Setting up environment for installation

This document describes the steps to run BFT simulation in your computer

The program requires Python 3.7.3 or higher version to be installed in your computer
If you are not sure please install Python .

get help from https://docs.python.org/3/using/index.html

The program also requires Python compatible plotting library to show graphs. For this you need to install matplotlib. Depending on the system, the required instruction in install matplotlib can be found here https://matplotlib.org/

The program is tested in windows environment but with the right installation should be able to run in any platform. The plotting library is however device dependent. So some variation of behavior can be expected in different environments. As it is not possible to test the software in every environment, I can't guarantee that the simulated graphs will be rendered in the same way in every platform. It is advisable to run them in Windows 10 for the expected behavior.

How to run the simulation

Go to the directory where your unzipped files are. You should be able to locate a python file named bft_simulation.py

Make sure it has executable privilege. Now run the python code with following command

python bft_simulation.py

You should be able to watch six screens with six different simulation one after another.

How to write code for new simulation

Also although it is not necessary but possible to write your own simulation by creating a different environment. Here is how you can do it.

In main module of the python code you can instantiate a substation with following code

```
st = Substation(0)
substation_list.append(st)
```

You can create a non faulty PMU and add it to the substation network

```
pmu = st.add_pmu(signal_generator)
```

To create a faulty pmu you need to pass the second parameter as non zero

You can create a non faulty PMU and add it to the substation network

pmu = st.add_pmu(signal_generator,1)

You can create a PDC as below and add it to the substation network

pdc = st.add_pdc()

You can mark the PDC as faulty with following API

pdc.set_faulty(1)

Now as the environment and network is all set you can run a simulation by transmitting signal from any PMU by the following command for a specified duration.

pmu.start_transmit(st,duration)