18-875/19-739 Engineering and Economics of Electric Energy Systems Take Home Midterm Exam Due March 7, 2020

Question 1. (100 pts)

Code in Matlab AC power flow using Matpower case (MPC) format. Your function should take an MPC structure as input and return an MPC structure of the solution. You are not allowed to use Matpower functions. Your program should handle networks with a single generator and/or a single load on a bus or no generator and/or no load. Use 10^{-7} for convergence accuracy.

1. (a) (50pts points) Plain vanilla AC power flow with no reactive power limits enforced. Provide demo case showing results with and without reactive power limit violations.
2. (b) (30pt points) Enforce reactive power limits. Provide demo case showing results with and without reactive power limit violations.
3. (c) (10pts points) Add transformers with fixed tap setting. Provide demo case showing results with different tap settings.
4. (d) (10pts points) Add transformers with fixed phase shift. Provide demo case showing results with different phase shift setting.

Part A

Firstly, I used five cases to test if my AC power flow function works.

```
% load the cases
case4gs = loadcase(case4gs); case9 = loadcase(case9);
case14 = loadcase(case14); case18 = loadcase(case18);
case22 = loadcase(case22);
```

I compared my results of P, Q, V, θ with the runpf() results and they are matched. I think the tiny difference is caused by choosing 10^{-7} for our convergence accuracy.

a1_diffBus_cas...	14x2 double
a1_diffBus_cas...	18x2 double
a1_diffBus_cas...	22x2 double
a1_diffBus_cas...	[0,0;-8.9420e-1...
a1_diffBus_cas...	9x2 double
a1_diffGen_ca...	[4.1198e-07,-1....
a1_diffGen_ca...	[-2.6700e-09,-9...
a1_diffGen_ca...	[-5.0022e-11,-2...
a1_diffGen_ca...	[-4.6855e-07,4....
a1_diffGen_ca...	[-9.6091e-06,-6...

So, my function works. Then, I chose case 9 as my demo case.

When there is no reactive power exceed the limits.

case9_MY.gen

	1	2	3	4	5	6	7	8	9	10
1	1	71.6410	27.0459	300	-300	1.0400	100	1	250	10
2	2	163.0000	6.6536	300	-300	1.0250	100	1	300	10
3	3	85.0000	-10.8597	300	-300	1.0250	100	1	270	10

And the voltages and angles are:

8	9
1.0400	0
1.0250	9.2800
1.0250	4.6648
1.0258	-2.2168
1.0127	-3.6874
1.0324	1.9667
1.0159	0.7275
1.0258	3.7197
0.9956	-3.9888

As we can see, all the reactive power of generators is in the range [-300,300] MVAr.

Now, I put 360 MVAr reactive power to the bus 3 to exceed the maximum reactive power limits of generator 3.

a2_case9_violatedMax_MATnoQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	71.6410	27.0459	300	-300	1.0400	100	1	250	10
2	2	163	6.6537	300	-300	1.0250	100	1	300	10
3	3	85	349.1403	300	-300	1.0250	100	1	270	10

And the voltages and angles are:

a2_case9_violatedMax_MATnoQ.bus

	6	7	8	9
1	0	1	1.0400	0
2	0	1	1.0250	9.2800
3	0	1	1.0250	4.6648
4	0	1	1.0258	-2.2168
5	0	1	1.0127	-3.6874
6	0	1	1.0324	1.9667
7	0	1	1.0159	0.7275
8	0	1	1.0258	3.7197
9	0	1	0.9956	-3.9888

We can see the reactive power at generator 3 is exceeded 300 MVAr, which violate the maximum setting of the generator 3.

And then, I put -360 MVAr reactive power to the bus 3 to exceed the minimum reactive power limits of generator 3.

a3_case9_violatedMin_MYnoQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	71.6410	27.0459	300	-300	1.0400	100	1	250	10
2	2	163.0000	6.6536	300	-300	1.0250	100	1	300	10
3	3	85.0000	-370.8597	300	-300	1.0250	100	1	270	10
4										

And the voltages and angles are:

a3_case9_violatedMin_MYnoQ.bus

	8	9	10	11
1	1.0400	0	345	
2	1.0250	9.2800	345	
3	1.0250	4.6648	345	
4	1.0258	-2.2168	345	
5	1.0127	-3.6874	345	
6	1.0324	1.9667	345	
7	1.0159	0.7275	345	
8	1.0258	3.7197	345	
9	0.9956	-3.9888	345	
10				

We can see the reactive power at generator 3 is less then -300 MVAr, which violate the minimum setting of the generator 3.

Lastly, I used runpf() for both cases, I go the same result.

Part B

In part B, we are supposed to write a function to enforce reactive power limits. If the generator violates the reactive power constrains, we turn the PV bus to the PQ bus where the generator locates and set the generator output at the limits. And then run the power flow again until the generator's constraints are not violated. We assume the slack bus doesn’t have reactive power limits in our case.

Again, I use case 9 as my demo. Firstly, when there no reactive power violation.

b1_case9_MY_noQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	71.6410	27.0459	300	-300	1.0400	100	1	250	10
2	2	163.0000	6.6536	300	-300	1.0250	100	1	300	10
3	3	85.0000	-10.8597	300	-300	1.0250	100	1	270	10
4										

And the voltages and angles are:

b1_case9_MY_noQ.bus

	8	9
1	1.0400	0
2	1.0250	9.2800
3	1.0250	4.6648
4	1.0258	-2.2168
5	1.0127	-3.6874
6	1.0324	1.9667
7	1.0159	0.7275
8	1.0258	3.7197
9	0.9956	-3.9888

I got the same result as I didn't enforce reactive power limits.

Now, I put 360 MVAr reactive power to the bus 3 to exceed the maximum reactive power limits of generator 3 and ran it again.

b2_case9_MY_MaxWithQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	72.7237	61.2455	300	-300	1.0400	100	1	250	10
2	2	163.0000	46.2236	300	-300	1.0250	100	1	300	10
3	3	85.0000	300.0000	300	-300	1.0250	100	1	270	10

And the voltages and angles are:

b2_case9_MY_MaxWithQ.bus

	8	9	10
1	1.0400	0	
2	1.0250	9.9291	
3	0.9045	6.4351	
4	1.0069	-2.2926	
5	0.9671	-3.7227	
6	0.9450	3.0944	
7	0.9640	1.2781	
8	1.0018	4.2351	
9	0.9733	-4.0396	

As we can see, the reactive output of generator 3 is at the upper bound limit and voltage at generator 3 dropped.

Comparing the result with MATPOWER result, they are the same.

b2_case9_MATPOWER_MaxWithQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	72.7237	61.2455	300	-300	1.0400	100	1	250	10
2	2	163	46.2236	300	-300	1.0250	100	1	300	10
3	3	85	300	300	-300	1.0250	100	1	270	10

And then, I put -360 MVAr reactive power to the bus 3 to exceed the minimum reactive power limits of generator 3.

b3_case9_MY_MinWithQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	72.0065	-10.2579	300	-300	1.0400	100	1	250	10
2	2	163.0000	-36.5230	300	-300	1.0250	100	1	300	10
3	3	85.0000	-300.0000	300	-300	1.0250	100	0	270	10

And the voltages and angles are:

b3_case9_MY_MinWithQ.bus

	8	9	10
1	1.0400	0	
2	1.0250	8.5160	
3	1.1578	2.9557	
4	1.0464	-2.1841	
5	1.0623	-3.7541	
6	1.1283	0.7705	
7	1.0726	0.0508	
8	1.0520	3.0946	
9	1.0200	-4.0017	
10			

As we can see, the reactive output of generator 3 is at the lower bound limit and the voltage increased

b3_case9_MATPOWER_MinWithQ.gen

	1	2	3	4	5	6	7	8	9	10
1	1	72.0065	-10.2579	300	-300	1.0400	100	1	250	10
2	2	163	-36.5230	300	-300	1.0250	100	1	300	10
3	3	85	-300	300	-300	1.0250	100	1	270	10
4										

Comparing the result with MATPOWER result, they are the same.

Part C

I got the equation from the MATPOWER manual.

$$Y_{br} = \begin{bmatrix} \left(y_s + j\frac{b_c}{2}\right)\frac{1}{\tau^2} & -y_s\frac{1}{\tau e^{-j\theta_{\text{shift}}}} \\ -y_s\frac{1}{\tau e^{j\theta_{\text{shift}}}} & y_s + j\frac{b_c}{2} \end{bmatrix}.$$

Where τ is the tap ratio and θ is the phase shift. From the equations, we are supposed to divide the square of tap ratio for all the "from" bus and divide tap ratio with the impedance connect to them.

I chose case 14 as my demo case since it contains transformers at the beginning. For my first run, I used the original case 14 and got the bus voltage and angle. The original tap ratio is in the red box.

8	9
1.0600	0
1.0450	-4.9826
1.0100	-12.7251
1.0177	-10.3129
1.0195	-8.7739
1.0700	-14.2209
1.0615	-13.3596
1.0900	-13.3596
1.0559	-14.9385
1.0510	-15.0973
1.0569	-14.7906
1.0552	-15.0756
1.0504	-15.1563
1.0355	-16.0336

```
%   fbus    tbus    r   x   b   rateA   rateB   rateC   ratio   angle   status  a
mpc.branch = [
    1   2   0.01938 0.05917 0.0528  0   0   0   0   0   1   -360    360;
    1   5   0.05403 0.22304 0.0492  0   0   0   0   0   1   -360    360;
    2   3   0.04699 0.19797 0.0438  0   0   0   0   0   1   -360    360;
    2   4   0.05811 0.17632 0.034   0   0   0   0   0   1   -360    360;
    2   5   0.05695 0.17388 0.0346  0   0   0   0   0   1   -360    360;
    3   4   0.06701 0.17103 0.0128  0   0   0   0   0   1   -360    360;
    4   5   0.01335 0.04211 0   0   0   0   0   0   1   -360    360;
    4   7   0   0.20912 0   0   0   0   0.978   0   1   -360    360;
    4   9   0   0.55618 0   0   0   0   0.969   0   1   -360    360;
    5   6   0   0.25202 0   0   0   0   0.932   0   1   -360    360;
    6   11  0.09498 0.1989  0   0   0   0   0   0   1   -360    360;
    6   12  0.12291 0.25581 0   0   0   0   0   0   1   -360    360;
    6   13  0.06615 0.13027 0   0   0   0   0   0   1   -360    360;
    7   8   0   0.17615 0   0   0   0   0   0   1   -360    360;
```

The generator output are:

c1_noTranformer_MY.gen

	1	2	3
1	1	232.3933	-16.5493
2	2	40.0000	43.5571
3	3	-8.5094e...	25.0753
4	6	-1.2988e...	12.7309
5	8	4.4409e-...	17.6235
6			

And then, I modified the case 14 and used different tap setting. I changed branches 4-7 4-9 5-6 6-11 ratio 0.7,0.8,0.9,0.7 and ran it again.

8	9
1.0600	0
1.0450	-5.2782
1.0100	-13.4654
0.9658	-10.1353
0.9829	-8.7629
1.0700	-14.6962
1.2411	-12.3241
1.0900	-12.3241
1.2635	-13.5813
1.3039	-13.8861
1.4111	-14.2928
1.0701	-15.2508
1.0817	-15.3586
1.1685	-15.1125

c2_withTranformer.branch

	1	2	3	4	5	6	7	8	9
1	1	2	0.0194	0.0592	0.0528	0	0	0	0
2	1	5	0.0540	0.2230	0.0492	0	0	0	0
3	2	3	0.0470	0.1980	0.0438	0	0	0	0
4	2	4	0.0581	0.1763	0.0340	0	0	0	0
5	2	5	0.0570	0.1739	0.0346	0	0	0	0
6	3	4	0.0670	0.1710	0.0128	0	0	0	0
7	4	5	0.0134	0.0421	0	0	0	0	0
8	4	7	0	0.2091	0	0	0	0	0.7000
9	4	9	0	0.5562	0	0	0	0	0.8000
10	5	6	0	0.2520	0	0	0	0	0.9000
11	6	11	0.0950	0.1989	0	0	0	0	0.7000
12	6	12	0.1229	0.2558	0	0	0	0	0
13	6	13	0.0662	0.1303	0	0	0	0	0
14	7	8	0	0.1762	0	0	0	0	0
15	7	9	0	0.1100	0	0	0	0	0
16	9	10	0.0318	0.0845	0	0	0	0	0
17	9	14	0.1271	0.2704	0	0	0	0	0
18	10	11	0.0821	0.1921	0	0	0	0	0
19	12	13	0.2209	0.1999	0	0	0	0	0
20	13	14	0.1709	0.3480	0	0	0	0	0

The generator output are:

c2_withTransformer_MY.gen

	1	2	3
1	1	242.6786	-1.7433
2	2	40.0000	97.2955
3	3	1.7494e-...	56.1137
4	6	1.7007e-...	63.6843
5	8	4.4409e-...	-93.5287

Lastly, I compared my data with the MATPOWER to make sure everything is correct.

```
% use MATPOWER function to make sure the answer is correct
c2_withTransformer_MAT = runpf(c2_withTranformer);
c2_withTransformer_MY = ACPF_Parta(c2_withTranformer);
c2_diffBus_withTransformer = c2_withTransformer_MY.bus(:,VTheta_bus) - ...
    c2_withTransformer_MAT.bus(:,VTheta_bus);
c2_diffGen_withTransformer = c2_withTransformer_MY.gen(:,PQ_gen) - ...
    c2_withTransformer_MAT.gen(:,PQ_gen);
c2_Ybus_MY = findYbus(c2_withTranformer);
c2_Ybus_MATPOWER = full(makeYbus(c2_withTranformer));
c2_diffY_withTransformer = c2_Ybus_MY - c2_Ybus_MATPOWER;
```

Part D

In part D, we are supposed to implement the phase shift of transformers.

$$Y_{br} = \begin{bmatrix} \left(y_s + j\frac{b_c}{2}\right)\frac{1}{\tau^2} & -y_s\frac{1}{\tau e^{-j\theta_{\text{shift}}}} \\ -y_s\frac{1}{\tau e^{j\theta_{\text{shift}}}} & y_s + j\frac{b_c}{2} \end{bmatrix}.$$

From the equation, we can see for the “from” to “to” bus we use a negative sign for the phase shift and “to” to “from” bus we use positive phase shift. And the phase shift in the mpc file is in degrees.

I used the mpc I modified previously to get the results for my initial ran. And phase shifts are all 0 degree and I got the bus voltage and angle

8	9
1.0600	0
1.0450	-5.2782
1.0100	-13.4654
0.9658	-10.1353
0.9829	-8.7629
1.0700	-14.6962
1.2411	-12.3241
1.0900	-12.3241
1.2635	-13.5813
1.3039	-13.8861
1.4111	-14.2928
1.0701	-15.2508
1.0817	-15.3586
1.1685	-15.1125

c2_withTransformer_MY.branch

	1	2	3	4	5	6	7	8	9	10	11
1	1	2	0.0194	0.0592	0.0528	0	0	0	0	0	1
2	1	5	0.0540	0.2230	0.0492	0	0	0	0	0	1
3	2	3	0.0470	0.1980	0.0438	0	0	0	0	0	1
4	2	4	0.0581	0.1763	0.0340	0	0	0	0	0	1
5	2	5	0.0570	0.1739	0.0346	0	0	0	0	0	1
6	3	4	0.0670	0.1710	0.0128	0	0	0	0	0	1
7	4	5	0.0134	0.0421	0	0	0	0	0	0	1
8	4	7	0	0.2091	0	0	0	0	0.7000	0	1
9	4	9	0	0.5562	0	0	0	0	0.8000	0	1
10	5	6	0	0.2520	0	0	0	0	0.9000	0	1
11	6	11	0.0950	0.1989	0	0	0	0	0.7000	0	1
12	6	12	0.1229	0.2558	0	0	0	0	0	0	1
13	6	13	0.0662	0.1303	0	0	0	0	0	0	1
14	7	8	0	0.1762	0	0	0	0	0	0	1
15	7	9	0	0.1100	0	0	0	0	0	0	1
16	9	10	0.0318	0.0845	0	0	0	0	0	0	1
17	9	14	0.1271	0.2704	0	0	0	0	0	0	1
18	10	11	0.0821	0.1921	0	0	0	0	0	0	1
19	12	13	0.2209	0.1999	0	0	0	0	0	0	1
20	13	14	0.1709	0.3480	0	0	0	0	0	0	1

The generator output are:

c2_withTransformer_MY.gen

	1	2	3
1	1	242.6786	-1.7433
2	2	40.0000	97.2955
3	3	1.7494e-...	56.1137
4	6	1.7007e-...	63.6843
5	8	4.4409e-...	-93.5287

And then, I added phase shift for all transformers, I changed the phase shift of branch 4-7 4-9 5-6 6-11 to 10,20,10,10 degrees. And ran it again.

8	9
1.0600	0
1.0450	-5.3506
1.0100	-13.6427
0.9639	-10.3643
0.9827	-8.7705
1.0700	-23.5243
1.2382	-25.2790
1.0900	-25.2790
1.2618	-28.0980
1.3027	-29.2769
1.4104	-31.4931
1.0709	-24.5470
1.0800	-25.1251
1.1661	-27.7447

d2_withTransformer_MY_Phase.branch

	1	2	3	4	5	6	7	8	9	10	11
1	1	2	0.0194	0.0592	0.0528	0	0	0	0	0	
2	1	5	0.0540	0.2230	0.0492	0	0	0	0	0	
3	2	3	0.0470	0.1980	0.0438	0	0	0	0	0	
4	2	4	0.0581	0.1763	0.0340	0	0	0	0	0	
5	2	5	0.0570	0.1739	0.0346	0	0	0	0	0	
6	3	4	0.0670	0.1710	0.0128	0	0	0	0	0	
7	4	5	0.0134	0.0421	0	0	0	0	0	0	
8	4	7	0	0.2091	0	0	0	0	0.7000	10	
9	4	9	0	0.5562	0	0	0	0	0.8000	20	
10	5	6	0	0.2520	0	0	0	0	0.9000	10	
11	6	11	0.0950	0.1989	0	0	0	0	0.7000	10	
12	6	12	0.1229	0.2558	0	0	0	0	0	0	
13	6	13	0.0662	0.1303	0	0	0	0	0	0	
14	7	8	0	0.1762	0	0	0	0	0	0	
15	7	9	0	0.1100	0	0	0	0	0	0	
16	9	10	0.0318	0.0845	0	0	0	0	0	0	
17	9	14	0.1271	0.2704	0	0	0	0	0	0	
18	10	11	0.0821	0.1921	0	0	0	0	0	0	
19	12	13	0.2209	0.1999	0	0	0	0	0	0	
20	13	14	0.1709	0.3480	0	0	0	0	0	0	
21											

The generator output are:

d2_withTransformer_MY_Phase.gen

	1	2	3
1	1	244.9385	-2.1469
2	2	40.0000	99.1122
3	3	2.5156e-...	57.2369
4	6	9.6542e-...	69.6869
5	8	0	-91.6745

Lastly, I compared my results with MATPOWER results to make sure they are all correct.

```
% use MATPOWER function to make sure the answer is correct
d2_withTransformer_MAT_Phase = runpf(d1_withTranformer_Phase);
d2_withTransformer_MY_Phase = ACPF_Parta(d1_withTranformer_Phase);
d2_diffBus_withTransformer_phase = d2_withTransformer_MY_Phase.bus(:,VTheta_bus) - ...
    d2_withTransformer_MAT_Phase.bus(:,VTheta_bus);
d2_diffGen_withTransformer = d2_withTransformer_MY_Phase.gen(:,PQ_gen) - ...
    d2_withTransformer_MAT_Phase.gen(:,PQ_gen);
d2_Ybus_MY_Phase = findYbus(d1_withTranformer_Phase);
d2_Ybus_MATPOWER_Phase = full(makeYbus(d1_withTranformer_Phase));
d2_diffY_withTransformer = d2_Ybus_MY_Phase - d2_Ybus_MATPOWER_Phase;
```

I felt very happy and rewarding to implement AC power flow by myself (;