Brief introduction to
State Estimation (SE) program

Rui Bo

1. State Estimation (SE) program

This program intends to enhance the state estimation code included in MATPOWER (in sub directory 'extras\state_estimator') by defining a generic interface, adding system observability check, and rewriting some of the codes.

Book "Computational Methods for Electric Power Systems" authored by Mariesa Crow is referenced during implementation of the programs, and the notations are in accordance with those in the book.

Please note that execution of the program requires the MATPOWER package is included in the searching path of MATLAB environment.

The author would like to thank Fangxing Li for his help in developing the program. One of his courses taught at the University of Tennessee, Knoxville, "Computational Methods for Power System Analysis", has been a valuable source.

1.1. Features

- ✓ A generic interface of the state estimation is defined
- ✓ 8 types of measurements are supported including real power line flow through from end, real power line flow through to end, generator real power output, voltage angle, reactive power line flow through from end, reactive power line flow through to end, generator reactive power output, voltage magnitude (shortened for PF, PT, PG, Va, QF, QT, QG, Vm respectively)
- ✓ Measurement variances for each of the 8 kinds of measurements could be separately specified.
- ✓ Input data integrity check is provided such as completeness and consistency check
- ✓ System observability check is provided and possible reasons are identified, such as not-in-observation variables, dependent variables, etc. This type of information can guide users in changing measurement settings to make system observable.
- ✓ Source codes are self-explanatory
- ✓ Sufficient comments are provided in the source codes

1.2. Limitations

- ✓ Presently bad data detection is not implemented yet
- ✓ The measurement indices of the interface may be further improved to facilitate the use of the program

1.3. Demonstration

Running 'test_se.m' will generate the following outputs. The test case is a 6 bus system, which is included in the state estimation program package.

```
================================================================================
|     Comparison of measurements and their estimations                         |
|     NOTE: In the order of PF, PT, PG, Va, QF, QT, QG, Vm (if applicable)     |
================================================================================
    Type         Index       Measurement    Estimation
                  (#)           (pu)           (pu)
 -----------  -----------   -----------    ----------
      PF           1           0.1200         0.1474
      PF           2           0.1000         0.0783
      PT           3          -0.0400        -0.0399
      PG           1           0.5800         0.5757
      PG           2           0.3000         0.3034
      PG           3           0.1400         0.1336
      Vm           2           1.0400         1.0258
      Vm           3           0.9800         0.9790

[Weighted sum of error squares]:    5.417915
```

Running 'test_se_14bus_err.m' will generate the following outputs. The program identifies some possible reasons for system being not observable, which are not-in-observation variables. This type of information can guide users in changing measurement settings to make system observable.

```
>> test_se_14bus_err
Warning: The following variables are not observable since they are not related with any measurement!
varNames =

    'Va8'    'Va10'    'Va14'    'Vm8'    'Vm10'    'Vm14'


idx_trivialColumns =

     4     9    13    17    22    26

??? Error using ==> doSE at 134
doSE: system is not observable

Error in ==> run_se at 37
[V, success, iterNum, z, z_est, error_sqrsum] = doSE(baseMVA, bus, gen, branch, Ybus, Yf, Yt, V0, ref,

Error in ==> test_se_14bus_err at 54
[baseMVA, bus, gen, branch, success, et, z, z_est, error_sqrsum] = run_se(casename, measure, idx, sigma
```

2. File Structure

The file structure is as follows.

1) Source code files.
   run_se.m: run state estimation
   doSE.m: perform state estimation
   outputsesoln.m: show state estimation results on screen
   checkDataIntegrity.m: check input data integrity
   isobservable.m: check if system is observable through provided measurements
2) Case data files.
   case3bus_P6_6.m: a 3 bus system in problem 6.6 in book 'Computational Methods for Electric Power Systems' authored by Mariesa Crow
3) Test files.
   test_se.m: test state estimation solver on a 3-bus system
   test_se_14bus.m: test state estimation solver on the IEEE 14-bus system

test_se_14bus_err.m: test state estimation solver on the IEEE 14-bus system with measurement issues leading to system not observable