Toolbox for Power System Computing - TPSC

I. Next Steps (Instead of Introduction)

- Power flow - check and documentation.
- Device distribution - check and documentations
- Use information if a generator is in-service or out-of-service.

II. Data

A. General Data of Power Systems

The power system is represented as a structure within the Matlab software package. When loading a power system, the variable of structural type named "data" is loaded into the workspace. The names and descriptions of the structure's fields are displayed in Table I.

Table I
FIELDS OF THE STRUCTURE USED TO DESCRIBE THE POWER SYSTEM (*data*).

Field	Description
data.nBuses	number of nodes
data.fn	nominal system frequency
data.nBranches	number of branches
data.nGens	number of generators
data.slackNo	number of a slack bus
data.baseMVA	base power
data.bus	matrix whose rows represent groups of information used to describe nodes in the system
data.branch	matrix whose rows represent groups of information used to describe branches in the system.
data.generator	matrix whose rows represent groups of information used to describe generators in the system.
data.gencost	matrix whose rows represent groups of information used to describe generators in the system from an economic perspective

The matrix containing information about the nodes of the corresponding power system consists of fifteen columns. The column number, description, and unit of the quantitiy are shown in Table II.

Table II
BUS DATA (*data.bus*).

Column	Description	Unit
1	bus number	
2	bus type (1=PQ, 2=PV, 3=slack)	
3	active power demand	p.u.
4	reactive power demand	p.u.
5	active power of shunt conductance at the voltage 1 [p.u.]	p.u.
6	reactive power of shunt conductance at the voltage 1 [p.u.]	p.u.
7	area number	
8	voltage magnitude	p.u.
9	voltage angle	rad
10	base voltage	kV
11	loss zone	
12	maximum voltage magnitude	p.u.
13	minimum voltage magnitude	p.u.
14	active power generation	p.u.
15	reactive power generation	p.u.

The matrix containing information about the branches of the corresponding power system consists of thirteen columns. The column number, description, and unit of the quantitiy are shown in Table III.

Table III
BRANCH DATA (*data.branch*).

Column	Description	Unit
1	from bus	
2	to bus	
3	resistance	p.u.
4	reactance	p.u.
5	total line susceptance	p.u.
6	long term rating (0=unlimited)	p.u.
7	short term rating (0=unlimited)	p.u.
8	emergency rating (0=unlimited)	p.u.
9	transformer off nominal turns, taps at "from bus", pi equivalent at "to bus"; value 0 indicates transimission line rather that transformer	p.u.
10	transformer phase shift angle (positive $\Rightarrow$ delay)	rad
11	branch status (0=in service, 1=out of service)	
12	minimum bus voltage angle difference	rad
13	maximum bus voltage angle difference	rad

The matrix containing information about the generators of the corresponding power system consists of ten columns. The column number, description, and unit of the quantitiy are shown in Table IV.

Table IV
GENERATOR DATA (*data.generator*).

Column	Description	Unit
1	bus number	
2	active power output	p.u.
3	reactive power output	p.u.
4	maximum reactive power output	p.u.
5	minimum reactive power output	p.u.
6	regulated voltage magnitude	p.u.
7	base power of machine	p.u.
8	machine status (0=in service, 1=out of service)	
9	maximum active power output	p.u.
10	minimum active power output	p.u.

Several power systems are already available in the directory ’data/power_systems’. To define a new power system, a user has to write a script with the structure and fields as previously defined. In the case that a user wants to use one of the cases available in the Matpower package, the script "data/matpower2tpsc" can be employed. For example, a user needs to place a Matpower case ’caseName.m’ in the current working directory or in one of the directories defined in a list of Matlab search paths. Subsequently, the power system will be saved in the ’data/power_systems’ directory, and the prefix "TPSC" will be added to specify that this structure is part of the TPSC package. However, users do not have to worry about this because the only requirement in every routine is to enter a ’caseName’.

B. Extended Verison with Measurement Devices

This is a structure that extends the structure described in the previous subsection with data about measurement devices installed in a power system. Structure *data* is now enriched with two new fields:

data.pmu data about phasore measurement units installed in a power system, and it is presented in detail in Table V.
data.scada data about conventional SCADA measurements, and it is presented in detail in Table VI.

Table V
PMUS DATA (*data.pmu*).

Column	Description	Unit
1	bus number	
2	number of current channels, -1 for all incident branches	
3	magnitude measurement standard deviation	%
4	phase angle measurement standard deviation	rad
5	frequency measurement standard deviation	Hz
6	RoCoF measurement standard deviation	Hz/s
7	Reporting frequency	Hz

Table VI
SCADA DATA (*data.scada*).

Column	Description	Unit
1	bus number	
2	measurement type (1-active power flow, 2-reactive power flow, 3-active power injection, 4-rective power injection, 5 - branch current magnitude, 6- bus voltage manitude	
3	line number (type 1, 2 and 6) or bus number (type 3, 4 and 5)[1]	
4	standard deviation	p.u.
5	Reporting frequency	Hz

Similarly to power system data (structured as 'data'), users can input information about power system measurement devices while adhering to the format outlined in Tables V and VI.

On the other hand, a recommended approach involves utilizing the 'run_device_distribution.m' script. This script offers various options for configuring measurement devices, considering their type, standard deviation, and reporting frequency (important for tracking and dynamic modes). Detailed information is available in the Section about routines.

C. *Measurement Data*

In this subsection, the data structure of the measurements generated by the power system toolbox will be defined. The measurements are generated by running power flows and perturbing the true quantities with random Gaussian noise. As defined in the previous subsection, two types of distinguishable measurement types are supported, namely legacy (SCADA) measurements and measurements from PMUs. The primary user of the generated measurement values are different state estimation algorithms. In this regard, the measurements can be generated in a static or tracking (dynamic) manner. In the 'tracking' mode, measurements are taken over a specific time period, while in the 'static' mode, measurements are taken for a single, distinct moment in time.

Measurements from PMUs are separeted in two different tables: synchrophasor measurements VII, and frequency and rocof measurements VIII.

Table VII
PMU MEASUREMENTS (*measurements.synpmu*).

Column	Description	Unit
1	time index	
2	bus number	
3	phasor measurement type (1-current flow, 2-current injection, 3-voltage)	
4	line number (type 1) or bus number (type 2 or 3)	
5	magnitude measurement value	p.u.
6	phase angle measurement value	rad
7	magnitude exact value	p.u.
8	phase angle exact value	rad

Measurements from a SCADA system are collected in a matrix whose columns are described in Table below.

[1] Line number corresponds to a row number of the given line in *data.branches* matrix. When a measurement device is on a "to" side of a line, its number is given with a minus sign.

Table VIII
PMU MEASUREMENTS (*measurements.fpmu*).

Column	Description	Unit
1	time index	
2	bus number	
3	frequency measurement	Hz
4	frequency exact value	Hz
5	rate of change of frequency (rocof) measurement	Hz/s
6	rate of change of frequency (rocof) exact value	Hz/s

Table IX
SCADA MEASUREMENTS (*measurements.scada*).

Column	Description	Unit
1	time index	
2	bus number	
3	measurement type	
4	line number (type 1, 2 and 6) or bus number (type 3, 4 and 5)	
5	measurement value	p.u.
6	exact value	p.u.

III. Routines

A. Power Flows

1) Basics: Power flows, also known as load flow analysis, are fundamental calculations in electrical power systems engineering. They involve solving a set of equations to determine the steady-state operating conditions of an electrical network. The goal of power flow computations is to find the voltages, currents, and power flows at various nodes and branches of the network under given load and generation conditions.

Key aspects of power flow computations include:

1) **Voltage Magnitudes and Angles**: Power flow calculations determine the magnitude and phase angle of voltages at each node in the network. These values indicate the electrical potential and phase relationship at different points in the system.
2) **Active and Reactive Power Flows**: Power flow computations calculate the active (real) and reactive power flows in each transmission line, transformer, and generator. This information is essential for maintaining the proper balance between generation and consumption in the system.
3) **Load Balancing**: Power flow analysis ensures that the supply of power from generators matches the demand from loads, resulting in a balanced system. Deviations from this balance can lead to voltage instability and other issues.
4) **Voltage Stability Assessment**: Power flow computations help assess voltage stability by identifying voltage limits that should not be exceeded to prevent voltage collapse.
5) **Contingency Analysis**: Power flow analysis can be extended to evaluate the impact of potential network failures, such as line outages or equipment failures, to understand their effects on power flows and system stability.
6) **Generator Dispatch**: Power flow calculations assist in determining the optimal settings for generator output to minimize losses and improve overall system efficiency.
7) **Optimal Power Flow (OPF)**: Advanced versions of power flow analysis, such as optimal power flow, optimize the generator outputs and other control variables to meet various objectives while satisfying network constraints.

Power flow computations are critical for planning, designing, and operating power systems effectively and reliably. They help ensure that the system operates within acceptable limits, avoids overloads, maintains voltage stability, and meets load demands while efficiently utilizing available generation resources.

2) Run Settings: To run a power flow analysis in TPSC package a user is provided with the function:

```
[ results ] = run_power_flows(pfsettings, data);
```

Function arguments are structures: 'pfsettings' whose fileds are user-defined and used to set the computations, and 'data', which presents power system data from Table I. The function's output 'results' is already described in the section about data.

'pfsettings' contains the following fields:

1) **domain:** a field used to determain whether the computations will be done in 'complex' or 'real' domain. In power flow analysis, state variables are complex nodal voltages. The classical approach involves presenting complex numbers in either their polar or rectangular forms and conducting computations in the real domain. A recently introduced alternative is to utilize Wirtinger calculus and perform calculations with complex numbers.
2) **method:** used to specify the method for conducting power flow computations.
3) **start:** used to select whether the inital values of state variables will be set to $1\angle 0$[p.u.], or they will be assigned predefiend values (columns 8 and 9 from data.bus).
4) **maxNumberOfIter**: used to select the maximum number of iterations that can be conducted before convergence. If this limit is reached, TPSC notifies the user that the power flow analysis did not converge.
5) **eps**: tolerance value when checking convergence.
6) **postprocess:** used to determine whether postprocessing of power flow results needs to be performed (=1) or not (=0).

Currently available power flows solvers are shown in Table X.

Table X
POWER FLOW SOLVERS.

No.	domain	method	description
1	'complex'	'cnr_pf'	...

TPSC provides a specialized script for conducting power flow anaysis in the root directory of package called 'power_flows.m'. Users input the name of the desired power system (which must be stored in 'src\power_systems'), and they can customize computation specifics using the 'pfsettings' structure.

In addition, TPCS implements the function:

```
results_pf(data, pfsettings, results);
```

This function is used to display the results of power flow computations. It is interactive, allowing the user to decide what information to print to the terminal.

B. Measurements Generator

C. Static State Estimation

D. Dynamic State Estimation