POLITECNICO
MILANO 1863

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Electric Power Systems Project 2025-2026: Power Flow Analysis on Medium Sized Systems

Family name	Given Name	Student ID
Broetto	Edoardo	10852787
Profili	Andrea	10838150
Rizzello	Joshua	10844421
Savoldelli	Michele	10834079

E-mail given in the registration form: michele.savoldelli@mail.polimi.it

Academic Year 2025 - 2026

1 Slack as reference

The reason why there is no changes in voltage and current magnitute by modifying the angle of the slack bus is because this bus is the reference for all the network and the PF equations consider just the angle differences from each bus to the reference. So there will be only an angle shift if the slack angle changes.

2 Sensitivity Analysis

In order to understand the effect of the phase shifters on the system, as an example, their impact on the 2-3 line is evaluated by modifying the setting of one phase shifter at a time by 2°, over a range from -30° to 30°, and collecting in a table the corresponding active power flowing through the considered line.

Table 1: Active power flow on line 2–3 for different phase shifter angles

Phase (°)	**MW (PST 4–7)**	**MW (PST 4–9)**	**MW (PST 5–6)**
0	72.6	72.6	72.6
2	72.5	72.6	72.9
4	72.4	72.5	73.1
6	72.2	72.4	73.4
8	72.1	72.3	74.7
10	72.0	72.3	74.0
12	71.9	72.2	74.3
14	71.8	72.2	74.6
16	71.7	72.1	75.0
18	71.7	72.1	75.4
20	71.6	72.1	75.8
22	71.6	72.1	76.2
24	71.6	72.1	76.6
26	71.7	72.1	77.0
28	71.7	72.2	77.5
30	71.8	72.2	77.9
-2	72.8	72.7	72.4
-4	73.0	72.8	72.2
-6	73.1	72.9	72.0
-8	73.3	73.0	71.8
-10	73.5	73.1	71.6
-12	73.7	73.2	71.4
-14	73.9	73.4	71.3
-16	74.2	73.5	71.2
-18	74.4	73.6	71.0
-20	74.7	73.8	70.9
-22	75.0	74.0	70.8
-24	75.3	74.2	70.8
-26	75.6	74.4	70.9
-28	76.0	74.6	70.8
-30	76.3	74.8	70.8

By analyzing the data, it is also possible to assess the linearity of the three characteristics. In particular, by calculating the linear regression of the three curves, we obtain the following results.

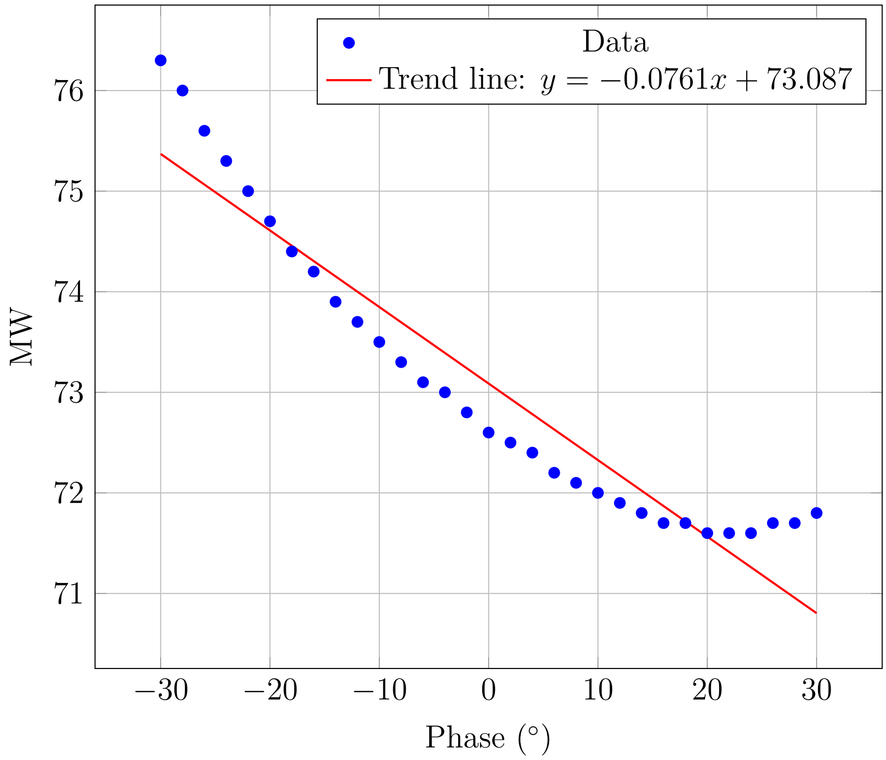

Figure 1: Sensitivity of line 2–3 for PST 4–7

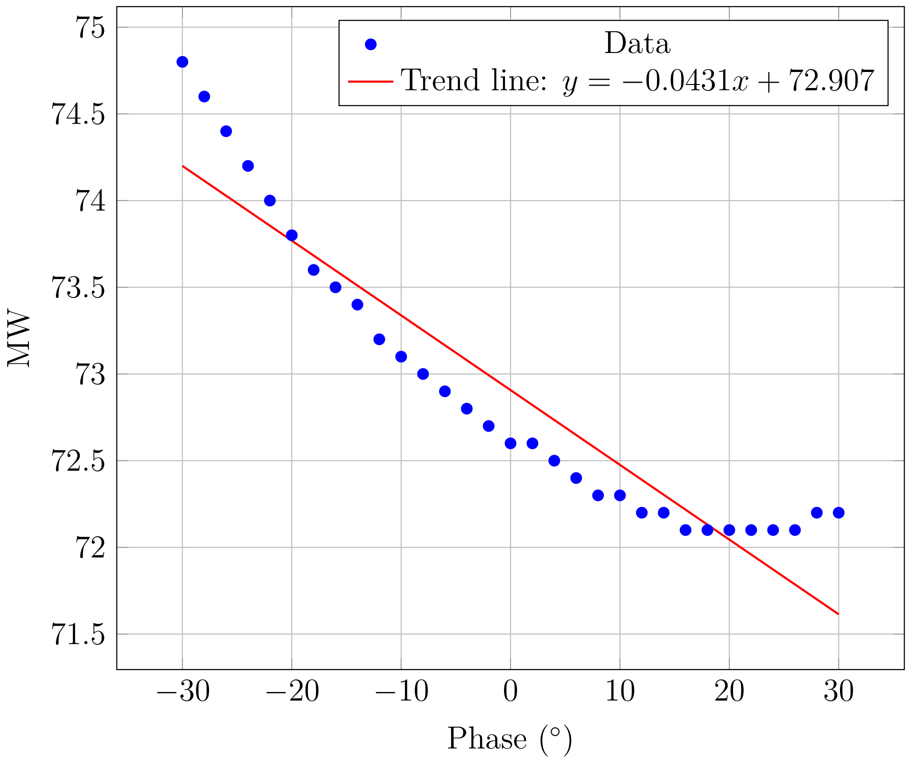

Figure 2: Sensitivity of line 2–3 for PST 4–9

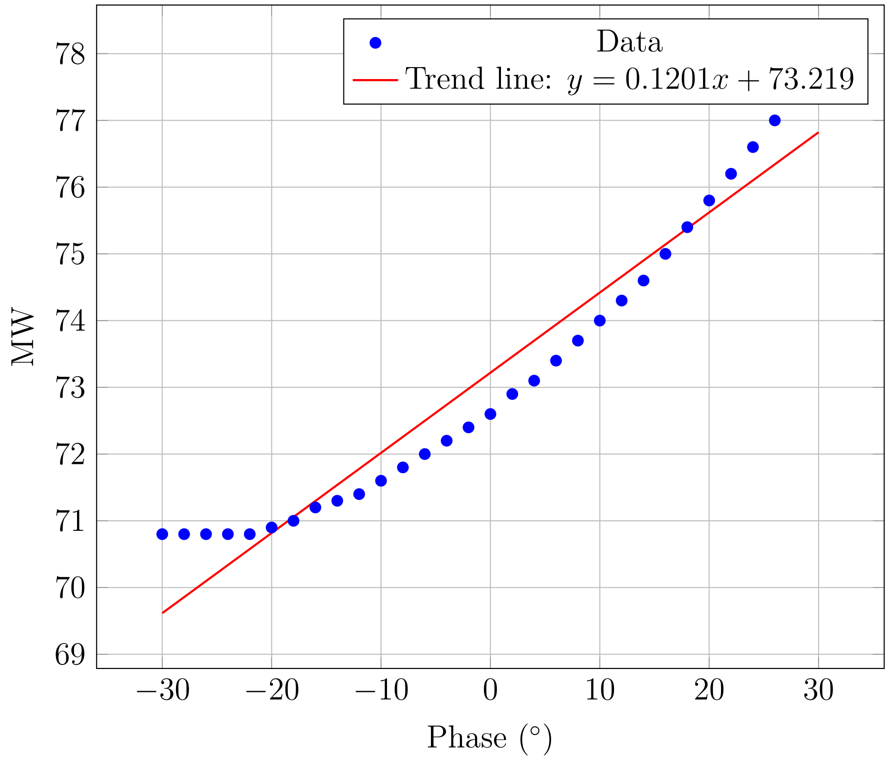

Figure 3: Sensitivity of line 2–3 for PST 5–6

Table 2: Linear regression results for the three phase shifter characteristics

PST	Trend line	Regression R^2	Linearity evaluation
4–7	$y = -0.0761x + 73.087$	0.903	Good linearity
4–9	$y = -0.0431x + 72.907$	0.883	Fair linearity
5–6	$y = 0.1201x + 73.219$	0.939	Very good linearity

A sort of saturation can be observed at the extremes of the curves, but the overall behavior remains almost linear.
It is relevant to say that, for small variation of the phase shifter angle (variation of the admittance angle θ_{pq} at a specific branch) the active power transferred follows a linear relation according to the main formula:

$$P_{pq} = \frac{V_p V_q}{X_{pq}} sin(\delta_p - \delta_q - \theta_{pq})$$

but over a certain value it starts to become non linear.
In general, since we are not observing the line of the PST but a different line in an interconnected network, the actual relation between PS angle and active power more complex due to the influence of the other busses of the network. Indeed, the PST changes the power transferred on its line, changing the overall power flow.

3 Overload correction

The system under consideration immediately appears to be in a critical condition, as there are four overloads, one of which reaches as much as 193% of the thermal limit of the line. Since these issues are severe, it is not possible to restore stability by adjusting only the phase shifters. The network analysis reveals that a large portion of the active power must travel across the grid to supply the load at bus 3, the largest in the system, which causes most of the problems. For this reason, the most intuitive solution is to generate the required power directly at bus 3 using its generator, thereby reducing the distance the active power must travel. Even after testing several combinations of generated power and modifying the phase shifter settings, the initial idea remains the most effective, leading to the minimum generated power variation. Hence this is the considered configuration 1:

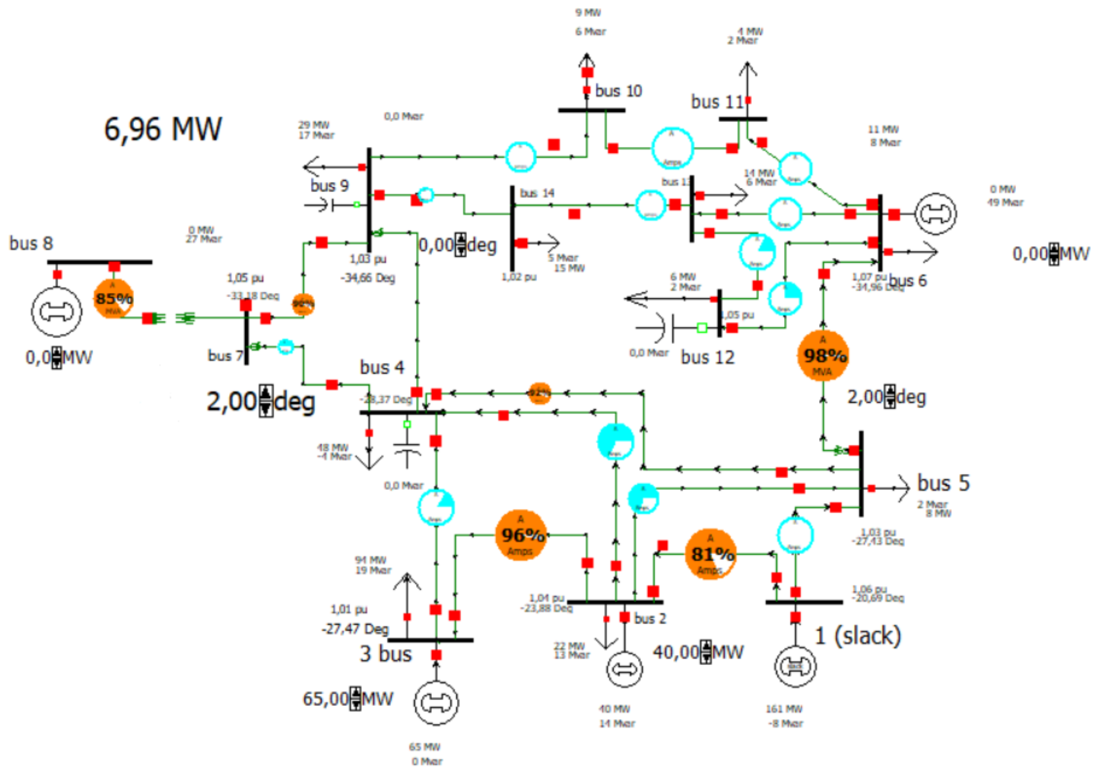

Figure 4: Configuration 1

4 Generator activation costs analysis

The solution to the cost problem is strictly related to the fixed costs of activation of the generators. In fact, since the generator no. 3 must be activated in any case to solve the overloads, the too-high fixed costs of the generators impose a solution involving only the generator no. 3 (money dependant), no. 1 and no. 2 (not money dependant). So there are no changes in the configuration: the configuration defined above remains the best one even considering the costs.

$P_{G3}[MW]$	65
$P_{G4}[MW]$	0
$P_{G5}[MW]$	0
Costs [€]	285,38

5 Impact of capacitors on losses

Closing one capacitor at a time, the obtained power losses results are shown in the table below. The capacitor connected to bus 9 (C2) is the one that, if switched on, brings to the lowest power losses.

Bus	Closed capacitor	Losses (MW)
4	C1	6,99
9	C2	6,79
12	C3	7,26

6 Capacitors linearity

In order to determine if the relationship is linear, the difference in power losses due to the closing of capacitors referred to as "Delta" and calculated for both single and double capacitor closure was evaluated. This was then compared against the ideal superposition scenario, which is the cumulative sum of power loss changes resulting from closing each capacitor individually.

	Initial power losses value (MW)		6,96	
Bus	Closed capacitor	Losses (MW)	Delta	Ideal superposition sum
4	C1	6,99	0,03	/
9	C2	6,79	-0,17	/
12	C3	7,26	0,3	/
9,4	C2 & C1	6,89	-0,07	0,19
9,12	C2 & C3	7,1	0,14	0,13

As observed in the table, the discrepancy between the measured ΔMW and the expected superposition values in both C2 C1 and C2 C3 scenarios indicates that superposition does not hold, suggesting a non-linear relationship. This occurs because the addition of a capacitor alters the voltages at nearby buses and in particular the one on the second capacitor's bus: this leads to a different reactive power injection compared to the ideal superposition case, because in that case the voltage at the second capacitor's bus would be different and so its reactive power injection. A variation in reactive power leads to a different current flow, resulting in different power losses.

7 Transformer and capacitors optimal configuration

By starting from the previous configuration (2) and modifying only the LTC transformer tap ratio and the capacitor switch positions, power losses are reduced and line overloads, particularly on the line from bus 8 to bus 7, are mitigated. This improvement is achieved thanks to the LTC transformer and the capacitor connected to bus 9, with capacitor settings guided by the information obtained from points 8 and 9. The capacitor linked to bus 9 is indeed the one that brings to the lowest power losses.

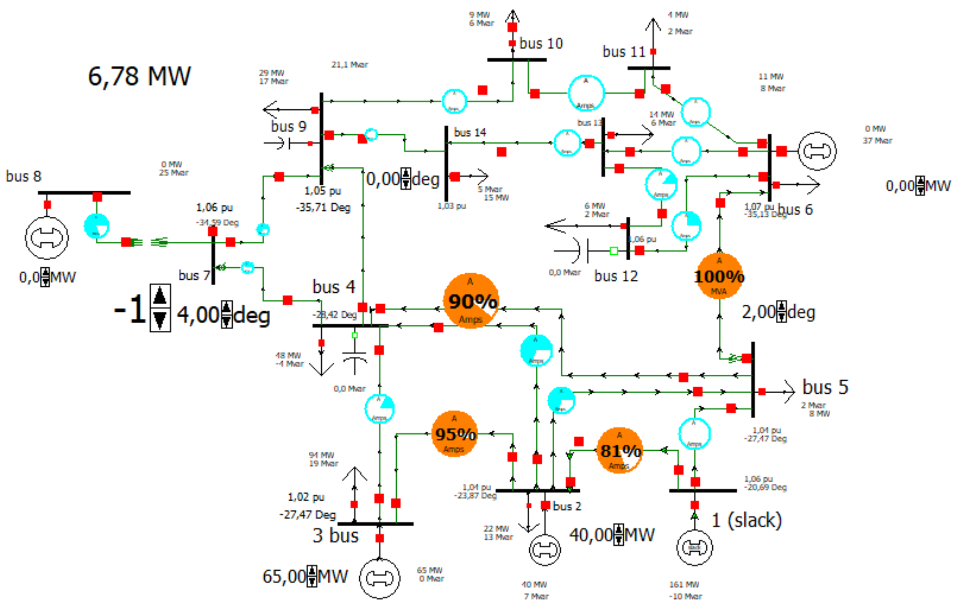

Figure 5: Configuration 3

8 Contingency Analysis

By analyzing configuration 3 with the contingency analysis tool and setting the evaluation to single faults, a list of all possible faults and the overloads they cause is obtained. To compile a list of the most critical faults, it is necessary to define specific evaluation parameters.

The first parameter is the number of overloaded lines caused by the fault, because the higher the number of overloaded lines, the higher the number of possible combinations of subsequent faults that could potentially lead to a system blackout.

If several faults result in the same number of overloads, a second evaluation parameter is considered, observing which lines are overloaded by the fault. In particular, a fault is considered more dangerous if it overloads lines that have already been identified as critical according to the first criterion.

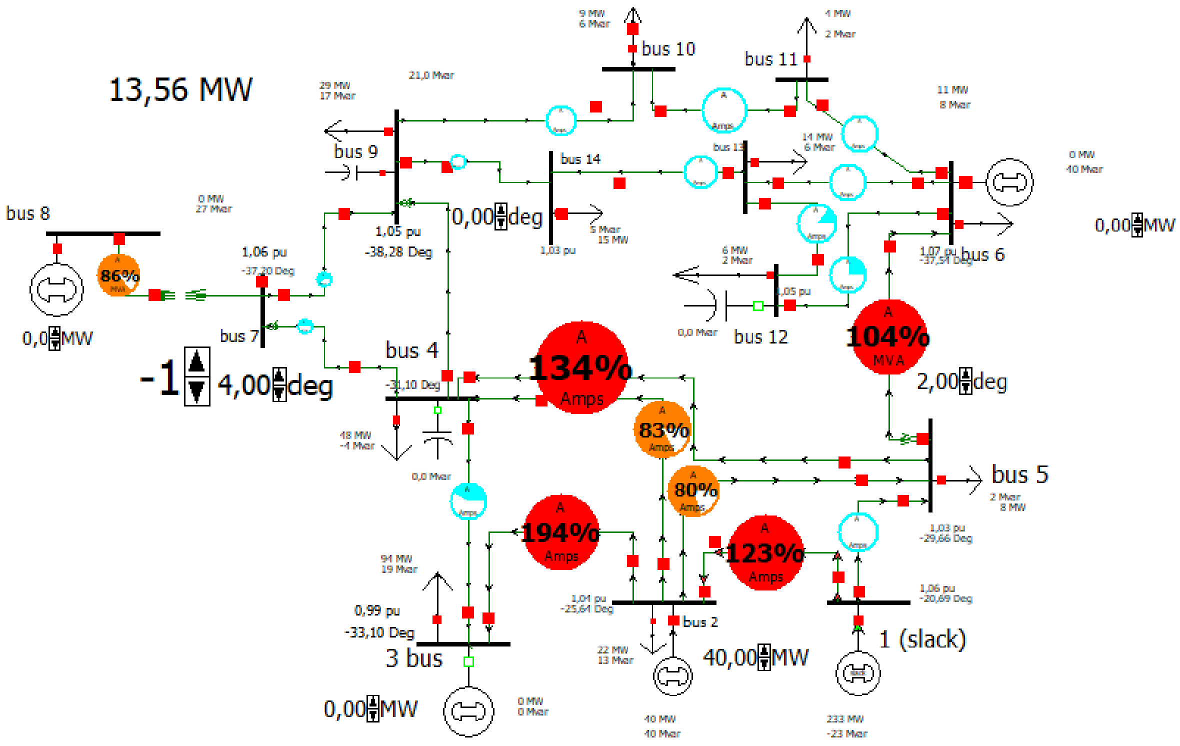

Figure 6: 1st worst case - generator on bus 3 disconnected

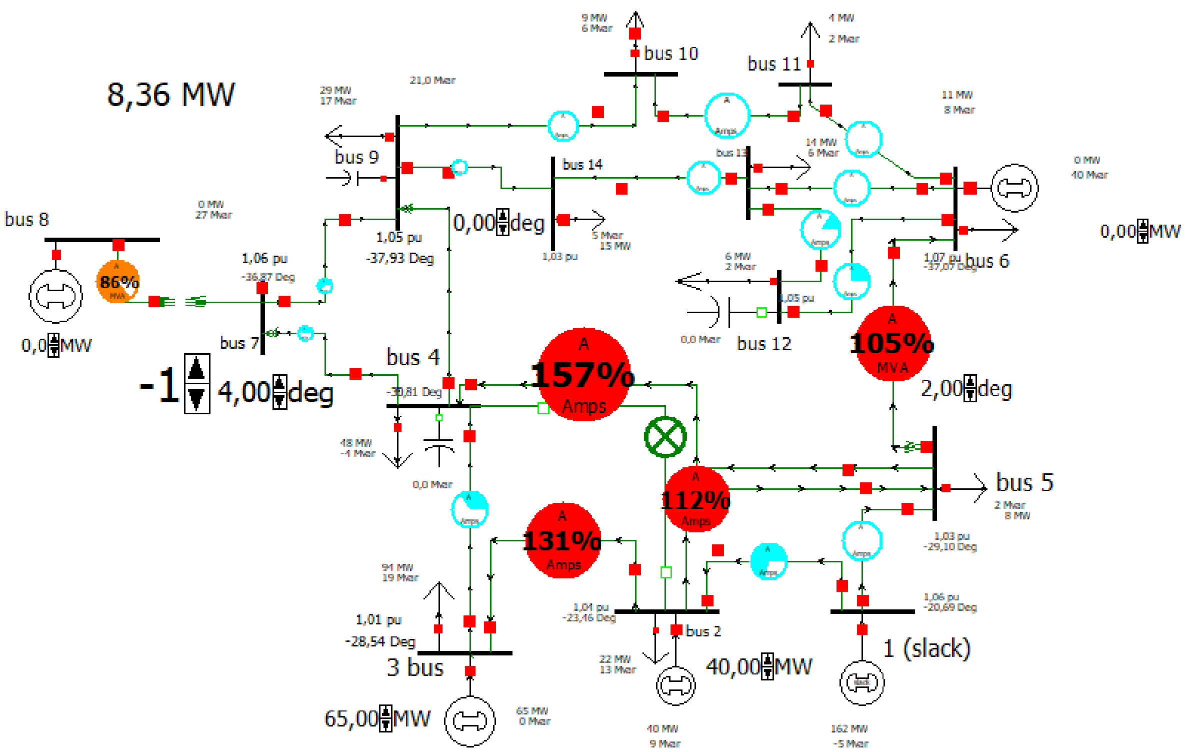

Figure 7: 2nd worst case - line connetecting bus 2 and 4

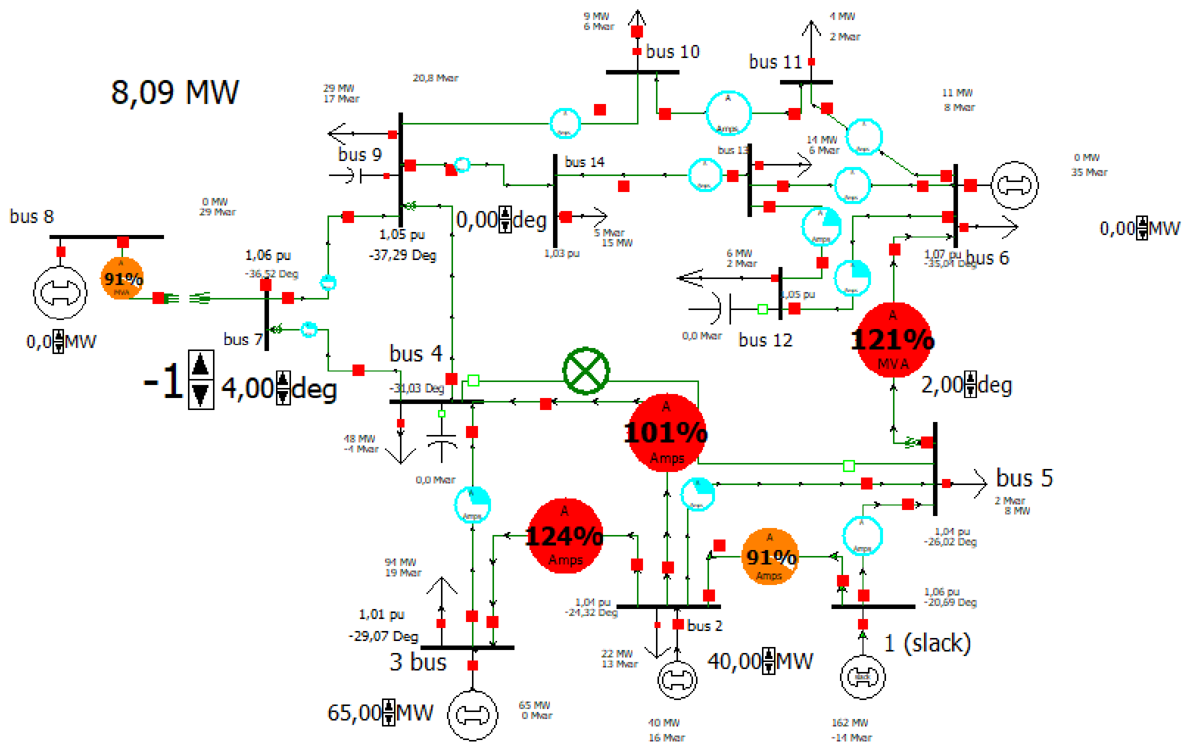

Figure 8: 3rd worst case - line connecting bus 4 and 5

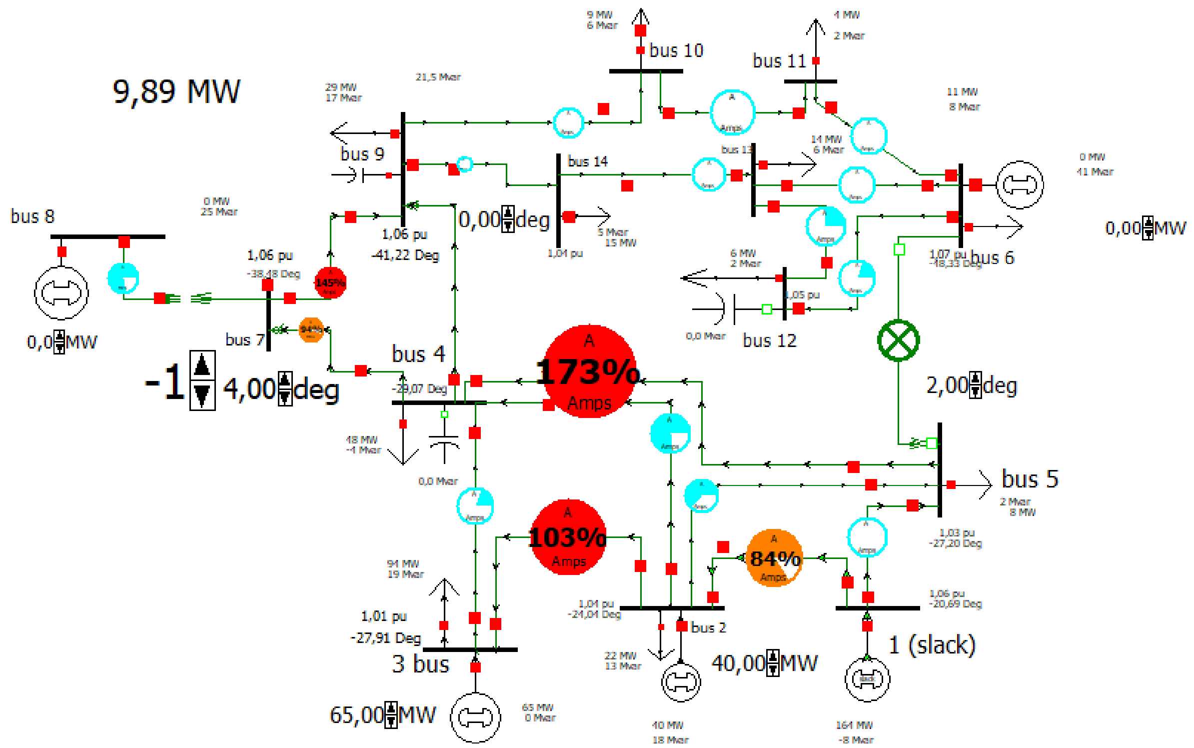

Figure 9: 4th worst case - line connecting bus 5 and 6

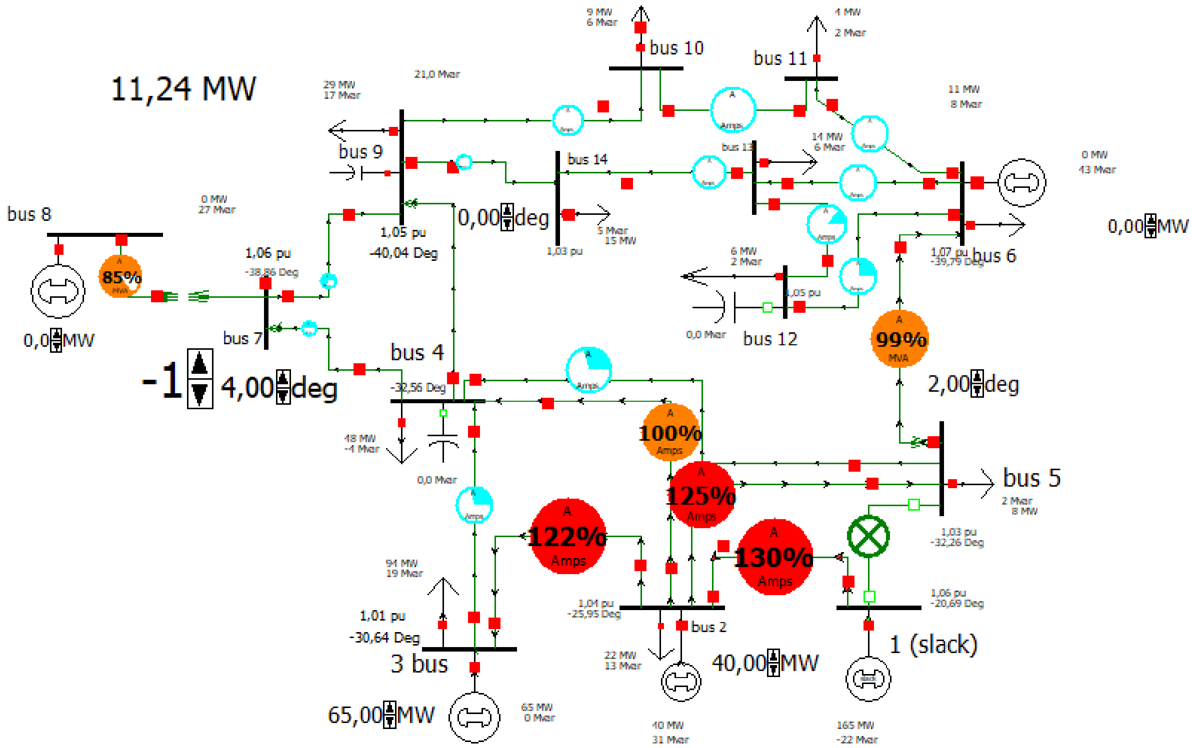

Figure 10: 5th worst case - line connecting bus 1 and 5

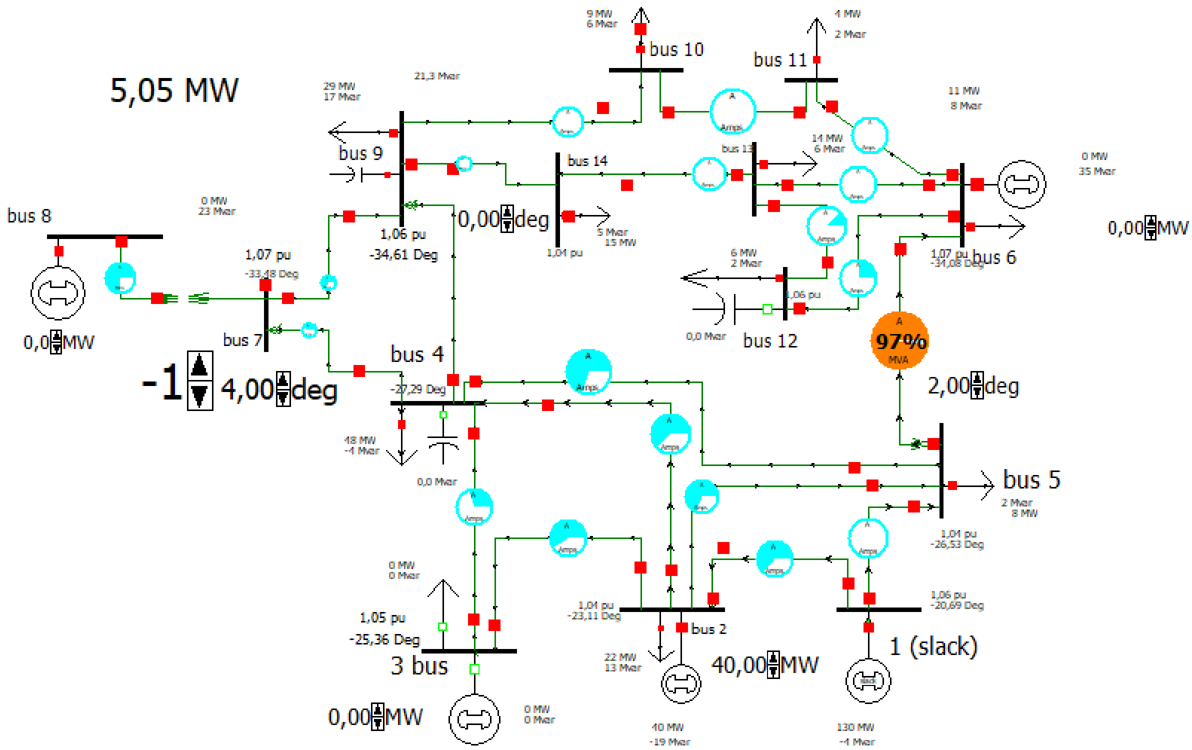

Figure 11: Solution to the 1st worst case

The solution to the worst contingency cannot be achieved solely by adjusting the generated power or by varying the tap changer or phase shifter; it is therefore necessary to disconnect a load. The selected load is the one connected to bus 3, like the faulty generator. Not only is it the largest load in the network, but losing the local generation that supplies it predominantly makes it necessary to transfer a lot of power through several network lines, overloading them.

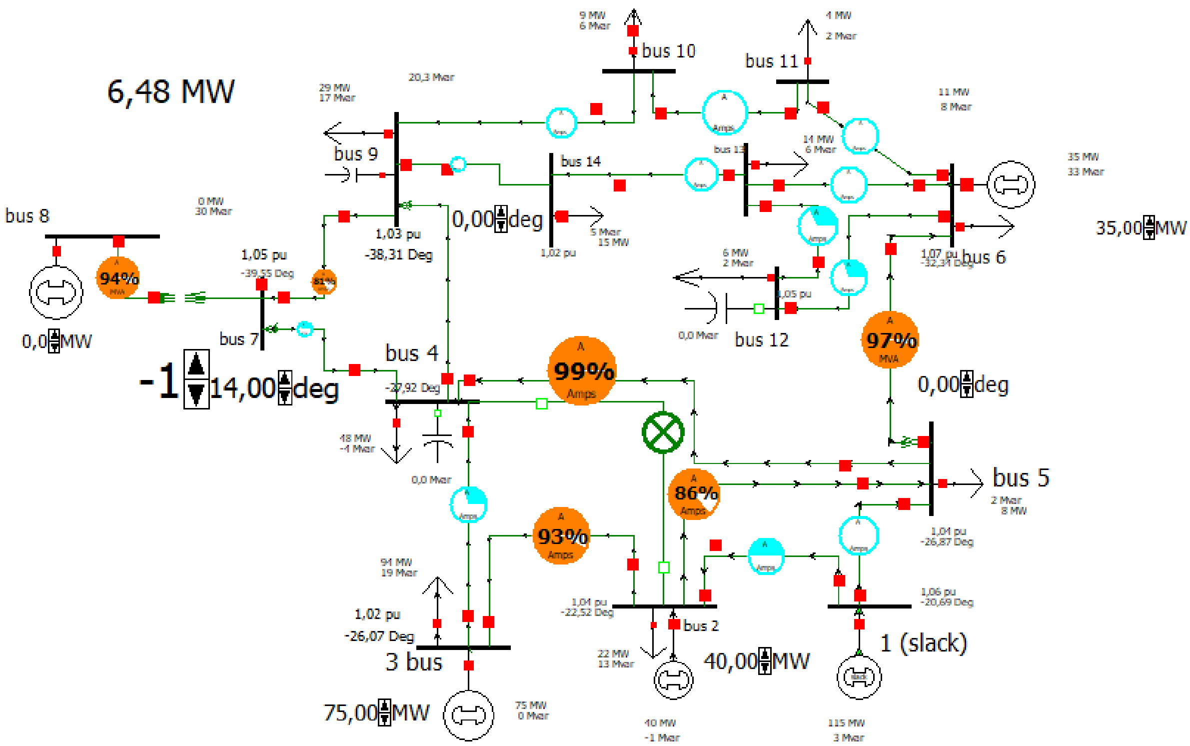

Figure 12: Solution to the 2nd worst case

The solution to the outage of the 4-6 line can be achieved without shedding any load. By increasing the generation of the unit at bus 3 and turning on the generator at bus 6, as well as adjusting the phase-shifter settings on the lines between buses 4-7 and 5-6, it is possible to eliminate all existing overloads.

9 Is the system N-1 secure?

No, the system is not N-1 secure because, according to TERNAs definition, a system must not reach abnormal conditions after the loss of a critical network element in order to be considered N-1 secure. Indeed, as shown by the contingencies reported in the previous section, the loss of a single element can cause up to four overloads, making the system non-compliant with the N-1 security criterion.

10 Analysis of PF computational methods

Since there are different mathematical methods to solve a Power Flow, that differ, for example, by the speed of convergence or the robustness in case of wide networks, there will be used and analyzed these 4: *Newton-Raphson, Gauss Seidel, DC PF and Fast Decoupled*. It is useful to give a little review of these methods before going into details.

- **Newton-Raphson**: it uses the sequential linearisation. The fundamental concept is to linearise the function that characterise the network and, from an initial guess, evaluate the variable we are interested to know. It is the fastest since it has a quadratic convergence while approaching the solution.
- **Gauss Seidel**: it is still an iterative method that consists in writing the equations in an implicit form ($ex : V_i^{(k+1)} = f(V_1^k, V_2^k, Y, ...)$) and solving them repeatedly until the solution converges, starting from an initial guess. The key feature is that each new value is computed using the most recently updated values of the other variables, which usually improves convergence. It is used to generate an initial guess of a system before solving a Power Flow with faster methods. However, on large networks it can converge very slowly or even fail to converge, especially in the presence of negative branch reactances.
- **DC PF**: it is a non iterative method and it is useful for a first and general view of the network in terms of voltages angles and real power. The losses are neglected and all the voltages are assumed equal to one. The equations can be summarised in this matrix relation: $[\tilde{P}] = [\tilde{B}][\delta]$, where $\tilde{B}$ is a matrix that takes into account only the reactances of the branches.
- **Fast Decoupled PF**: it is by far one of the most used method for computing a PF, since it is 40% computationally lighter than the Newton-Raphson method. The ability to be lighter derives from the fact that the Jacobian is not computed at every iteration (NR does it), and from other physical assumptions that are valid in most of the real networks.

Now it is time to give an explanation of the results that have been obtained from the resolution of the power flow.
There have been considered **50 iterations** with an accuracy of $0.01 \frac{MW}{Mvar}$ and the results are the following:

Bus	Volt [kV]	δ [Deg]
1	73,14	-20,69
2	72,105	-23,87
3	70,211	-27,47
4	71,273	-28,42
5	71,596	-27,47
6	14,766	-35,13
7	14,673	-34,59
8	19,62	-34,59
9	14,553	-35,71
10	14,488	-35,89
11	14,578	-35,64
12	14,56	-35,98
13	14,493	-36,05
14	14,278	-36,86

Table 3: Newton Raphson

Bus	Volt [kV]	δ [Deg]
1	73,14	-20,69
2	72,105	-23,87
3	70,211	-27,47
4	71,273	-28,42
5	71,596	-27,47
6	14,766	-35,13
7	14,673	-34,59
8	19,62	-34,59
9	14,553	-35,71
10	14,488	-35,89
11	14,578	-35,64
12	14,56	-35,98
13	14,493	-36,05
14	14,278	-36,86

Table 4: Fast Decoupled

With respect to these results it is possible to say that the differences between Newton Raphson and Fast Decoupled are negligible and both are converging to a stable Power Flow solution.

Bus	Volt [kV]	δ [Deg]
1	73,14	-20,69
2	72,105	-23,81
3	69,69	-27,37
4	71,186	-28,27
5	71,554	-27,34
6	14,766	-34,85
7	14,664	-34,31
8	19,62	-34,31
9	14,542	-35,41
10	14,479	-35,59
11	14,573	-35,35
12	14,558	-35,68
13	14,491	-35,75
14	14,271	-36,56

Table 5: Gauss Seidel

Bus	Volt [kV]	δ [Deg]
1	69	-20,69
2	69	-24,06
3	69	-28,07
4	69	-28,68
5	69	-27,68
6	13,8	-35,77
7	13,8	-35,19
8	18	-35,19
9	13,8	-36,51
10	13,8	-36,81
11	13,8	-36,49
12	13,8	-36,88
13	13,8	-37,04
14	13,8	-38,04

Table 6: DC PF

It is not possible to say the same as in the previous cases since the Power Flow using Gauss Seidel method is not converging to a solution according to our mismatches and number of iterations. One question to ask would be if the power flow converges considering an higher number of iterations. The answere is no, the power flow using Gauss Seidel method does not converge, it diverges for any number of iterations.
From the PowerWorld results, it becomes clear that the problematic elements of the solution are the voltage at bus 3 and the power output required from the generator at that bus -the same generator used to relieve the overload on branch 23.
Since a $P = 65MW$ from the generator has been requested on bus 3, and the bus clearly passeses from a PV to a PQ one, the Gauss Seidel method results in failure. The link between the switching of the type of bus and the diverging of PF using Gauss Seidel is verifiable by computing the same power flow with a power that avoids the change of bus type ($ex : P = 40MW$) and observing that a solution can be found without problem (if $P = 40MW$ the number of iterations required is 116).

The results of the Power Flow by using the DC method are pretty clear. The voltages are considered constant to their nominal value, the losses are neglected and only the angles are computed. To have a comparison of the accuracy of this method, the highest percent difference between the δ_{DC} compared to the δ_{NR} is $PG = 3,15\%$.

Now an analysis of power flow methods by computing a **single iteration** can be carried out. An accuracy of $0.01\frac{MW}{Mvar}$ as before is considered; the results are the following:

Bus	Mismatch MW	Mismatch Mvar	Mismatch MVA
1 (slack)	0	0	0
2	11,55	-0,14	11,55
3	-1,21	-0,63	1,36
4	-38,85	6,85	39,44
5	2,11	21,81	21,91
6	3,69	-3,08	4,8
7	9,57	19,16	21,42
8	-1,93	-3,57	4,06
9	-16,58	-5,1	17,35
10	-1,58	-2,74	3,16
11	-3,69	9,39	10,09
12	-3,73	6,86	7,81
13	-6,94	12,63	14,41
14	-3,31	-3,5	4,82

Table 7: Newton Raphson

Bus	Volt (kV)	Angle (Deg)
1	73,14	-20,69
2	72,152	-22,76
3	69,904	-25,18
4	71,418	-25,33
5	71,517	-24,92
6	14,963	-31,01
7	14,669	-30,69
8	19,913	-30,53
9	14,654	-30,97
10	14,603	-31,1
11	14,636	-30,78
12	14,614	-30,94
13	14,571	-31,04
14	14,475	-31,85

Table 8: Newton Raphson 1 it

Bus	Mismatch MW	Mismatch Mvar	Mismatch MVA
1 (slack)	0	0	0
2	-19,87	0	19,87
3	-15,04	0	15,04
4	-10,18	81,82	82,45
5	-2,16	0,03	2,16
6	-45,32	0	45,32
7	-28,34	8,6	29,62
8	-0,46	0	0,46
9	-30,58	14,33	33,77
10	-3,8	17,95	18,35
11	0	0	0
12	3,19	15,04	15,38
13	-9,57	5,39	10,98
14	-0,02	0,01	0,02

Table 9: Gauss Seidel

Bus	Volt [kV]	δ [Deg]
1	73,14	-20,69
2	72,105	-20,13
3	69,69	-20,92
4	69,491	-20,51
5	70,819	-20,6
6	14,766	-21,97
7	14,234	-21,6
8	19,62	-21,56
9	13,924	-21,82
10	13,808	-21,71
11	14,23	-21,94
12	14,165	-22,07
13	14,295	-22,18
14	13,829	-23,04

Table 10: Gauss Seidel 1 it

Bus	Mismatch MW	Mismatch Mvar	Mismatch MVA
1 (slack)	0	0	0
2	16,43	0	16,43
3	9,6	0	9,6
4	-23,7	-0,66	23,71
5	-14,25	0,42	14,25
6	50,3	0	50,3
7	-2,74	-0,03	2,74
8	0,44	0	0,44
9	1,26	1,1	1,67
10	3,45	0,11	3,45
11	-15,14	0,21	15,14
12	-11,94	0,16	11,94
13	-21,17	0,4	21,17
14	3,81	-0,17	3,81

Table 11: Fast Decoupled

Bus	Volt [kV]	δ [Deg]
1	73,14	-20,69
2	72,105	-23,9
3	69,69	-27,46
4	71,462	-27,62
5	71,743	-26,81
6	14,766	-35,49
7	14,7	-33,77
8	19,62	-33,81
9	14,591	-35
10	14,537	-35,22
11	14,689	-34,72
12	14,734	-34,91
13	14,641	-35,09
14	14,33	-36,34

Table 12: Fast decoupled 1 it

Here there is a comparison among the maximum mismatch ([MVA]) after one iterarion of each method.

Newton Raphson	Gauss Seidel	Fast Decoupled
39,44	82,45	50,3

As expected, for this network the most effective method is Newton-Raphson, provided appropriate initial values are used.

11 Implementation of Fast-Decoupled in Matlab

We show the implementation of a Matlab script that performs a Fast Decoupled power flow by providing any input data. We assumed that the input datas are always presented in the same format provided for this project.

FD is based on a mixed formulation of power flows:

- voltages in polar coordinates and
- elements of [Y] in rectangular coordinates

So the equations become: $\begin{cases} P_p = V_p \sum_{q=1}^{n} (G_{pq} V_q \cos\delta_{pq} + B_{pq} V_q \sin\delta_{pq}) \\ Q_p = V_p \sum_{q=1}^{n} (G_{pq} V_q \sin\delta_{pq} - B_{pq} V_q \cos\delta_{pq}) \end{cases}$

Assuming the real-reactive decoupling, and computing the Jacobian elements, and then assuming also $\delta_p - \delta_q \approx 0 \Rightarrow \begin{cases} \cos\delta_{pq} \approx 1 \\ \sin\delta_{pq} \approx 0 \end{cases}$

and considering $\begin{matrix} G_{pq} << B_{pq} \\ Q_p << B_{pp} V_p^2 \end{matrix}$

We obtain $\begin{cases} \frac{\Delta P_p}{V_p} = \sum_{q=1}^{n} (-B_{pq})(V_q \Delta\delta_q) \\ \frac{\Delta Q_p}{V_p} = \sum_{q=1}^{n} (-B_{pq})(\Delta V_q) \end{cases}$

And with further assumptions: $\begin{cases} \left[\frac{\Delta P}{V}\right] = [B'][\Delta\delta] \\ \left[\frac{\Delta Q}{V}\right] = [B''][\Delta V] \end{cases}$

At this point, the elements of the $[B']$ and $[B'']$ matrices are strictly negated elements of the imaginary part of the bus admittance matrix $[Y]$

- $[B']$ is built omitting the row and column relevant to the slack bus
- $[B'']$ is built omitting rows and columns relevant to the slack and PV busses

11.1 Code

```
clear; clc;

%download of the information from the excel
bus_data=readtable('Data.xlsx', Sheet='Bus');
gen_data=readtable('Data.xlsx', Sheet='Generators');
branch_data=readtable('Data.xlsx', Sheet='Branches');

A_ref = 100; %MW
%construction of the Y matrix by inspection, without considering the phase
%shifter/tap changer of the transformers
Y=NaN(length(bus_data.Node));
for i=1:length(bus_data.Node)
    for j=1:length(bus_data.Node)
        if(i==j) %on the diagonal
            cond=(branch_data.From_node==i|branch_data.To_node==i); %the line that is connected to the bus
            Y(j,i)=sum((branch_data.R_pu_(cond)+1i*branch_data.X_pu_(cond)).^-1+(1i*branch_data.B_pu_(cond)+...
            branch_data.G_pu_(cond))/2);
        else %off diagonal
            cond = (branch_data.From_node == i & branch_data.To_node == j) |...
                   (branch_data.From_node == j & branch_data.To_node == i); % the line that is between a bus and the other
            Y(j,i)=-sum((branch_data.R_pu_(cond)+1i*branch_data.X_pu_(cond)).^-1);
```

```
        end
    end
end
%------
slack = bus_data.Node(contains(bus_data.Type,'SLACK')); %which bus is slack
PV_buses = [bus_data.Node(contains(bus_data.Type,'PV'))]; %indices of PV_buses
%------
%find the the branch where the transformer is
t=contains(branch_data.Type,'Transformer');
%gives the condition of which transofrmer is on HV-MV
c=bus_data.Vn_kV_(branch_data.From_node(t~=0))>bus_data.Vn_kV_(branch_data.To_node(t~=0));
%create the phase shifters indeces matrix (first col=HV, second col=MV)
Ph_shifters=[branch_data.From_node(t).*c, branch_data.To_node(t).*c]+...
[branch_data.To_node(t).*(c==0),branch_data.From_node(t).*(c==0)];
%------

%calculate the parameters of the transformers
y_sc=1./(1i*branch_data.X_pu_(t)+branch_data.R_pu_(t));
b0 = branch_data.G_pu_(t)+1i*branch_data.B_pu_(t);
tap_position = [0; 0; 0; 0]; %put in each row the tap position of the corresponding PST in the
                             %vector above
tap_nominal = [0; 0; 0; 0];
ontr = [branch_data.Off_nominalTapRatio_pu_(t)];      %off nominal tap ratio of each transformer

delta_V_pu = branch_data.Step(t).*contains(branch_data.TypeOfController(t),'AVR'); %delta_V pu
omega = [0; 0; 0; 0;];%insert the angle position
N_pu = ontr.*(1+(tap_position-tap_nominal).*delta_V_pu).*exp(1j*omega);

%CODE TO INSERT THE TRANSFORMERS MATRIXES INTO Y
if any(N_pu-1) %true if at least 1 element of N_pu-1 is different from zero (i.e. not all N_pu=1)
    for i=1:length(Ph_shifters)
        %Insert A
        %the first element remains the same because we already included y_sc-jb0/2 in Y
        %Insert B
        Y(Ph_shifters(i,2), Ph_shifters(i,1)) = Y(Ph_shifters(i,1), Ph_shifters(i,2))+y_sc(i)-y_sc(i)/N_pu
        %Insert C
        Y(Ph_shifters(i,1), Ph_shifters(i,2)) = Y(Ph_shifters(i,2), Ph_shifters(i,1))+y_sc(i)-y_sc(i)/(N_pu')
        %Insert D
        Y(Ph_shifters(i,1), Ph_shifters(i,1)) = Y(Ph_shifters(i,1), Ph_shifters(i,1))-(y_sc(i)-1j*b0(i)/2)+...
        (y_sc(i)-1j*b0(i)/2)/abs(N_pu)^2

    end
end


N = length(bus_data.Node);
deltas = zeros(N, 1);

rows = true(1, N);
cols = true(1, N);

%Construction of [B']
rows(slack) = false;  % exclude the slack row
cols(slack) = false;  % exclude the slack column
B_prime = -imag(Y(rows, cols));

%Construction of [B'']
if ~isempty(PV_buses)
    rows (PV_buses) = false; %exclude also the PV buses rows
    cols (PV_buses) = false; %exclude also the PV buses columns
end
B_second = -imag(Y(rows, cols));

V = ones(N, 1);
if ~isempty(PV_buses)
    V(~rows) = gen_data.V_pu_; %Assign the correct voltage value to PV_buses
end

P_tilde(~rows) = gen_data.Pgen_MW_-bus_data.P_load_MW_(~rows); %PV buses and slack
P_tilde(rows) = -bus_data.P_load_MW_(rows);                    %other buses
P_tilde = P_tilde';
P_tilde = P_tilde./A_ref; %in PU
Q_tilde(rows) = -bus_data.Q_load_Mvar_(rows); %the same
Q_tilde(~rows) = 0;
Q_tilde = Q_tilde.';
```

```matlab
Q_tilde = Q_tilde./A_ref; %in PU

epsilon = 1e-4; %Precision required

%Compute the initial values for P
P_c = P_calc(deltas, slack, V, Y);
%Compute the initial residues
res_P = (P_tilde-P_c)./V;
deltas(1:end ~= slack) = deltas(1:end ~= slack) + B_prime^-1*res_P(1:end ~= slack);
%Compute the initial values for Q
Q_c = Q_calc(deltas, slack, V, Y, PV_buses);
%Compute the initial residues
res_Q = (Q_tilde-Q_c)./V;
V(rows) = V(rows) + B_second^-1*res_Q(rows);

%Iterations
counter = 0;
while max(abs([res_P; res_Q])) > epsilon

    P_c = P_calc(deltas, slack, V, Y);
    res_P = (P_tilde-P_c)./V;
    deltas(1:end ~= slack) = deltas(1:end ~= slack) + B_prime^-1*res_P(1:end ~= slack);
    Q_c = Q_calc(deltas, slack, V, Y, PV_buses);
    res_Q = (Q_tilde-Q_c)./V;
    V(rows) = V(rows) + B_second^-1*res_Q(rows); %rows is a logic vector that contains false on the rows
                                                % corresponding to PV buses AND slack bus. We want to
                                                % perform the computation only on PQ buses, so this
                                                % is the way.

    Q_c_PV = Q_calc (deltas, slack, V, Y, setdiff(1:length(deltas), [slack; PV_buses])); %This calculates Q AT the PV buses
    %Check if the generators reach the Q limits. In that case, remove their
    %index from PV_buses and include them in rows and cols
    Q_gen = Q_c_PV(PV_buses) + bus_data.Q_load_Mvar_(PV_buses)/A_ref;
    for n = 1:length(Q_gen)
        if ( Q_gen(n)>gen_data.Qmax_Mvar_(gen_data.Bus==PV_buses(n))/A_ref |...
        Q_gen(n)<gen_data.Qmin_Mvar_(gen_data.Bus==PV_buses(n))/A_ref )

            %Re-construction of [B''] after changing the bus type
            rows (PV_buses(n)) = true; %re-include this row
            cols (PV_buses(n)) = true; %re-include this column
            B_second = -imag(Y(rows, cols));

            %Q_tilde of the new PQ bus
            if      Q_gen(n)>=gen_data.Qmax_Mvar_(gen_data.Bus==PV_buses(n))/A_ref
            Q_tilde(PV_buses(n)) = (gen_data.Qmax_Mvar_(gen_data.Bus==PV_buses(n))-bus_data.Q_load_Mvar_(PV_buses(n)))/A_ref;
            elseif Q_gen(n)<=gen_data.Qmin_Mvar_(gen_data.Bus==PV_buses(n))/A_ref
            Q_tilde(PV_buses(n)) = (gen_data.Qmin_Mvar_(gen_data.Bus==PV_buses(n))-bus_data.Q_load_Mvar_(PV_buses(n)))/A_ref;
            end

            PV_buses(n) = 0;
         end
    end
    PV_buses(PV_buses==0) = [];
    counter = counter + 1;
end


%Functions
function P = P_calc(delta, slack, Voltages, Adm)
    P= zeros(length(delta), 1);
    for i=1:length(delta)
        if i~= slack
            P(i)=Voltages(i)*(Voltages.'*(abs(Adm(i, :).').*cos(delta(i)-delta-angle(Adm(i,:).'))));
        end
    end
end

function Q = Q_calc(delta, slack, Voltages, Adm, PV_location)
    Q= zeros(length(delta), 1);
    for  i = setdiff(1:length(delta), PV_location) %setdiff creates a vector of integers from 1 to
                                                   %length(delta)=N excluding the ones in PV_location
        if i ~= slack
            Q(i)=Voltages(i)*(Voltages.'*(abs(Adm(i, :).').*sin(delta(i)-delta-angle(Adm(i,:).'))));
        end
    end
```

```
end
```