

Procedure for Dynamic Simulation with Fault using PSS/E v33

This procedure is for power system stability simulations using PSS/E (version 33 is used as example, as well a standard 2-area 6-bus 5-branch test system).

1. Import raw file
File -> Import -> Power Flow Raw Option,
Select 33 as the Raw file version,
Browse the system raw file and load it.
2. Get initial condition by solving power flow
Power Flow -> Solution -> Solve (NSOL/FNSL/...)
You can either use the default methods/settings or make some changes if you want.
Only if the power flow converges, the dynamic simulation can continue...
3. Import dyr file
File -> Open,
Select the associated dynamic file (.dyr file).
4. Setup the output variables
Channel file is used to save the results. PSS/E will write the dynamic simulation results to this channel file.
Dynamics -> Channel Setup Wizard,
change parameters for output if needed.
5. Setup dynamic simulation options
Dynamics -> Simulation options,
E.g. “Set relative machine angles”, “Relative to machine”, click “Select”, choose the machine needed. When you are done, click “OK”.
6. Initialize the channel file
Dynamics -> Simulation -> Perform Simulation (STRT/RUN)...
Set “% Constant admittance” to 100 and “% constant current” to 0, as shown in Fig. 1;
then, click “OK”,
Initialize a channel file by clicking block “...” and naming it, then click “Initialize”, and then click “Close”.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpglab.github.io/resources>

Email: xplipower@gmail.com | xli83@central.uh.edu

Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

7. Simulate the pre-fault steady-state system condition for 1s.

Dynamics -> Simulation -> Perform Simulation (STRT/RUN),

Run to “1”, Print every “10”, Write every “10”, Plot every “10”, for the last 3 parameters, change them if needed.

Then, click Run, and then click “Close”.

8. Apply fault

Disturbance -> Apply a bus fault (you can select other fault type as needed),

Click “Select”, double click “bus 5”, click “OK”,

Click “OK”.

9. Simulate the under-fault system condition

Simulate this under-fault situation for some time Δt , try and error method can be used to identify the critical clearing time.

Dynamics -> Simulation -> Perform Simulation (STRT/RUN),

Run to “1.1”, Print every “1”, Write every “1”, Plot every “1”, for the last 3 parameters, change them if needed.

Then, click Run, and then click “Close”.

10. Clear fault

Disturbance -> Clear Fault,

Select “Bus 5”, click “Go”.

11. Simulate the post-fault system condition.

Dynamics -> Simulation -> Perform Simulation (STRT/RUN),

Run to “6.0”, Print every “10”, Write every “10”, Plot every “10”, for the last 3 parameters, change them if needed.

Then, click “Run”, click “Close”.

12. Plot the relative machine angle curves

File -> New -> Plot Book,

File -> Open, select the channel created in Step 6,

Make sure View -> Plot Tree View is selected. Then, switch to Plot Tree View, as shown in Fig. 2.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpplab.github.io/resources>

Email: xlipower@gmail.com | xli83@central.uh.edu

Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

Plot Data -> Channel Files -> results, then drag the machine angles with flags such as “ANGL 1”, “ANGL 2” to the PlotBook, as shown in Fig. 3.

13. Repeat the process with different clearing times, until critical clearing time is determined.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpplab.github.io/resources>
Email: xplipower@gmail.com | xli83@central.uh.edu
Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

Tip

A very efficient and helpful tip if we have to do this same process but with different clearing time again and again is to use the I/O control recorder functionality:

- Open PSS/E software,
- I/O Control -> Start recording, save as Python Program file (e.g., .py file),
- Perform Step 1 to Step 12,
- I/O Control -> Stop recording.

Then, open the Python Program file, just change the fault clearing time and then,

- I/O Control -> Run program Automation file, select the same file, done!

Note that each time you repeat this simulation, you may have to close PSS/E and reopen it in order to run another simulation with different fault clearing time; also the files location have to be the same (or you need to change the code in the file). Sample python codes are given in the end.

Also Note that the Xd” for each generator in the dynamic file has to be the same value as the imagine part of Z source impedance (ZX) for the associated generator in the raw file.

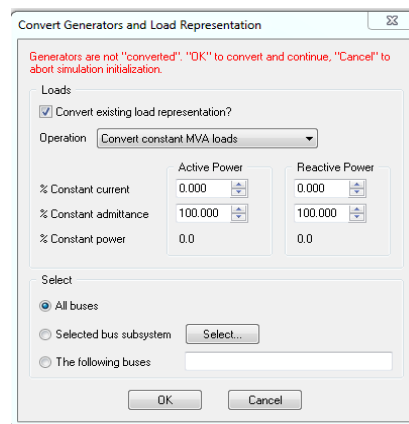


Fig. 1. Constant admittance model for Load Representation.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpplab.github.io/resources>
Email: xplipower@gmail.com | xli83@central.uh.edu
Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

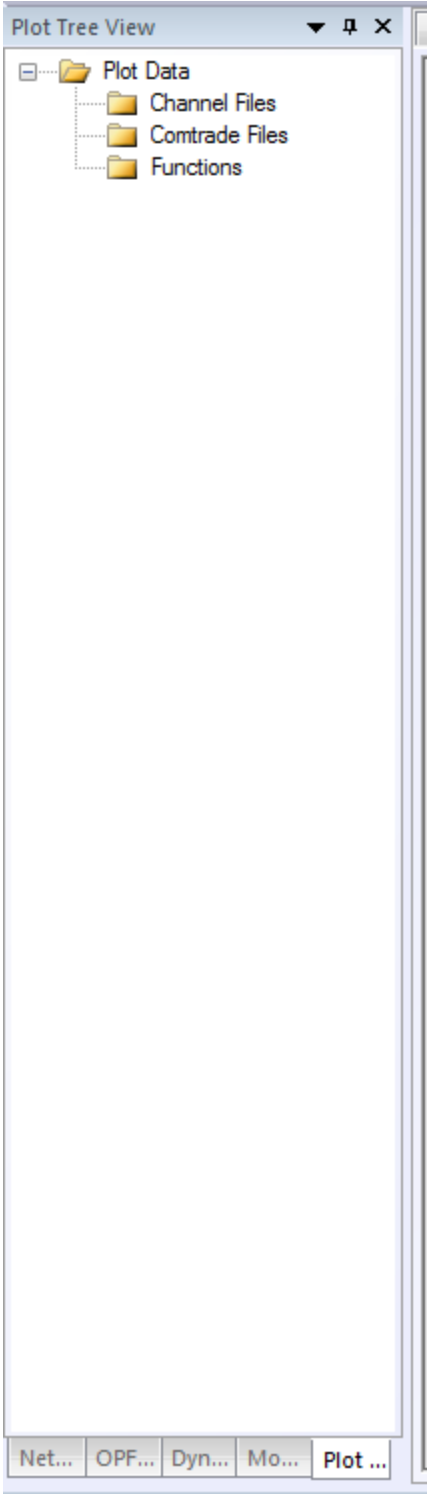


Fig. 2. Plot Tree View.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpplab.github.io/resources>
Email: xplipower@gmail.com | xli83@central.uh.edu
Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

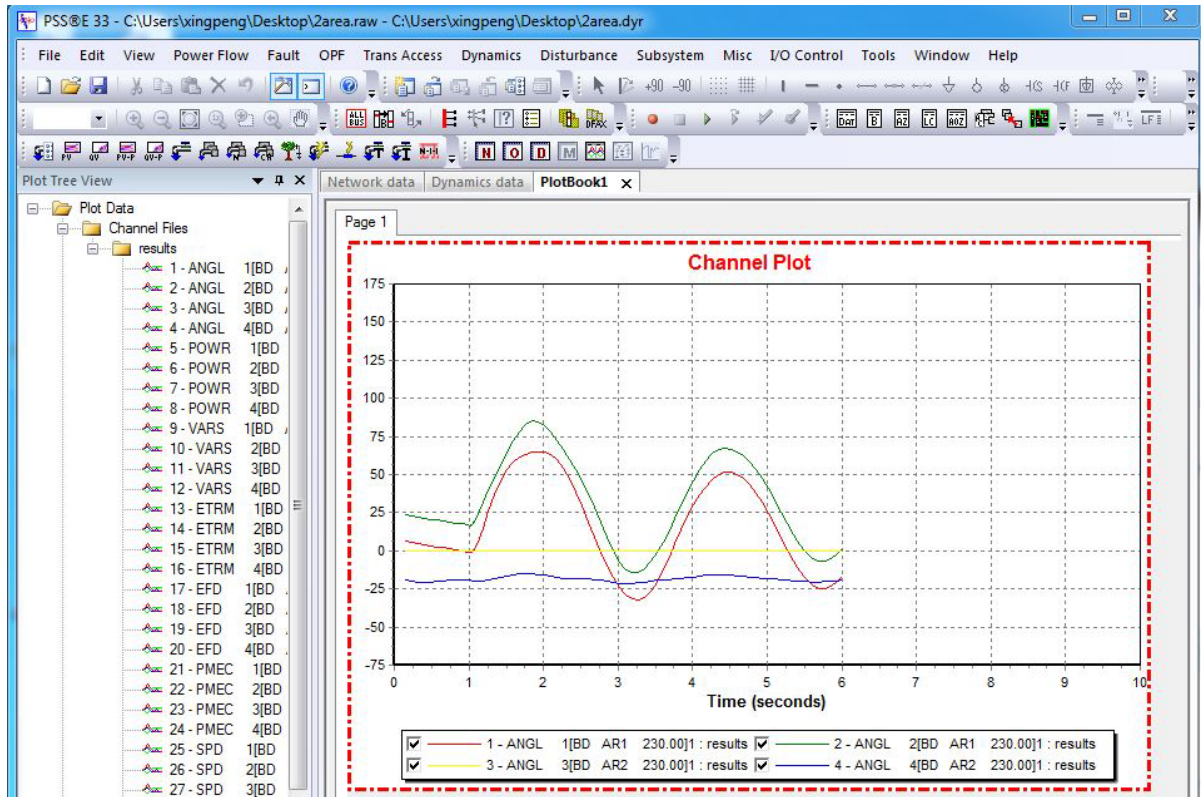


Fig. 3. Relative machine angle curves.

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpglab.github.io/resources>
Email: xplipower@gmail.com | xli83@central.uh.edu
Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

Sample python code:

```
# File:"C:\Users\xingpeng\Desktop\sss.py", generated on SAT, xxx 16:52, release 33.05.00
psspy.readrawversion(0,r""33""",r""C:\Users\xingpeng\Desktop\2area.raw""")
psspy.fns1([0,0,0,1,1,0,99,0])
psspy.dyre_new([1,1,1,1],r""C:\Users\xingpeng\Desktop\2area.dyr""", "", "", "")
psspy.chsb(0,1,[-1,-1,-1,1,1,0])
psspy.chsb(0,1,[-1,-1,-1,1,2,0])
psspy.chsb(0,1,[-1,-1,-1,1,3,0])
psspy.chsb(0,1,[-1,-1,-1,1,4,0])
psspy.chsb(0,1,[-1,-1,-1,1,5,0])
psspy.chsb(0,1,[-1,-1,-1,1,6,0])
psspy.chsb(0,1,[-1,-1,-1,1,7,0])
psspy.chsb(0,1,[-1,-1,-1,1,13,0])
psspy.chsb(0,1,[-1,-1,-1,1,16,0])
psspy.set_relang(1,3,r""1""")
psspy.cong(0)
psspy.conl(0,1,1,[0,0],[0.0, 100.0,0.0, 100.0])
psspy.conl(0,1,2,[0,0],[0.0, 100.0,0.0, 100.0])
psspy.conl(0,1,3,[0,0],[0.0, 100.0,0.0, 100.0])
psspy.strt(0,r""C:\Users\xingpeng\Desktop\results.out""")
psspy.change_channel_out_file(r""C:\Users\xingpeng\Desktop\results.out""")
psspy.run(0, 1.0,10,10,10)
psspy.dist_bus_fault(5,1, 230.0,[0.0,-0.2E+10])
psspy.change_channel_out_file(r""C:\Users\xingpeng\Desktop\results.out""")
psspy.run(0, 1.083,10,10,10)
psspy.dist_clear_fault(1)
psspy.change_channel_out_file(r""C:\Users\xingpeng\Desktop\results.out""")
psspy.run(0, 6.0,10,10,10)
pssplot.newplotbook()
pssplot.insertpage()
```

Author: Xingpeng Li, Assistant Professor, University of Houston. Link: <https://rpplab.github.io/resources>
Email: xlipower@gmail.com | xli83@central.uh.edu
Log: this file was first created on Nov. 28th, 2015 @ArizonaStateUniversity – documented for class projects for EEE575 Power System Stability. It was then further cleanup up and slightly modified on March 22, 2023.

```
pssplot.setselectedpage(0)
pssplot.openchandatafile(r"C:\Users\xingpeng\Desktop\results.out")
pssplot.dragdropplotdata(r"results",r"1 - ANGL 1[BD AR1 230.00]1")
pssplot.dragdropplotdata(r"results",r"2 - ANGL 2[BD AR1 230.00]1")
pssplot.dragdropplotdata(r"results",r"3 - ANGL 3[BD AR2 230.00]1")
pssplot.dragdropplotdata(r"results",r"4 - ANGL 4[BD AR2 230.00]1")
```

Disclaimer:

The author doesn't make any warranty for the accuracy, completeness, or usefulness of any information disclosed and the author assumes no liability or responsibility for any errors or omissions for the information (data/code/results etc) disclosed.