their maximal power. The ratio between the maximal power of the load and the nominal power admissible on the line connected to the specific PCC is calculated. This ratio is calculated by applying equation (2).

$$\text{Ratio [\%]} = \frac{P_{PCC\_i}}{\sqrt{3} \cdot U_i \cdot I_i} \cdot 100 \qquad (2)$$

Where: P_{PCC_i}: is the maximal power at the PCC *i*. U_i: is the line three-phases voltage (400 V). I_i: is the nominal current capacity of the line or fuse connected to the PCC *i*.

In the GUI, next to the option of the algorithm, there is a variable named "*Power Limitation [%]*". This variable allows defining the threshold. PCC with a ratio higher than the value defined by the user is not considered. This first reduction is useful for grids where restrictions defined by the distribution system operator occur.

During the second step, PCCs are ordered according to their influence on the conditions of the grid. Classification of PCC is realised by considering the annual energy consumption and the distance of the PCC from the substation. Overloading of lines is an effect mainly caused by high energy consumption. In contrast, a decrease in voltage is related to the energy consumption and the distance from the main substation or busbar.

Assumption: since SLP are used to model existing loads. The behaviours of these loads are similar (according to their specific class). Thus, high energy consumption is related to a higher peak of power. Where in reality it is not always the case.

However, distance and energy consumption have a different degree of influence on the conditions of the grid. A study on a three nodes grid has been realised. This analysis allowed investigating the influence of energy consumption and distance on the loading of line and the voltage at the PCC.

Results are presented as surface plots below. Regarding the voltage (Fig. 10), the distance will play a role starting from around 1000 m and considering an energy demand higher than 85 MWh. By increasing the distance up to 2000 m, problems occur starting from annual energy consumption of 30 MWh. Note that a typical load of class H0 (households) has an annual energy consumption lower than 30 – 40 MWh.

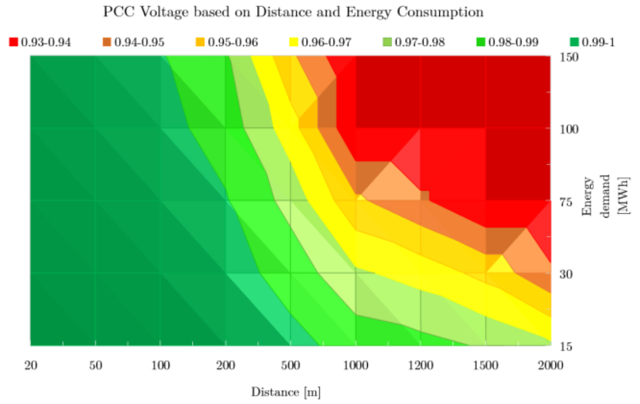

Fig. 10. Voltage according to the distance of the PCC from the substation and the energy consumption.

Regarding line loading (Fig. 11), one can notice that the evolution of loading does not change if distance increases (by keeping the same value of energy demand). Therefore, the line loading seems to be completely independent of the distance.

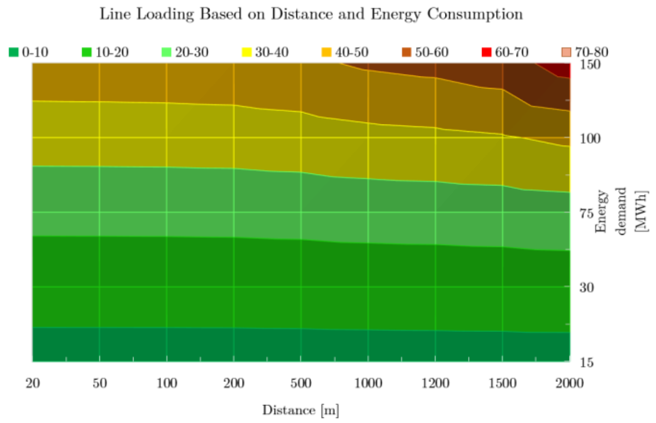

Fig. 11. Line loading according to the distance of the PCC from the substation and the energy consumption.

Energy demand has a bigger impact on the grid compared to the distance. Therefore, the energy should be prioritised, but also the distance should be considered when it becomes important (> 1500/2000 [m]). In according with the results described above, energy consumption and distance are involved in the classification of potential PCC as follow:

$$\text{FC} = 0.99 \cdot E_i + 0.01 \cdot d_i \qquad (3)$$

Where: FC: is the Factor of Classification; E_i: is the energy consumption at the PCC$_i$ in [MWh/y]; d_i: is the distance of the PCC$_i$ from the main substation in [m].

Thus, the FC is composed of 99% from the energy demand and 1% from a distance. This weight applied to both variables allows for prioritising most energy consumption loads. However, the distance will play a role in the case where energy consumptions are similar and when distances achieve important values (> 2000 [m]). By logic, the PCC with the highest FC would be prioritised than a PCC with a lower FC.

The last step defines a reduced number of potential PCC according to the classification. Therefore, a new number of possible combinations is calculated, and simulations are performed by considering all these new combinations.

Note that this is a simple study focused on a specific grid. Thus, the influence of these parameters could change since this influence depends on the topology of the grid. More precisely, studies are needed to properly define the impact of the energy demand compared to the distances. One solution could be to define a dynamic algorithm that changes the weights attributed to energy demand and distance according to the grid's topology.

E. Example of Application

Let us consider a grid with six potential PCC (TABLE II.), and like to study the impact by installing three CF. According to these conditions, the total number of combinations is twenty.

TABLE II. LIST OF POTENTIAL PCC AND THEIR RELATIVE PARAMETERS FOR THE APPLICATION OF THE ALGORITHM.

Load	Ratio	Distance [m]	Energy [MWh/y]	FC
PCC 1	15	120	10	11.1
PCC 2	45	25	30	29.95
PCC 3	40	100	25	25.75
PCC 4	85	15	100	99.15
PCC 5	75	5	50	49.55
PCC 6	50	30	33	32.97

By introducing the algorithm and considering a ratio limitation of 80%, PCC 4 is not considered since its ratio exceeds the limit value. Potential PCC are classified as presented in TABLE III.

TABLE III. LIST OF PCC BY APPLYING THE ALGORITHM.

Load	Ratio	Distance [m]	Energy [MWh/y]	FC
PCC 5	75	5	50	49.55
PCC 6	50	30	33	32.97
PCC 2	45	25	30	29.95
PCC 3	40	100	25	25.75
PCC 1	15	120	10	11.1

Only the classification of these potential PCC does not reduce the number of combinations. Besides, it should be defined an optimised number of effective PCC to consider. This particular situation offers three cases:

- Amount of PCC equal to the number of CF to install (in this case is equal to three). This selection will drastically reduce the number of combinations (from 20 to 1), but the results could be overestimated for a specific case.
- Amount of PCC equal to the number of CF to install plus one (CF+1). In this case, the number of combinations is four.
- Amount of PCC equal to the number of CF to install plus two (CF+2), for a total of ten combinations.

In cases of complex grids such as the SW extensive grid, more possibilities appear. This selection should be made to achieve similar results compared to the initial situation with all PCC as quickly as possible. Fig. 12 illustrates a comparison of a number of combinations as a function of the number of CF installed. Moreover, this comparison is realised for five cases of potential PCC.

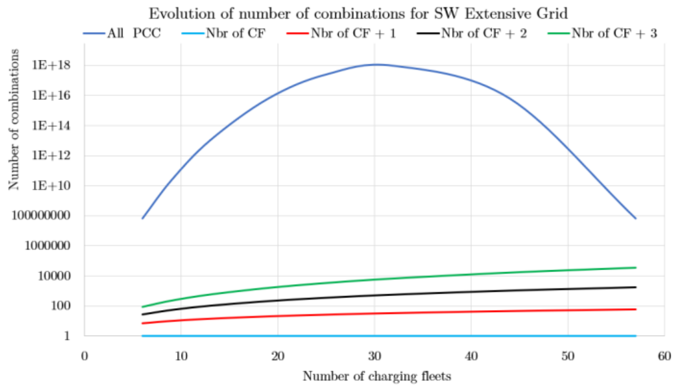

Fig. 12. Combinations according to the number of CF for five cases. SW case.

As illustrated in Fig. 12, the algorithm allows significantly decreasing the number of combinations. However, the optimised number of effective PCC to consider should be defined empirically by means of simulations.

III. POWER SYSTEM MODELS AND ASSUMPTIONS

The following section provides information concerning the models of power grids used to test and verify the OSM's functionality and empirically define the optimised number of effective PCC.

A. Simplified Grid

Initially, a simplified grid with ten PCC (Fig. 13) has been defined and used to test the algorithm and to analyse the optimal number of effective PCC. The simplified grid allows performing a large number of simulations in a reduced time compared to the extensive grid. Therefore, a more in-depth study on the performance of the algorithm is possible.

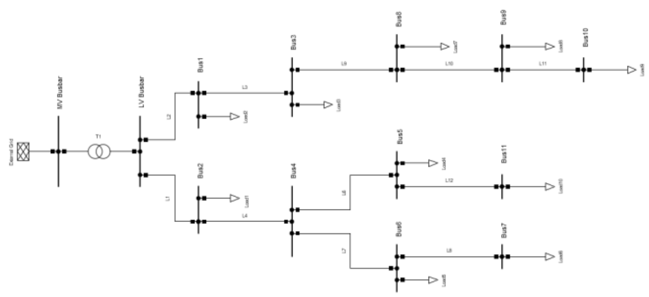

Fig. 13. Simplified grid (10 PCC).

To perform time-series simulations, loads are associated with SLP according to their class, as shown in TABLE IV.

TABLE IV. SIMPLIFIED GRID INFORMATION OF TOTAL PCC.

Consumer class	SLP	Number of PCC	Number of customers
Residential	H0	9	9
Commercial	G0	1	1

B. Stadtwerk Winterthur Extensive Grid

A simplification of the SW Extensive Grid [8] (Fig. 14) has been implemented to study the algorithm's behaviour on a more complex grid than the previous one. Moreover, this grid has been employed to verify the functionality and performance of the OSM.

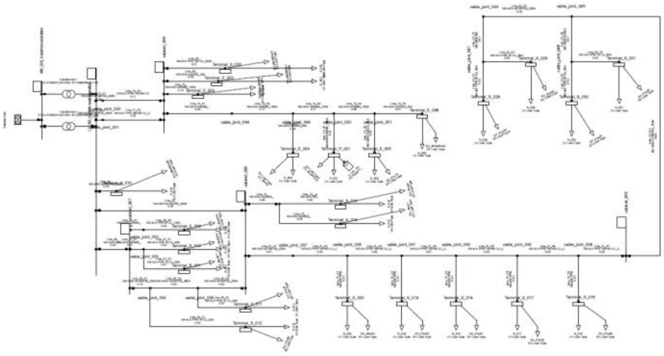

Fig. 14. Simplified Stadtwerk Winterthur Extensive Grid.

This grid includes 24 loads (or PCC). According to their class, these loads were assigned to their respective standard load profile as depicted in TABLE V.

TABLE V. SW EXTENSIVE GRID INFORMATION OF TOTAL PCC.

Consumer class	**SLP**	**Number of PCC**	**Number of customers**
Residential	H0	17	142
Commercial	G0/G2/G6	6	49
Agriculture	L0	1	1

Note that one PCC could represent more than one customer (e.g. block of flats).

C. Initial conditions and assumptions of simulations

There are no standards that establish how to study the integration of EVs in a LVDG. Neither is related to the analysis of results achieved in simulations. In this work, the percentage of integration defines the number of CF. The reference is the total number of PCC in the grid. However, a CF could be composed of several charging stations according to the number of customers connected to the same PCC. This decision derives from the fact that the conditions at the PCC are provided while details regarding the specific customer are unknown. Moreover, considering the percentage of integration based on the number of customers will significantly increase the number of possible combinations.

The variables analysed are line loading and voltage magnitude at the PCC. Results presented in the next chapter are extracted by considering the behaviour of the whole grid. Furthermore, the percentage of times when lines are overloaded, and PCC are subjected to undervoltage situations are provided. This value of percentage is calculated by considering all lines or PCC within the grid. More precisely, by computing the number of points where the limitation is exceeded, afterwards, divide it by the total number of points of calculation. For instance, let us assume a grid composed of m lines, where the behaviour of lines is calculated over a total of n values. The percentage of time where lines are overloaded (loading > 100%) is calculated as illustrated in Fig. 15.

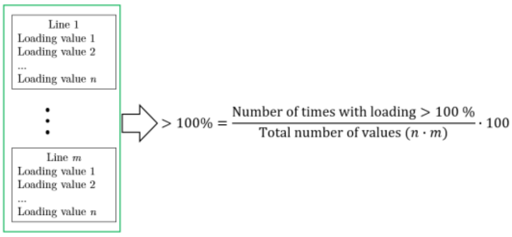

Fig. 15. Percentage of times where lines are overloaded.

The same approach of calculation is applied to the voltage at the PCC. Simulations are yearly based, but the simplification to consider one year as 24 days (two days per month) instead of 365 days has been adopted as shown in EVLPG work [10].

IV. RESULTS

The following section provides, initially, the empirical analysis regarding the application of the algorithm. In parallel, a study related to an optimal compromise of the number of repetitions and precision of results is presented. Moreover, the limitations of the OSM are explained.

A. Application of the Algorithm for Simplifications

1) Number of Repetitions

Repeat the simulations multiple times will provide more generical results by covering a wider range of possible load profiles. While, if the number of repetitions is not sufficiently high, the results are applicable only for specific cases. In order to investigate all possible scenarios, an infinite number of simulations (or repetitions) should be performed. Therefore, the time required to accomplish these simulations will be incredibly high (days or even weeks). Since simulations are based on a statistical approach, it is impossible to obtain similar results from two different simulations. Therefore, simplifications can be applied by accepting results with a certain margin of error.

Simulations by changing the number of repetitions have been performed in the simplified 10 PCC grid. The method used to realise these simulations were "All combinations". Results of line loading for 20% and 40% integration are presented in TABLE VI. and TABLE VII. , respectively. These results include the overall average loading of the whole grid, the maximal value of loading achieved, the percentage of time where lines are overloaded, and the simulation time.

TABLE VI. LINE LOADING RESULTS FOR 20% INTEGRATION.

No. of repetitions	No. of simulations	Average loading	Maximal loading	% > 70%	% > 100%	Time
10	450	12.34	229.15	3.75	1.37	30 min
40	1'800	12.36	250.90	4.08	1.50	2h 10min
60	2'700	12.37	230.69	4.09	1.50	3h 15min
100	4'500	12.37	243.99	4.09	1.51	5h 20min

The case with 100 repetitions is considered as a reference since the highest number of simulations forms it. According to the results presented in TABLE VI. , differences in average loading are lower than 1%. While, concerning the percentage of time where lines are overloaded, a variation of approximately 8% occurs in the case of ten repetitions. The time of simulations is drastically reduced by passing from a total time of 5 h 20 min to 30 min (11 times lower).

TABLE VII. LINE LOADING RESULTS WITH FOR 40% INTEGRATION.

No. of repetitions	No. of simulations	Average loading	Maximal loading	% > 70%	% > 100%	Time
10	2'100	13.32	290.17	4.20	1.69	2h 28m
40	8'400	13.31	289.56	4.55	1.85	12h 50m
60	12'600	13.31	287.87	4.56	1.85	19h
100	21'000	13.31	305.34	4.55	1.86	43h 50m

Results presented in TABLE VII. demonstrate the same behaviour as the previous case. Errors converge to 0% except for the case of ten repetitions. In this case, the percentage of time where lines are overloaded achieves an error of approximately 8-9%. However, the simulations time is subjected to a more significant reduction, passing from a total time of 43 h 50 min to 2 h 28 min (18 times lower). Nevertheless, due to the wider range of parameters and variables that define simulations, it is easy that divergences from the results occur. Therefore, a certain degree of error, even if limited, can be accepted. Hence, it is unnecessary to define many repetitions (> 100 or 60 repetitions). Already ten repetitions seem to be enough to provide results in the same range as the other cases by requiring a non-negligible reduction in time. Another aspect to consider is that a large number of simulations will provide a large number of results, thus it should be considered an increase of time for the analysis of results.

B. *Optimised Number of Potential PCC*

A study for the selection of an optimised number of potential PCC, according to the algorithm, has been performed. The results concerning line loading for 20% and 40% integration cases are presented in TABLE VIII. and TABLE IX. These results were derived from simulations based on ten repetitions. Furthermore, they are divided according to the selected number of potential PCC. "All PCC" means that all PCCs are considered as potential points, while "CF" means that the number of potential PCC is equal to the desired number of CF.

TABLE VIII. LINE LOADING WITH 20% INTEGRATION.

Method	No. of combinations	Average loading	Maximal loading	% > 70%	% > 100%	Time
All PCC	45	12.34	229.15	3.75	1.37	30 min
CF	1	11.65	179.60	3.83	1.40	1 min
CF + 1	3	12.14	187.81	4.12	1.60	2 min
CF + 2	6	12.03	199.10	3.97	1.45	4 min
CF + 3	10	11.95	191.65	3.91	1.39	7 min

TABLE IX. LINE LOADING WITH 40% INTEGRATION.

Method	No. of combinations	Average loading	Maximal loading	% > 70%	% > 100%	Time
All PCC	210	13.32	290.17	4.20	1.69	2h 28m
CF	1	12.56	173.93	4.38	1.68	1 min
CF + 1	5	12.50	195.14	4.19	1.64	3 min
CF + 2	15	12.94	231.57	4.48	1.83	12 min
CF + 3	35	13.03	262.04	4.66	1.93	27 min

The goal is to achieve results close to the case where all PCCs are considered in the shortest possible time. Results show a considerable similarity from all cases, especially regarding the percentage of time where lines are overloaded. Fig. 16 illustrates the average absolute errors for each method compared to the reference one, which is the "All PCC" method. These differences are separated according to the percentage of integration (20% and 40%). Moreover, for comparison purposes, the average errors have been calculated without including the variation concerning the maximal loading.

Fig. 16. Average of absolute error of line loading to the reference case.

As expected, the method "CF+3" achieves the closest results in most cases, but errors are generally lower than 10%. Due to the elevated number of parameters that influence simulations' results, there is no net distinction of a better method to use. Therefore, the time of simulation is considered as the priority criterium of selection. Thus, the "CF" method proposes a satisfactory compromise regarding results and time.

The "CF+1" method shows good compromise, but the time required to simulate is influenced by the size of the grid and the percentage of integration. These influences can be seen in TABLE X. , where the time required to simulate in the SW simplified extensive grid (24 PCC) is presented.

TABLE X. TIME OF SIMULATION BY USING THE "CF" AND "CF+1"

Percentage of integration	Method	No. of combinations	Time
40%	CF	1	2 min
	CF + 1	11	23 min
90%	CF	1	2 min
	CF + 1	23	1h 35min

The simulation time (in the case of 40% integration) is approximately ten times higher than the "CF" method. While, in the previous case (TABLE IX.), the difference was a factor of three. Therefore, the CF+1 method can be applied on a relatively small grid (10-15 PCC). These results demonstrate the possibility to decrease the time of simulation by achieving results with a certain margin of error (<10%) by adopting specific simplifications. Simplifications are not mandatory and can be changed according to the user's requirements.

C. *SW Simplified Grid (24 PCC)*

These simplifications have been implemented in the SW simplified grid. In particular, the reduction to ten repetitions and the deployment of the "CF" and "CF+1" methods. This grid has been simulated by considering a percentage of integration of 10% (TABLE XI.) and 90% (TABLE XII.). Line loading results are summarised in the tables below.

TABLE XI. LINE LOADING RESULTS WITH 10% INTEGRATION.

Method	No. of combinations	Average loading	Maximal loading	% > 70%	% > 100%	Time
All PCC	276	7.43	188.59	0.028	0.0025	10 h 15m
CF	1	7.33	55.11	0	0	2 m
CF + 1	3	7.79	142.98	0.054	0.0045	6 m

TABLE XII. LINE LOADING WITH 90% INTEGRATION.

Method	No. of combinations	Average loading	Maximal loading	% > 70%	% > 100%	Time
All PCC	276	11.09	217.99	0.221	0.026	12 h
CF	1	11.78	131.99	0.101	0.005	2 m
CF + 1	23	11.21	192.14	0.262	0.034	1h 35m

Results of average loading show that variations are around 1-5% as achieved in the previous case. In contrast, there are huge differences concerning the percentage of times where lines are overloaded. However, there is a tendency to reduce the error by increasing the number of potential PCC. The main cause of these differences is certainly linked to the behaviour of EV load profiles. In certain cases, load profiles are concentrated in a specific period over the day, and this concentration leads to a higher maximal loading value, but lines are stressed over a shorter period. On the other hand, load profiles could be distributed over the day, and hence lines are subjected to lower stress but for a longer period. These results demonstrate that an extensive grid is more sensitive to the introduction of simplifications, and additional in-depth studies regarding the limits of the simplifications should be investigated.

V. Conclusion and Outlook

A. Conclusion

This paper aims to provide an overview of developed a versatile open-source module focused on simulating the integration of EVs in a LVDG. This module is based on Python's scripting language, which allows calculating EV load profiles and distributing them across a defined grid. Time-series simulations are executed using secondary software to evaluate the behaviour of the power grid. Three open-source power analysis tools: Pandapower, PyPSA, and OpenDSS, have been compared to replace the commercial tool PowerFactory. Results have demonstrated that all three software achieve similar results (average error < 2%) compared to PowerFatory. However, OpenDSS is characterised by a high level of performance by solving simulations approximately seven times faster than PF. Thus, OpenDSS has been chosen as alternative software to integrate with the OSM.

The distribution of EV load profiles across the grid is defined automatically in the OSM. To obtain an impact on LVDG, all possible combinations of CF distribution should be simulated. Simulate all distributions requires a significant amount of time, considering the number of combinations and the complexity of the grid. Therefore, the OSM proposes a simple algorithm that allows reducing this number of distributions. This algorithm provides PCC classification based on their annual energy consumption and their distance from the substation. Only a reduced number of PCC are considered as potential PCC instead of all existing PCC. An additional aspect that affects the time of simulation is the number of repetitions per simulation. Results have demonstrated that in a simple grid (10 PCC), significative simplifications are admissible. In terms of the number of repetitions, already ten repetitions seem to be enough to provide valid results by requiring a non-negligible reduction in time: from several hours to few hours or even minutes. The number of potential PCC can be minimised, and simulations still provide results with a margin of error lower than 10 %. However, the behaviour on an extensive grid (24 PCC) is entirely different. Following the results, this grid is more sensitive to the introduction of simplifications, and results indicate variations between 20% and 100% compared to the reference simulation. Therefore, an additional in-depth study regarding the limits of simplifications should be realised.

In addition, these analyses have revealed a weakness of the OSM related to the iterative method used to perform time-series simulations. This module can be applied exclusively in radial grids. In certain conditions, the utilisation module is limited due to the high demand for Random-Access Memory, and therefore module allows integrating a maximum of 30 CF.

B. Outlook

The first version of the OSM can be already implemented to estimate EVs' impact on an LVDG. This simulation module can become a powerful tool in the field of grid analysis. However, as for all kinds of software, improvements and upgrades are necessary. Therefore, consecutive works –to enhance the actual module – could be proposed. Few possible improvements are highlighted:

- Define simplifications limits for extensive grids.
- Implement unbalanced power flow calculation to integrate one-phase and two-phase chargers.
- Investigation on a dynamic algorithm for the selection of potential PCC able to change the weights attributed to energy demand, and distance according to the topology of the grid.
- Investigation of the benefits via parallel computation techniques for performance enhancements.

There are always improvements to be implemented, and most of them depend strictly on the users' personal experiences. Therefore, the OSM is available in GitHub at the following link [15], and it can be used and modified.

References

[1] H. Shaftel, R. Jackson, S. Callery and D. Bailey, "Overview: Weather, Global Warming and Climate Changes," NASA's Jet Propulsion Laboratory, [Online]. Available: https://climate.nasa.gov/resources/global-warming-vs-climate-change/.

[2] Federal Office for the Environment, Federal Office for the Environment, [Online]. Available: https://www.bafu.admin.ch/bafu/en/home.html.

[3] United Nations, "Paris Agreement," in Framework Convention on Climate Change (2015) Adoption of the Paris Agreement, Paris, France, 2015.

[4] S. Wappelhorst and H. Cui, "Growing momentum: Global overview of government targets for phasing out sales of new internal combustion engine vehicles," The International Council on Clean Transportation, November 2020. [Online]. Available: https://theicct.org/blog/staff/global-ice-phaseout-nov2020.

[5] Swiss Federal Government, "Ordinance on the Reduction of CO2 Emission," Fedlex: The publication platform for federal law, 2012. [Online]. Available: https://www.fedlex.admin.ch/eli/cc/2012/856/en#chap_1.

[6] "How to stop global warming ?," Solar Impulse Foundation, [Online]. Available: https://solarimpulse.com/global-warming-solutions#.

[7] S. a. Abbas, X. Ai, A. Masood and S. Iqbal, "Electric Vehicles and their Impacts on Integration into Power Grid: A Review," in *IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, China, 2018.

[8] C. Allenspach, R. Högger, A. Obushevs, P. Korba, H. Reist, "Implementation of Quasi-Static Time Series Simulations for Analysis of the Impact of Electric Vehicles on the Grid" *IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University*, 2019.

[9] Y. Yu, A. Shekhar, G. C. R. Mouli, P. Bauer, N. Refa, and R. Bernards, "Impact of uncontrolled charging with mass deployment of electric vehicles on low voltage distribution networks," *IEEE Transportation Electrification Conference Expo (ITEC)*, 2020.

[10] L.Giorgi, A.Obushevs, P.Korba, "Electric Vehicles Load Profile Generator Based on the Probability Density Functions," *IEEE 62th International Scientific Conference on Power and Electrical Engineering of Riga Technical University*, 2021.

[11] "Pandapower: open source tool for power system modeling, analysis and optimisation with a high degree of automation." [Online]. Available: http://www.pandapower.org/.

[12] "PyPSA: Python for Power System Analysis," [Online]. Available: https://pypsa.org/.

[13] "OpenDSS: electric power Distribution System Simulator " [Online]. Available: https://sourceforge.net/projects/electricdss/.

[14] DigSilent, Power System Solutions, "PowerFactory," [Online]. Available: https://www.digsilent.de/en/powerfactory.html.

[15] L. Giorgi, A. Obushevs, "Electric Vehicle Load Profiles Generator for LV grid studies" [Online]. Available: https://github.com/A-Obushevs/EVLPG

[16] MathWorks, "MATLAB," [Online]. Available: https://ch.mathworks.com/products/matlab.html.

[17] M. J. Evans and J. S. Rosenthal, "Chapter1.4: Uniform Probability on Finite Spaces," in *Probability and Statistics - The Science of Uncertainity, Second Edition*, Toronto, 2010, pp. 14-20.

Dr. Artjoms Obushevs received the BSc, MSc, PhD degree in electrical engineering from the Riga Technical University, in 2008, 2010 and 2014 respectively. Since 2018, he is a Research Associate in the Electric Power Systems and Smart Grids group at the Institute of Energy Systems and Fluid Engineering of the Zurich University of Applied Science ZHAW in Winterthur, Switzerland. His main research is focused on methods of mathematical modelling of electrical networks and systems elements; development of power systems planning; dynamic optimisation methods and decision systems. Author and co-author of more than 50 peer-reviewed journal and conference papers in the field of power systems.