Hi Jacob,

I was able to run the simulation from commit [a065b1a](https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fsendecomp%2FKoosha-Interdependency%2Fcommit%2Fa065b1a0422b03d521998baaedc3eaa46ba5379a&data=05%7C01%7Cslbdb%40mst.edu%7C976600ea85dc419be08708da691d7bae%7Ce3fefdbef7e9401ba51a355e01b05a89%7C0%7C0%7C637937872609384520%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=%2FI9rqdr8KHeaO93vS9Ivly%2F0q8dfYOqM0kJ88D7sSzU%3D&reserved=0) (my latest commit on the repo). I had to make the following changes in order to make the code work:

1. changed all instances of ['str1', 'str2'] to append('str1', 'str2') in cascade.m and runpsat.m. I noticed that you have addressed this issue in a later commit. This issue exists due to a change from MATLAB 2019 to MATLAB 2020. I also noticed that the latest PSAT (which you are using) supports MATLAB 2019a (and not 2022a). See the following screenshot from Dr. Milano's website:  
   A screenshot of a computer

   AI-generated content may be incorrect.
2. added the following lines to @PMclass/subsref.m:  
   A screen shot of a computer code

   AI-generated content may be incorrect.  
   I noticed that you have also mentioned this change in your README docx document. I am not sure why store attribute is not substituted between the two objects, but I don't see any problem with making this change in the code.

For my test, I ran the following lines in the command line:

i = initial\_failures(20, 3, 3, 1, 1, 1, 1, 0, 1)

cascade("ieee14\_smart", i)

Unlike what you have mentioned in your document, I'm using ieee14\_smart.m (instead of ieee14.m). You can open these two files in data folder and see that unlike ieee14.m, ieee14\_smart.m includes 3 SSSC devices (FACTS) and 3 PMU devices. You will definitely need the \_smart version (smart grid as opposed to traditional power grid with no cyber components) in order to inject your cyber faults.

I did not inspect your branch with GPS faults, but at least from the function call, I can see that some of your input arguments for initial\_failures may be wrong. nl is the number of transmission lines and for IEEE-14 this number is 20. The number 210 that you captured from my paper reflects the total number of line failures including single-line and dual-line failures (20 choose 2 + 20 choose 1 = 190 + 20 = 210). Therefore, the first input argument needs to be corrected.

In my simulations, I did not account for the failure of communication links since my simulation environment did not properly capture the nature of such failures. So, I do not have any confidence in the sanity of the code if you specify mc > 0 (mc is the "maximum number of concurrent communication failures").

I thought the simplest approach would be to start from the last known "working code" and then get to your branch. Please review these comments and see if you can run the code from my latest commit. Once you do that successfully, check to see if your code with the additional changes also works. You can contact me any time if you have any other questions. My availability is mostly towards the end of the week, so please expect some delay.

Best regards,

Koosha



Hi Jacob,

The try-catch structure is there only to let the code fail gracefully and continue the program in cases where PSAT cannot complete the PF simulation. These errors are indeed the power grid cascading failure cases that we are looking for.

You are getting **Unrecognized field name "sv"** error because there are still some issues with the simulation framework. By removing the try-catch, you will start seeing other errors that were previously handled. To sum it up, you don't need to trace the **Array indicices must be positive integers or logical values** error. Instead, you need to figure out why you are getting **Unrecognized field name "sv"** error.

I recommend that you print the values that give you these errors and check the logs (from er in the catch, or just above where you are getting errors) to see what's wrong with the data that is passed to these statements. cascade.m also prints a log in **failure\_seq folder/<FILE\_NAME>.txt**. I have attached the log that I am getting when I run ieee14\_smart. You can compare this with yours to at least see where your code stops.