PRO1: Load Management of an Electricity Distribution Network

Monika Topel
Priscila Costa
Filippo Padovani

Motivation

Electric vehicles (EVs) are growing fast in Ellevio’s network and in some areas the installation start to challenge the power quality. More and more household customer are acquiring EVs.

This is of course a positive trend but can impact negatively on the electrical network in terms of the quality of voltage and the load rates.

In this project some of the questions are ‘how many EVs the network can host without exceeding the network limits?’, ‘What happen in the system under a contingency (N-1) case?’, ‘What are the solutions for the network host 115% EVs?’, etc.

Among another questions that we hope you will be able to answer in the end of this project.

So, let’s start!

Project Questions

- **Step 1.** Network in its original state:
  - 1.1 How’s the state of the network? Bus voltage [pu] and line/trafo loading [%].
  - 1.2 What happen in the system in a contingency (N-1) case? Are there lines that will be critical? Which lines?
    - *Consider that the maximum voltage allowed is 1.1 pu, the minimum is 0.9 pu, and the maximum line loading is 100%.*
  - 1.3 If the maximum voltage allowed is now 1.05 pu, the minimum is 0.95 pu, and the line loading is still 100%. What happen? Are there lines that will be critical? Which lines?
- **Step 2.** After the Charging Stations (CS) implementation and adding a loadshape to the existent customers (time series simulation):
  - 2.1 How’s the state of the network if compared with step 1? Better, worse, same as before…
- **Step 3.** With EV recharging on the system:
  - 3.1 Choose an amount of EVs and connect to the system. How’s the state of the network under EV recharging?
  - 3.2 How many (maximum) EVs the network can host without exceeding the limits? (max_pu=1.1, min_pu=0.9, line_loading=100) Is it more or less if it compared with what you chose previously at 3.1?
  - 3.3 What happen in a N-1 case considering the maximum EVs in the system? (max_pu=1.1, min_pu=0.9, line_loading=100)
  - 3.4 If we already have some critical lines at 3.3, how many EVs should be instead for not occur violations in a N-1 situation in the system?
- **Step 4.** Sensitivity studies:
  - 4.2 Investigate what needs to be done for the network to host 115% EVs (based from the number found in from task 3.2). In this step you can either improve the state of the network manually and/or make use of an optimization of the loads.

Step 1: CIGRE Networks

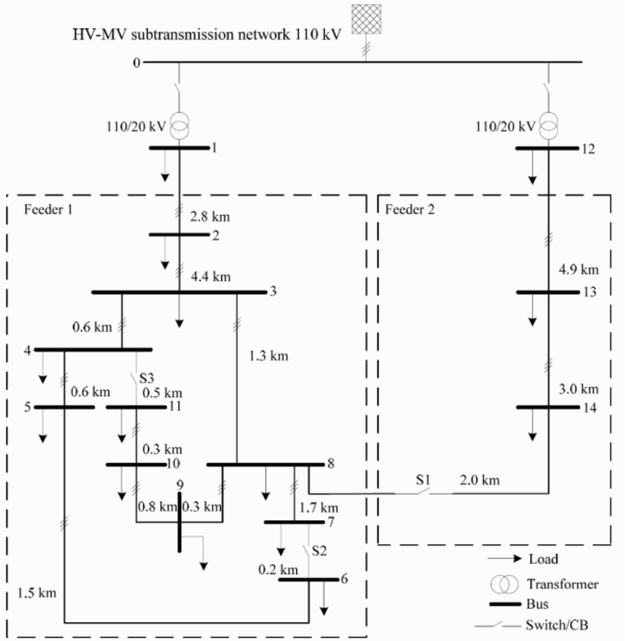
Medium voltage distribution network

Link to the Network information in pandapower

This pandapower network includes the following parameter tables:

- switch (8 elements)
- load (18 elements)
- ext_grid (1 elements)
- line (15 elements)
- trafo (2 elements)
- bus (15 elements)

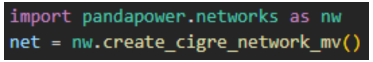
All of you will be working with the same network!



Step 1: CIGRE Networks

Medium voltage distribution network

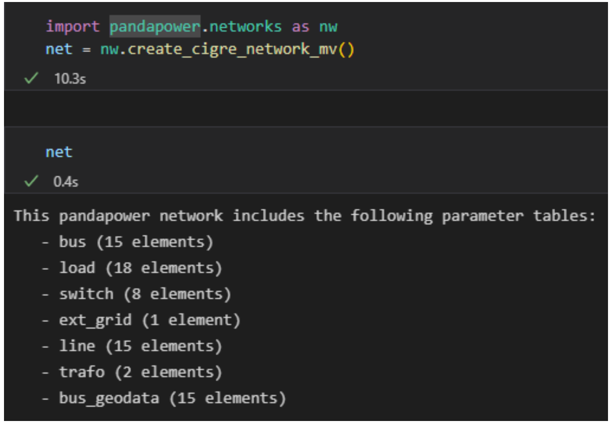
To use this Network in pandapower, you can use the following code:



```
import pandapower.networks as nw
net = nw.create_cigre_network_mv()
```

After that, you can see that the network is already implemented.

If you run ‘net’, you will receive the network parameters, as follow:



```
import pandapower.networks as nw
net = nw.create_cigre_network_mv()
```

✓ 10.3s

```
net
```

✓ 0.4s

```
This pandapower network includes the following parameter tables:
   - bus (15 elements)
   - load (18 elements)
   - switch (8 elements)
   - ext_grid (1 element)
   - line (15 elements)
   - trafo (2 elements)
   - bus_geodata (15 elements)
```

1. How’s the state of the network in its original state? For instance, bus voltage [pu] and line/trafo loading [%].

Don’t forget to answer the questions 1.2 and 1.3 as well.

Step 2: Network Modifications - CS

The idea here is that you will connect EVs in the low voltage side of the network (0.4 kV). Now, the voltage at the customer’s level is 20 kV, so we need transformers.

We will be using Basic Standard Types, and you can choose among 3 different types:

- 0.25 MVA 20/0.4 kV
- 0.4 MVA 20/0.4 kV
- 0.63 MVA 20/0.4 kV

Basically, they just differ from each other in the rated apparent power (more power, more capacity to host loads). And you can choose the types that suits you better, for what you have in mind to answer the project questions!

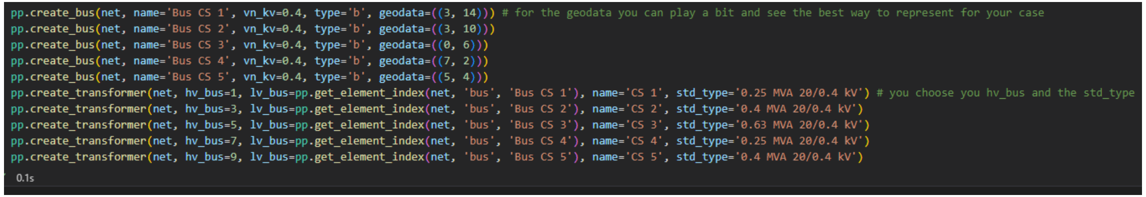
We will be allocating the EVs in charging stations (CS). So, the rule here will be that you can create the CS wherever you want (what means that you have 15 options (the buses). Got it!?). Where you decide to allocate you CS you need to connect a transformer there. And, to create the transformer we need the low voltage (LV) bus for it (where the EVs will be also connected).

We will have **5 CS** (all groups will be the same amount of it. Just the location/rated apparent power combinations will differ among you).

So, we need to create 5 LV buses and 5 transformers.

Step 2: Network Modifications - CS

The code to do that is as follow:

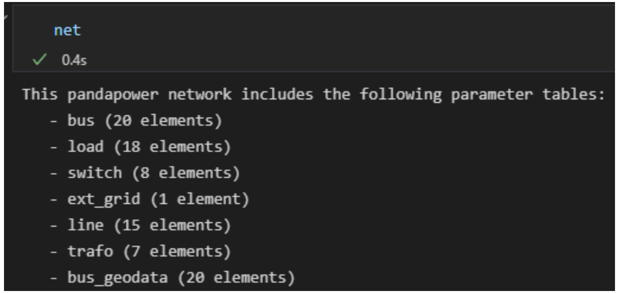


```
pp.create_bus(net, name='Bus CS 1', vn_kv=0.4, type='b', geodata=((3, 14))) # for the geodata you can play a bit and see the best way to represent for your case
pp.create_bus(net, name='Bus CS 2', vn_kv=0.4, type='b', geodata=((3, 10)))
pp.create_bus(net, name='Bus CS 3', vn_kv=0.4, type='b', geodata=((0, 6)))
pp.create_bus(net, name='Bus CS 4', vn_kv=0.4, type='b', geodata=((7, 2)))
pp.create_bus(net, name='Bus CS 5', vn_kv=0.4, type='b', geodata=((5, 4)))
pp.create_transformer(net, hv_bus=1, lv_bus=pp.get_element_index(net, 'bus', 'Bus CS 1'), name='CS 1', std_type='0.25 MVA 20/0.4 kV') # you choose you hv_bus and the std_type
pp.create_transformer(net, hv_bus=3, lv_bus=pp.get_element_index(net, 'bus', 'Bus CS 2'), name='CS 2', std_type='0.4 MVA 20/0.4 kV')
pp.create_transformer(net, hv_bus=5, lv_bus=pp.get_element_index(net, 'bus', 'Bus CS 3'), name='CS 3', std_type='0.63 MVA 20/0.4 kV')
pp.create_transformer(net, hv_bus=7, lv_bus=pp.get_element_index(net, 'bus', 'Bus CS 4'), name='CS 4', std_type='0.25 MVA 20/0.4 kV')
pp.create_transformer(net, hv_bus=9, lv_bus=pp.get_element_index(net, 'bus', 'Bus CS 5'), name='CS 5', std_type='0.4 MVA 20/0.4 kV')
```

0.1s

I chose the high voltage side buses of the transformer randomly. This do not mean is the best choice for allocate the CS in your system.

Now, if you run ’net’ again you will see the new elements add it to the list:



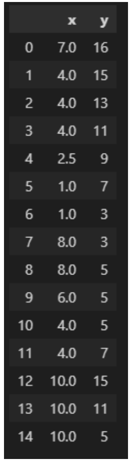
```
net
```

✓ 0.4s

```
This pandapower network includes the following parameter tables:
   - bus (20 elements)
   - load (18 elements)
   - switch (8 elements)
   - ext_grid (1 element)
   - line (15 elements)
   - trafo (7 elements)
   - bus_geodata (20 elements)
```

Step 2: Network Modifications - CS

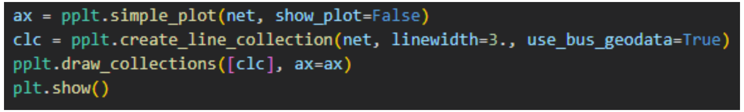
The geodata information for the buses that you need to create, for your specific case, it will depend on which buses you choose for the high voltage side of the transformer. For instance, if you run ’net.bus_geodata’ command before creating the new buses, you will get:



	x	y
0	7.0	16
1	4.0	15
2	4.0	13
3	4.0	11
4	2.5	9
5	1.0	7
6	1.0	3
7	8.0	3
8	8.0	5
9	6.0	5
10	4.0	5
11	4.0	7
12	10.0	15
13	10.0	11
14	10.0	5

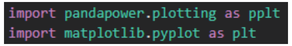
Here you can see the geodata for the 15 original buses. What I did, for my case, it was just subtract 1 point in each direction when I created the low voltage bus. For example:

- Bus CS 3 – is connected to the hv_bus 5 that have the geodata as (1, 7). So, for the lv_bus I used (0, 6). When you plot this network you will see that now the buses are not overplaced. What makes it easier the analysis in the future. You can play a bit with the coordinates and see the best option for you. The code to see the network plot is as follow:



```
ax = pplt.simple_plot(net, show_plot=False)
clc = pplt.create_line_collection(net, linewidth=3., use_bus_geodata=True)
pplt.draw_collections([clc], ax=ax)
plt.show()
```

- Don’t forget to import the right libraries:



```
import pandapower.plotting as pplt
import matplotlib.pyplot as plt
```

Step 2: Network Modifications – Loadshape

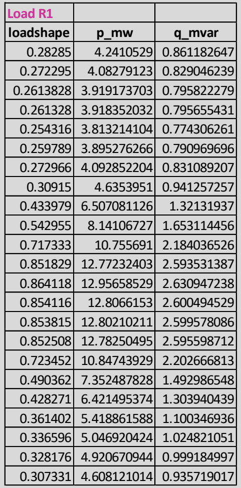
So, after allocate the CS, let’s see how to implement the EVs.
The CIGRE network already have 18 loads (customers) implemented (you can check the active and reactive power with the ´net.load` command).

But, these are just a picture of the system, which represent just 1 time step of simulation.You don’t have a time series simulation with that (daily, yearly, etc). So, we need to setup the code for a daily time series simulation, and replace the static loads for a daily loadshape.

All you have to do is use this loadshape with your existent loads:

00h: 0.28285
0.272295
0.2613828
0.261328
0.254316
0.259789
0.272966
0.30915
0.433979
0.542955
0.717333
0.851829
0.864118
0.854116
0.853815
0.852508
0.723452
0.490362
0.428271
0.361402
0.336596
0.328176
0.307331
23h: 0.297966

As you can see the loadshape is consisted of values that goes from 0 to 1.
So, you need to multiply this values for you actual power (you can see it with ’net.load’ command).
Then, after that you will have the load shape for your specific load during the 24 time steps.
For example, for load R1 we have an active power of 14.99400 and a reactive power of 3.044662. So, the daily power of this load will be:



Load R1		
loadshape	p_mw	q_mvar
0.28285	4.2410529	0.861182647
0.272295	4.08279123	0.829046239
0.2613828	3.919173703	0.795822279
0.261328	3.918352032	0.795655431
0.254316	3.813214104	0.774306261
0.259789	3.895276266	0.790969696
0.272966	4.092852204	0.831089207
0.30915	4.6353951	0.941257257
0.433979	6.507081126	1.32131937
0.542955	8.14106727	1.653114456
0.717333	10.755691	2.184036526
0.851829	12.77232403	2.593531387
0.864118	12.95658529	2.630947238
0.854116	12.8066153	2.600494529
0.853815	12.80210211	2.599578086
0.852508	12.78250495	2.595598712
0.723452	10.84743929	2.202666813
0.490362	7.352487828	1.492986548
0.428271	6.421495374	1.303940439
0.361402	5.418861588	1.100346936
0.336596	5.046920424	1.024821051
0.328176	4.920670944	0.999184997
0.307331	4.608121014	0.935719017

2. How’s the state of the network if compared with step 1?
Better, worse, same as before.. For you to run a time series simulation you can find more information and the code here.

Step 3: Network Modifications – EVs allocation

Finally, we can allocate the EVs to recharge in our system.

You can create the EV in the way you prefer, either as a ’load’ or as a ’storage’.
But, for both of those options you will use loadshapes provided by this source.

Download the file named ’**PEV-Profiles-L2.xlsx**’.

In this file you will find information for a yearly loadshape for different 348 EVs.
Then, you will use the same idea used before for the existent loads/customers to create the EV now (assigning the loadshape to your EV – storage, load).

Since you just need daily loadshapes, with this file you have tons of options.
You can choose how many vehicles you want in each CS in the step 3.1.
But of course this will be delimited from the previous parameters you chose for each CS previously (transformer rated apparent power).

Anyway, **be creative**!! ☺

But also be reasonable… you having one CS with 100 plugs available is not feasible for our network, neither as a CS.

Step 3: Network Modifications – EVs allocation

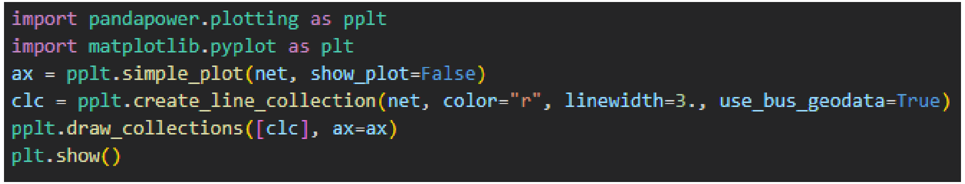
So, now all you need to do is run this network and understand what's happening with it to be able to answer the questions:

- 3.1 Choose an amount of EVs and connect to the system. How's the state of the network under EV recharging?
- 3.2 How many (maximum) EVs the network can host without exceeding the limits? (max_pu=1.1, min_pu=0.9, line_loading=100) Is it more or less if it compared with what you chose previously at 3.1?
- 3.3 What happen in a N-1 case considering the maximum EVs in the system? (max_pu=1.1, min_pu=0.9, line_loading=100)
- 3.4 If we already have some critical lines at 3.3, how many EVs should be instead for not occur violations in a N-1 situation in the system?

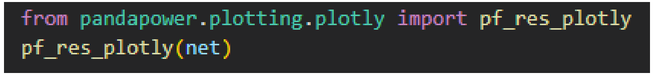
Step 4: Sensitivity studies

- 4.2 Investigate what needs to be done for the network to host 115% EVs (based from the number found in from task 3.2). In this step you can either improve the state of the network manually and/or make use of an optimization of the loads.

Last but not least, I'll provide you some pieces of code that it'll help you visualize what is happening in the network. But do not limit yourself to this options, you can find a lot of information regarding pandapower and how to visualize what you have in mind in the tutorials session, and in this link you can find the code for them.



```python
import pandapower.plotting as pplt
import matplotlib.pyplot as plt
ax = pplt.simple_plot(net, show_plot=False)
clc = pplt.create_line_collection(net, color="r", linewidth=3., use_bus_geodata=True)
pplt.draw_collections([clc], ax=ax)
plt.show()
```



```python
from pandapower.plotting.plotly import pf_res_plotly
pf_res_plotly(net)
```

Have fun ☺