Step-by-step Protocol to Quantify the Effect of Distributed Energy Resources on an Electrical Distribution Network

Miiyu Fujita
Electrical and Computer Engineering
McGill University
Montreal, Canada
miiyu.fujita@mail.mcgill.ca

Ralph Younan
Electrical and Computer Engineering
McGill University
Montreal, Canada
ralph.younan@mail.mcgill.ca

***Abstract*—With the need to transition to green energy, distributed energy resources (DER) such as rooftop solar photovoltaic generation, energy storage systems, and controllable loads like electric vehicle chargers will gradually be implemented in electric networks. With this change, the effect of DERs need to be quantified to evaluate how distribution networks can accommodate these additions without additional modifications or upgrades. In this paper, a step-by-step protocol that aims to quantify these effects using software tools is presented. The proposed protocol was then demonstrated on a test network, where the occurrence of voltage violations, overcurrent and overloading, voltage phase unbalance, losses in the networks, and the power flow through the substation transformer were all tracked as indicators of how the network was responding to the increasing integration of DERs.**

***Index Terms*—Distributed Energy Resources, Electricity Distribution Network, Hosting Capacity, GridLAB-D, Python**

I. Introduction

The global transition towards green energy and renewable energy resources has seen advancements in recent years, with annual renewable capacity additions breaking new records in 2021 [1]. However, these increases in deployment of renewable energy resources are not yet enough to reach global net-zero emission goals [2]. Although distributed generation and distributed energy resources do not necessarily imply renewable energy generation, integrating renewable distributed energy resources into the grid can lead to further decarbonization by eliminating costs related to centralized generation [3]. Furthermore, proper coordination of the grid and its resources may lead to even more advantageous outcomes, as it can allow for the harnessing of energy that might have otherwise been wasted [3]. Despite the potential benefits of DER integration into the grid, the integration of new technologies into the network near consumers must not endanger the safety of consumers. Thus, there is a need to quantify the amount of DERs that can be integrated into a network without violating any safety regulations or requiring updates to the network, i.e the hosting capacity of the network. [4] There has already been research done in past years in order to determine the hosting capacity of networks [4, 5] This paper therefore aims not to develop a new method to quantify hosting capacity. Rather, it is focussed on detailing a process of how to go about quantifying hosting capacity in a simulated environment using specific software tools, such that subsequent research that may require hosting capacity analysis, or similar analysis of the effect that DERs can have on the network, may be done with more ease and with more knowledge of the available software tools. This proposed process is the result of extensive research on the appropriate software to use and how to use it. In this way, this protocol can be used to perform simulations to evaluate the impact of new grid technologies when needed in the future.

This paper will first present the proposed protocol and the steps involved in its development in Section 2. The section will also introduce the two software tools used in the protocol, as well as how to use them within the context of future research, in terms of how they can be used to retrieve specific performance indicator values to gauge how DERs are affecting the networks evaluated. The subsequent section, Section 3, will then detail a case study performed on a known IEEE-13 feeder, to show how the proposed protocol may be used in the context of a network evaluation. Any discussion related to the protocol, the case study, or potential improvements will be done in Section 4.

II. Proposed Protocol

In order to develop a step-by-step protocol that could support and facilitate subsequent research that would require quantifying the relationship between DERs and the network, it was important to determine what technology and tools would be used for the analysis, as well as how the effect of DERs on the network could then be quantified using the tools chosen previously.

This research project was made possible through generous contributions from Mr. Ruben Gruber and the Natural Sciences and Engineering Research Council of Canada (NSERC).

A. Software Tools

The two main software tools used in the scope of this research are GridLAB-D and the Python programming language.

1) GridLAB-D: GridLAB-D is an open-source software [6] "power distribution system simulation and analysis tool" that offers "a flexible simulation environment" [7]. For this paper, GridLAB-D has been used as a simulation environment, in which sample electricity distribution networks can be modelled with different placements of PV, and from which different values can be retrieved to observe the effect these PV installations may have on the network. The GridLAB-D software comes with many classes, including ones that can model PV installations, as well as ones that can record and save simulation data into file formats easy to manipulate (ex: csv or JSON).

2) The Python Programming Language: The Python programming language has been used in order to manipulate the aforementioned output files from GridLAB-D when necessary, specifically when data parsing or calculations are required to obtain the desired metrics.

B. Quantifying the Effect of DERs on a network

As previously mentioned, hosting capacity refers to "the amount of new production or consumption [DERs] that can be connected to the grid without endangering the reliability or voltage quality for other customers" [4]. In other words, hosting capacity is a way to quantify the effect that DERs can have on the electricity distribution network by evaluating when certain limits or regulations are violated [5]. To determine the effect of DERs on a network, a protocol similar to a hosting capacity analysis could prove to be useful. It is to note then that, when evaluating hosting capacity, it is an important first step to determine the performance standards, or the limits that, when crossed, indicate that the network has reached its hosting capacity. In a similar vein, by determining performance indicators of a network, it could be easier to visualize and observe the effect that distributed energy resources may have on the network.

The performance standards suggested in this paper are chosen in order to more easily observe the changes in a network caused by the integration of DERs. As the main goal of this paper is to propose a protocol that can be followed using specific software tools, for each of the suggested performance standards, there is a corresponding proposed protocol using the software tools mentioned earlier to follow in order to retrieve the metrics successfully.

1) ANSI voltage standards: The ANSI C84.1 standard has been used in [5] as one of the QSTS PV hosting capacity metrics, in which the metrics were based on common standards. The ANSI C84.1 standard specifies two ranges, A and B. GridLAB-D comes equipped with a class that can detect and retrieve any violations of either range during the simulation, the "metrics collector" class [8]. Depending on what type of object the metrics collector object is attached to, a set of different values are aggregated over a user-specified time period [8]. The collection of aggregated values can then be recorded during the simulation period using a metrics collector writer object, which loops through all the metrics collector objects present in the file being simulated at user-specified intervals and writes the values to a JSON file [9]. The Python programming language is well-suited to manipulate JSON files, and thus can be used to parse through the output JSON file to retrieve the information needed, that being the number of times either of the ranges has been violated during the simulation. These values can be stored in a csv file, which can then be easily visualized if necessary using tools such as google sheets. The python script written for this purpose (parsing of JSON file and writing of values into CSV file format) during the scope of this research can be found here. Tracking and evaluating the occurrence of violations of these two ranges is therefore relatively straight-forward, can be quantified, and may thus offer useful insight as to how the DERs in the network are influencing its surroundings.

Fig. 1. Evaluating the network based on ANSI C84.1 Voltage Standard Violations

2) Overcurrent and Overloading - Thermal Violations: Overcurrent refers to "excessive currents or current beyond the acceptable current rating of equipment," [10] whereas overloading refers to "a running overcurrent" that could then cause the equipment in question to overheat [10]. As the safety of consumers is of concern, tracking the occurrence of the two can be a useful indicator of how the network is being changed by the presence of DERs. This metric is easily observable when simulating the network in GridLAB-D, using the GridLAB-D console. The software itself is equipped with a functionality that prints warning messages to its console as the simulation progresses. The user may then compare the warning messages between simulations with different DER penetration cases to observe the effect that the newly introduced resources are having on the surrounding network.

Fig. 2. Evaluating the network based on overcurrent and overloading

3) Voltage unbalance between phases: Voltage unbalances can occur when there is an unequal distribution of loading over phases, and can be the cause for overheating or overloading of equipment in the network [11]. Therefore, when evaluating the effect of DERs on a network in different scenarios (i.e different levels of DER penetration, DERs loaded on one of the phases more than the others), it is interesting and useful to observe how for different cases, the voltage balance between the three phases is affected.

The voltage unbalance between phases is calculated using Fortescue's transformation for three phases, which determines voltage unbalances using sequence voltages. Sequence voltages can be obtained through the following matrix multiplication:

$$Vs = A^{-1} \cdot Vp$$

Where:

$$Vp = [Va; Vb; Vc]$$

$$A = [111; 1a^2a; 1aa^2]$$

$$a = e^{j2\pi/3}$$

$$Vs = [V0; V1; V2]$$

Upon performing this multiplication, Vs will hold 3 sequence components: a positive sequence component, a negative sequence component, and a zero sequence component (V0 = zero sequence, V1 = positive sequence, V2 = negative sequence). An unbalance has occurred if the ratio of the negative sequence component over the positive sequence component is above 2% [12].

In the GridLAB-D software, there is a group recorder class that can record the values of one property from a group of objects, either all from the same class or with similar properties [13]. To obtain the phase voltages necessary for the voltage unbalance calculations, the user can use three group recorder objects, one for each phase. After the simulation, the user will be able to find 3 csv files, each containing phase voltages for the class of objects specified in the file [13]. These csv files can then be passed through the voltage-unbalance-calculator.py file found here to obtain a dictionary of how many times an unbalance has occurred and where they have occurred in the network. In this way, the occurrence of voltage unbalances during a simulation is quantifiable, can be measured across simulations, and can therefore be an indicator of how DERs and their placement within the network influence its surroundings.

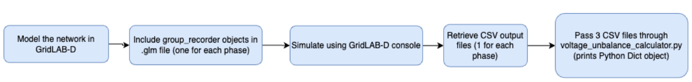

Fig. 3. Evaluating the network based on occurrence of voltage unbalances

4) Losses in the line: Many GridLAB-D classes (specifically the line classes) come with a property to track power loss. The variation of losses across simulations can be an interesting parameter to look at, as it is easily quantifiable, and it may provide insight into the effect of DERs on a network that may not initially be obvious [14]. To measure power loss, users can take advantage of the GridLAB-D group recorders, which, as previously mentioned, can track the values of one property of a specified group of objects from the same class or with similar properties [13]. When measuring losses in the line, line objects come with parameters to track losses in each individual phase as well as the power loss for all three phases together. The user may choose which parameter(s) they would like to track depending on the context of their research.

Fig. 4. Evaluating the network based on losses in the line

5) Flow of active/reactive power in substation transformer: Introducing distributed generation into the network may lead to the occurrence of reverse power flow, which is when the energy generated by the DERs exceeds consumer demand within the network, leading to power flowing in the opposite direction, meaning towards the substation [15]. Tracking the active and reactive power flowing through the substation transformer is possible in GridLAB-D, as the Transformer class inherits all properties of the Link class, including parameters for power flow through all phases and each individual phase [16]. Thus, using a recorder object in GridLAB-D, users wishing to track the power flow at the substation transformer may do so.

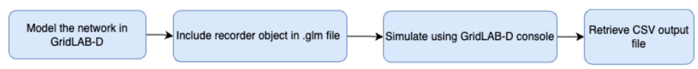

Fig. 5. Evaluating the network based on the flow of active and reactive power in the substation transformer

Evaluating the effect of DERs on a network using these performance indicators in practice may look like the following:

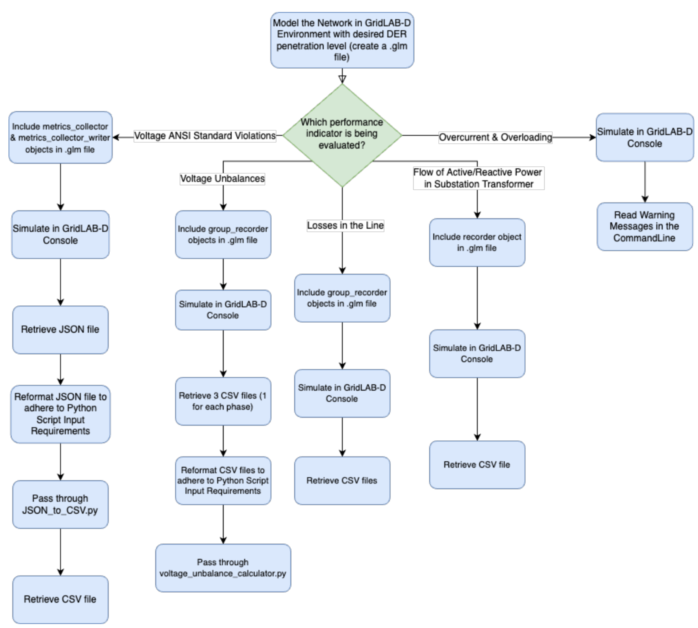

Fig. 6. Evaluating the effect of DERs on a network

Where every time a different case is being simulated (ex: at every level of DER penetration), the above flowchart is followed to obtain the values of the performance indicators.

III. Case Study

A. Introduction

In this section, the steps described in the last section will be applied and a case study is performed on the IEEE-13 test feeder. First, the network in figureXX is modeled in GridLAB-D using the .glm extension. A series of power flow simulations are then performed to obtain several datasets. These datasets are analyzed and, depending on the simulation context, the differences are explicitly pointed out. Results will be displayed using graphs and tables. A conclusion on this specific network will then be drawn from these simulations.

The IEEE-13 test feeder has this topology [17]:

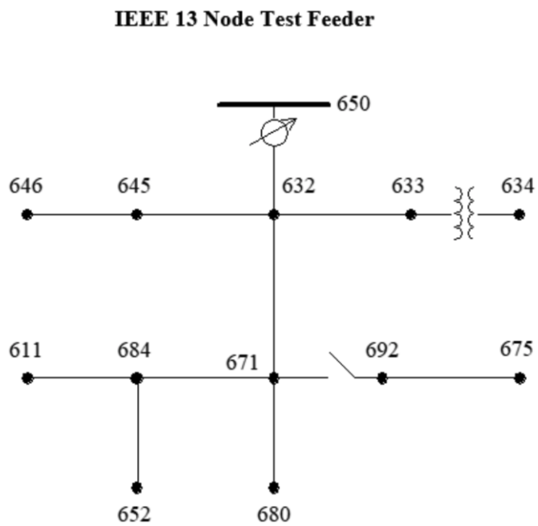

Fig. 7. Network topology used in the case study

B. Modeling

The initial and most important step of the process is to model the network in GridLAB-D. The starting point to writing a simulation program in GridLAB-D is adding all the different modules in the file. Internally, GridLAB-D uses UTC as its clock but has support for presenting time in any time zone. A clock statement which defines the local time in which the simulation will be run; the dates and times of the beginning and ending of the simulation period.

```
module generators;
module tape;
module powerflow{
    solver_method FBS;
    default_maximum_voltage_error 1e-9;
    line_limits False;
};
module climate;
object climate {
    name "Spokane WA";
    tmyfile "Shared Model Files/WA-Spokane-727850.tmy3";
    interpolate LINEAR;
}
module residential {
    implicit_enduses NONE;
    ANSI_voltage_check False;
};
#set profiler=1
#set randomseed=1;

clock {
    timestamp '2022-07-01 05:00:00';
    stoptime '2022-07-01 010:00:00';
    //timezone EST+8EDT;
}
```

Fig. 8. Module statements needed to be able to use certain classes and object during simulations

The next step in the modeling part of the process would be to configure lines. In this case, the test feeder's configuration is already specified.

Overhead Line Configuration Data:

Config.	Phasing	Phase	Neutral	Spacing
		ACSR	ACSR	ID
601	B A C N	556,500 26/7	4/0 6/1	500
602	C A B N	4/0 6/1	4/0 6/1	500
603	C B N	1/0	1/0	505
604	A C N	1/0	1/0	505
605	C N	1/0	1/0	510

Underground Line Configuration Data:

Config.	Phasing	Cable	Neutral	Space ID
606	A B C N	250,000 AA, CN	None	515
607	A N	1/0 AA, TS	1/0 Cu	520

Line Segment Data:

Node A	Node B	Length(ft.)	Config.
632	645	500	603
632	633	500	602
633	634	0	XFM-1
645	646	300	603
650	632	2000	601
684	652	800	607
632	671	2000	601
671	684	300	604
671	680	1000	601
671	692	0	Switch
684	611	300	605
692	675	500	606

Fig. 9. Line configurations for topology chosen [17]

An example of the configuration in GridLAB-D:

```
object line_configuration {
    name lc601;
    conductor_A olc6010;
    conductor_B olc6010;
    conductor_C olc6010;
    conductor_N olc6020;
    spacing ls500601;
}
```

```
// Phase Conductor for 601: 556,500 26/7 ACSR
object overhead_line_conductor {
    name olc6010;
    geometric_mean_radius 0.031300;
    resistance 0.185900;
}
```

Fig. 10. Line configurations written in GridLAB-D

Next, using the topology of the network the nodes and the lines that connect these nodes are modeled as such:

```
// Define line objects
object overhead_line {
    phases "BCN";
    name "ohl_632-645";
    from n632;
    to n645;
    length 500;
    configuration lc603;
}
```

```
// Create node objects with attached loads
object node {
    name n611;
    phases "CN";
    voltage_C -1200.8886+2080.000j;
    nominal_voltage 2401.7771;
}
```

Fig. 11. Nodes and lines in GridLAB-D

It is also necessary to add Meters objects in the network. Since recorders are usually connected to meters, they are helpful to the simulations because they contain parameters that are useful to recorders when running simulations. Meters are usually what is connecting a node to a specific house through lines.

```
// House 1 meter
object triplex_meter {
    name trip_meter1;
    phases CS;
    nominal_voltage 120;
}

// House 2 meter
object triplex_meter {
    name trip_meter2;
    phases AS;
    nominal_voltage 120;
}
```

Fig. 12. Meters in GridLAB-D

The following house object is necessary because we need them to define the power demand in the network.

```
// House 1
object house:..20 {
    parent trip_meter1;
    floor_area random.normal(1750,400);
    heating_setpoint heatingsetpoint4+random.normal(0,4);
    cooling_setpoint random.normal (80,2);
    thermostat_deadband random.normal (2,.5);
    object ZIPload {
        base_power ZIP1base_power*random.normal(1,.25);
        power_fraction 0.2;
        impedance_fraction 0.3;
        current_fraction 0.5;
        power_pf 0.9;
        current_pf 0.6;
        impedance_pf 0.8;
    };
```

Fig. 13. House object in GridLAB-D

Another important part to model for these simulations is the inverter. Since a solar PV installation is inherently DC and the electrical grid is AC, an inverter is needed to connect the two.

```
object inverter {
    name solar_inv8;
    phases CS;
    parent solar_meter8;
    generator_status ONLINE;
    inverter_type FOUR_QUADRANT;
   //four_quadrant_control_mode CONSTANT_PF;
    generator_mode SUPPLY_DRIVEN;
    inverter_efficiency .95;
    rated_power 6000;
}
```

Fig. 14. Inverter in GridLAB-D

Moreover, the last object to model for simulation purposes is the solar panel. This is the DER we are evaluating in this case study.

```
object solar {
    name solar_8;
    phases CS;
    parent solar_inv8;
    //generator_status ONLINE;
    //generator_mode SUPPLY_DRIVEN;
    panel_type SINGLE_CRYSTAL_SILICON;
    area 250 ft^2;
    tilt_angle 47.0;
    efficiency 0.135;
    orientation_azimuth 180; //equator-facing (South)
    orientation FIXED_AXIS;
    SOLAR_TILT_MODEL SOLPOS;
    SOLAR_POWER_MODEL FLATPLATE;
}
```

Fig. 15. Solar panel in GridLAB-D

The particularity in GridLAB-D is that, in order to have results, we have to include recorders in the .glm file. They can either be declared as their own object explicitly parented to the object with data of interest by adding a parent ...; statement to their definition or they can be implicitly parented by including their definition inside the definition of the object of interest. All that is generally required is a definition for the file to write the data out to, how often the values should be measured, the parameter of interest, and (if making an explicit recorder) the parent object.

Once the GridLAB-D model is written, the simulations can be performed.

```
object multi_recorder {
    property trip_meter1:measured_real_power,
    file "IEEE_13_Node_With_Houses.csv";
    interval 60;
    limit 60;
}
object multi_recorder {
    property ohl_684-611:power_losses;
    file "losses.csv";
    interval 60;
    limit 60;
}
```

Fig. 16. Recorders in GridLAB-D

C. Simulations

The next major step in the protocol described in the paper is to simulate our GridLAB-D model. Assuming there is no error in the modeling part, the simulation part should be straightforward. To be able to analyze how penetration of solar panels affects distribution networks, a gradual increase of solar panels in the network was simulated. Starting from no solar panel (0%) to solar panels on every house object (100%) with 15% increments. We will then track these 5 parameters:

1) ANSI Voltages: For the ANSI voltage standards, it is important to add a metrics collector object in the model. As shown in the figure below, this object is connected to the meters of each house.

```
// House 1 meter
object triplex_meter {
    name trip_meter1;

    phases CS;
    nominal_voltage 120;
      object metrics_collector {
          interval 300;
     };

}
```

Fig. 17. Metrics collector object attached to meter in GridLAB-D

For the data acquisition to work correctly, it is needed to add a metrics collector writer as it is demonstrated below. This is used to write the data collected in a JSON file that can be used in the python script mentioned in Section 2.

```
object metrics_collector_writer {
    interval 300;
    filename metrics_collector_output1.json;
}
```

Fig. 18. Metrics collector writer object in GridLAB-D

For this specific network, there were not many issues when adding solar panels to the network. To demonstrate that, below is shown the data that counts how many times the voltage range A and range B were violated. First we present the base case and the case in which the addition of solar panels perturbed the network. We can see that there were only 3 below rangeA violations and 3 below rangeB violations (assuming the exponential values are deemed as outliers).

Results were similar for every step from 0% to 85%, which corresponds to the case with 13 houses with solar panels. In

```
meter,above_rangeA,below_rangeA,above_rangeB,below_rangeB
trip_meter1,0.0,5.2e-322,0.423208731781238,0.32279401143406605
trip_meter10,0.0,0.0,0.0,0.0
trip_meter11,0.0,0.0,0.0,0.0
trip_meter12,0.0,0.0,0.0,3.424e-321
trip_meter13,0.0,0.0,0.0,0.0
trip_meter14,0.0,0.0,0.0,0.0
trip_meter15,0.0,0.0,0.0,0.0
trip_meter2,0.0,1.0,0.0,1.0
trip_meter3,0.0,1.0,0.0,1.0
trip_meter4,0.0,1.0,0.0,1.0
trip_meter5,0.0,0.0,0.0,0.0
trip_meter6,0.0,0.0,0.0,0.0
trip_meter7,0.0,0.0,0.0,0.0
trip_meter8,0.0,0.0,0.0,6.53e-321
trip_meter9,0.0,0.0,0.0,0.0
```

Fig. 19. ANSI voltage standard violations for the base case (0% solar panel penetration level)

the figure below, it is possible to observe that there are more violations but again, not many.

```
meter,above_rangeA,below_rangeA,above_rangeB,below_rangeB
trip_meter1,0.0,5.2e-322,0.4284283,0.0,0.0
trip_meter10,0.0,2.0,0.0,0.0,0.0
trip_meter11,0.0,0.0,0.0,0.0,0.0
trip_meter12,1.0,0.0,0.0,0.0,3.98e-312
trip_meter13,1.0,0.0,0.0,0.0,0.0
trip_meter14,0.0,0.0,0.0,0.0,0.0
trip_meter15,1.0,0.0,0.0,1.0,0.0
trip_meter2,0.0,0.0,0.0,0.0,0.0
trip_meter3,0.0,0.0,0.0,0.0,0.0
trip_meter4,0.0,0.0,2.0,0.0,1.0
trip_meter5,0.0,0.0,2.0,0.0,1.0
trip_meter6,0.0,0.0,2.0,0.0,2.0
trip_meter7,0.0,0.0,0.0,0.0,0.0
trip_meter8,2.0,0.0,0.0,1.0,6.27e-312
trip_meter9,1.0,0.0,0.0,1.0,0.0
```

Fig. 20. ANSI voltage standard violations for the base case (85% solar panel penetration level)

For this particular network the effect of adding solar panels do not affect the number of ANSI voltage standards violations. Perhaps this is due to the network being relatively small in size, which motivates testing the protocol on larger networks.

2) Overcurrents and Overloading: For overcurrents and overloading, no additional tool is needed; it is possible to observe this on the command line. Warnings will appear if line ratings are exceeded. There were no apparent violations for the base case as can be seen in the figure below.

```
meter,above_rangeA,below_rangeA,above_rangeB,below_rangeB
trip_meter1,0.0,5.2e-322,0.4284283,0.0,0.0
trip_meter10,0.0,2.0,0.0,0.0,0.0
trip_meter11,0.0,0.0,0.0,0.0,0.0
trip_meter12,1.0,0.0,0.0,0.0,3.98e-312
trip_meter13,1.0,0.0,0.0,0.0,0.0
trip_meter14,0.0,0.0,0.0,0.0,0.0
trip_meter15,1.0,0.0,0.0,1.0,0.0
trip_meter2,0.0,0.0,0.0,0.0,0.0
trip_meter3,0.0,0.0,0.0,0.0,0.0
trip_meter4,0.0,0.0,2.0,0.0,1.0
trip_meter5,0.0,0.0,2.0,0.0,1.0
trip_meter6,0.0,0.0,2.0,0.0,2.0
trip_meter7,0.0,0.0,0.0,0.0,0.0
trip_meter8,2.0,0.0,0.0,1.0,6.27e-312
trip_meter9,1.0,0.0,0.0,1.0,0.0
```

Fig. 21. Overcurrent warnings for the base case (0%)

At around 85% solar panel penetration (13 solar panel) we see the first warnings:

```
WARNING  [2022-07-01 01:00:00 PDT] : last warning message was repeated 119 times
WARNING  [2022-07-01 01:00:00 PDT] : Line:trip_line7 is at 177.03% of its emergency rating on phase 1!
WARNING  [2022-07-01 01:00:00 PDT] : last warning message was repeated 14 times
WARNING  [2022-07-01 01:00:00 PDT] : (null) - house:180 is outside of ANSI standards (voltage = 19 percent of nominal 12
0/240)
WARNING  [2022-07-01 01:00:00 PDT] : last warning message was repeated 119 times
WARNING  [2022-07-01 01:00:00 PDT] : Line:trip_line7 is at 177.03% of its emergency rating on phase 1!
```

Fig. 22. Overcurrent warnings for the base case (85%)

3) Voltage Unbalances: To determine the occurences of voltage unbalances, we need to add in GridLAB-D group recorders to track the voltages in each phase.

```
object group_recorder {
    name loads_voltages_A_recorder;
    group "groupid=meters";
    property voltage_A;
    file loads_volts_A1.csv;
    interval 1;
    limit 10000000;
}
```

Fig. 23. Group recorder instance for phase A voltages

For the base case there were no imbalances in the network. In the figure below, we can see that the count for unbalances is high for nodes 684, 611, 645, 646, and 652 because these nodes are not connected to each phase. This means that when recording voltages there will be one phase at 0V. When passed through the python script described in section 2, there will therefore be an "unbalance" at that node.

```
{'n633': 0, 'n632': 0, 'n650': 0, 'n671': 0, 'n680': 0, 'n684': 3601, 'n611': 3601, 'n634': 0, 'n645': 3601, 'n646': 3601, 'n652': 3601, 'n675': 0, 'n692':
0}
```

Fig. 24. Data for the count of voltage unbalances at each node in the network for the base case (0% penetration level)

```
{'n633': 0, 'n632': 0, 'n650': 0, 'n671': 0, 'n680': 0, 'n684': 3601, 'n611': 3601, 'n634': 0, 'n645': 3601, 'n646': 3601, 'n652': 3601, 'n675': 0, 'n692':
0}
```

Fig. 25. Data for the count of voltage unbalances at each node in the network for the case with 100% penetration level

When comparing the above figures, there is no difference. It means that for this network, the addition of evenly distributed solar panels does not affect the voltage unbalance between phases. To further evaluate this parameter, interesting next steps would be to evaluate what would happen if all houses and therefore all solar panels were placed on one phase on a specific branch.

4) Losses: To record losses in the network, a recorder object is needed. An example of that object is displayed below:

```
object group_recorder {
    name power_loss;
    group "class=overhead_line";
    property power_losses;
    file losses.csv;
    interval 1;
    limit 10000000;
    }
```

Fig. 26. The recorder needed to record the losses in the lines present in the network

Comparing the figures below, we can see similar trends for losses in the network. We can see lines 630-632 and 6321-671 having bigger losses than the rest of the lines in both cases. However the difference between them is that the loss when there is a penetration of 100% is lower on average than for the base case. It appears, for this network, that adding solar panels evenly in the network has decreased the amount of loss in the overall network. This can be explained due to the fact that power is generated near each node, so less power flows in between, through the lines.

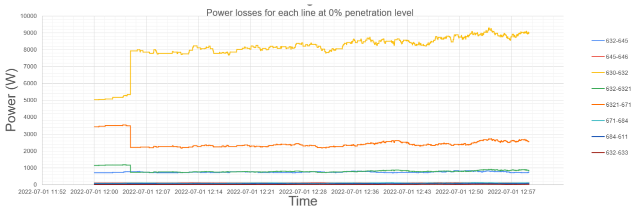

Fig. 27. Sample of the data acquired for losses in the lines of the network for the base case (0%)

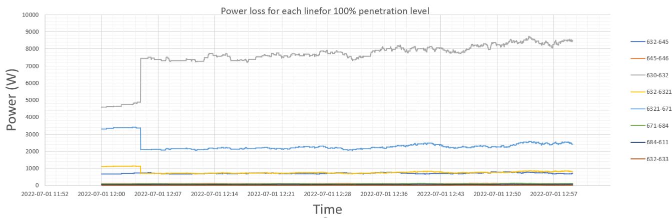

Fig. 28. Sample of the data acquired for losses in the lines of the network for the the case with 100% penetration level

5) Power Flow at the Substation: With the highest penetration level, where all houses are equipped with a solar panel (100%), the overall power going into the transformer is on average lower - although not by a considerable amount - than the base case.

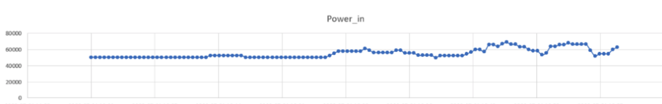

Fig. 29. Flow of apparent power through the substation transformer with the base case (0%)

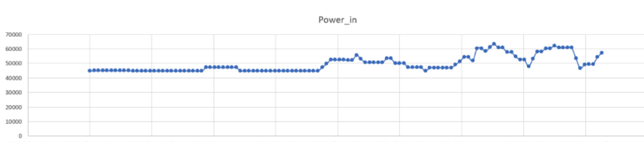

Fig. 30. Flow of apparent power through the substation transformer with the base case (100%)

D. Discussions and Conclusions

This paper built on previous work and focuses on proposing a procedure to quantify the effect of distributed energy resources on an electrical distribution network. The results obtained from the case study show that the network starts experiencing perturbations when 85% of penetration level is reached. From the two first parameters presented, it is possible to observe that when the penetration level reaches 85% - about 13 solar panels on 15 maximum - the network experiences voltage violations and some overcurrents in certain parts of the network. For the case study presented, there was no effect on the phase voltage unbalance. Gradually, adding solar panels evenly in the network did not change from base case at 0% to 100% penetration level. As for losses, it can be observed that there are less losses in the network when we increase the penetration level. We also notice that with increasing penetration levels, the power flowing from the high voltage side to the low voltage side of the transformer is reduced; this means that the demand on the low voltage side is less since the solar panels are providing a little part of the power demand. These results are obtained for the IEEE-13 network and cannot form a full conclusion on the effect of DERs in networks. They are displayed for the sole purpose of demonstration of the protocol presented in section 2 of the paper, and testing the protocol on bigger networks may therefore give more comprehensive results.

In conclusion, this paper proposes a process that could be used to evaluate the effect of DERs on a network, in hopes that it can facilitate subsequent research done using the GridLAB-D software. It was tested on a relatively small test distribution network. Testing the protocol on larger networks, as well as representations of real distribution networks, could be an interesting next step. For that, it would prove to be of use to develop a way of automating the scaling of the simulated network model, such that modeling a larger network would not have to be performed manually. The Python scripts should also be modified to accommodate for the larger data files that would come from simulating these larger networks. Moreover, it would be a good idea to use this method on a large number of networks. It would then be possible to make conclusions about how different penetration levels could affect networks in general. Additionally, the case study in this paper presented the addition of solar panels; as solar panels are not the only DERs that will be present in the future distribution networks, it would be interesting to see how other DERs, such as electric vehicle chargers and energy storage systems, would affect the grid.

ACKNOWLEDGMENTS

We would like to express our sincere appreciation to Prof. François Bouffard for his guidance, support and teachings throughout this summer project. He granted us the freedom to tap into our creativity to explore different paths during our research, while also guiding us when we hit roadblocks along the way. Having had the opportunity to work closely with him during this research, we witnessed and tried to learn as best we could from his extensive research experience. We also gratefully acknowledge the funding received towards our SURE awards that allowed us both to participate in this project. Specifically, I (Ralph Younan) would like to thank Mr. Ruben Gruber for his generous contribution which has allowed me to participate in this project, and I (Miiyu Fujita) would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support and funding.

REFERENCES

[1] H. Bahar, J. Moorhouse, Y. Abdelilah, P. Bojek, F. Briens, T. Criswell et al., "Renewable Energy Market Update - May 2022", International Energy Agency, Paris, France, 2022. Accessed: Aug. 2022. [Online]. Available: https://www.iea.org/reports/renewable-energy-market-update-may-2022

[2] Y. Abdelilah, H. Bahar, T. Criswell, P. Bojek, F. Briens, J. Moorhouse et al., "Renewables 2021", International Energy Agency, Paris, France, 2021. Accessed: Aug. 2022. [Online]. Available: https://www.iea.org/reports/renewables-2021/

[3] "Distributed Generation of Electricity and its Environmental Impacts." EPA, https://www.epa.gov/energy/distributed-generation-electricity-and-its-environmental-impactsdistributed (accessed Aug. 8, 2022)

[4] M. Bollen and S. Ronnberg, "Hosting Capacity of the Power Grid for Renewable Electricity Production and New Large Consumption Equipment," Energies, vol. 10, no. 9, p. 1325, Sep. 2017, doi: 10.3390/en10091325.

[5] Jain, Akshay Kumar, Kelsey Horowitz, Fei Ding, Nicolas Gensollen, Barry Mather, and Bryan Palmintier. 2019. Quasi-Static Time Series PV Hosting Capacity Methodology and Metrics: Preprint. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5D00-72284. https://www.nrel.gov/docs/fy19osti/72284.pdf.

[6] "GridLAB-D Wiki:GridLAB-D Tutorial Chapter 2 - GridLAB-D Key Attributes." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/GridLAB-D_Wiki:GridLAB-D_Tutorial_Chapter_2_-_GridLAB-D_Key_Attributes (accessed May 2022)

[7] "The Next-Generation Simulation Software." GridLAB-D.org. https://www.gridlabd.org/ (accessed May 2022)

[8] "Metrics collector." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/Metrics_collector (accessed Jul. 2022)

[9] "Metrics collector writer." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/Metrics_collector_writer (accessed Jul. 2022)

[10] "Difference Between Overcurrent Protection and Overload Protection." SIEMENS. https://support.industry.siemens.com/cs/document/30550697/difference-between-overcurrent-protection-and-overload-protection?dti=0lc=en-AR (accessed Aug. 2022)

[11] M. Krarti, "Utility Rate Structures and Grid Integration." in Optimal Design and Retrofit of Energy Efficient Buildings, Communities, and Urban Centers, Amsterdam, NL: Elsevier, 2018, ch. 4, sec. 7, pp. 189-245.

[12] F. Bouffard. ECSE 464. Class Lecture, Topic: "Symmetrical Components Asymmetrical Faults" Faculty of Electrical, Computer and Software Engineering, McGill University, Montreal, QC, 2021

[13] "Group recorder." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/Group_recorder (accessed Jul. 2022)

[14] "GridLAB-D Wiki:GridLAB-D Tutorial Chapter 3 - Basic Distribution System Modeling." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/GridLAB-D_Wiki:GridLAB-D_Tutorial_Chapter_3_-_Basic_Distribution_System_Modeling (accessed May 2022)

[15] J. P. Holguin, D. Celeita and G. Ramos, "Reverse Power Flow (RPF) Detection and Impact on Protection Coordination of Distribution Systems," 2019 IEEE Industry Applications Society Annual Meeting, 2019, pp. 1-6, doi: 10.1109/IAS.2019.8911979.

[16] "Power Flow User Guide." GridLAB-D Wiki. http://GridLAB-D.shoutwiki.com/wiki/Power_Flow_User_GuideTransformer (accessed May 2022)

[17] "IEEE PES Test Feeder." Resources – IEEE PES Test Feeder (accessed June 2022)