Purpose

This document is meant to explain PST additions and alterations created to accommodate AGC.

Area Definitions

To enable area calculations, each bus in the bus array must be assigned to an area in the area_def array. An example area_def array is shown below.

```
%% area_def data format
    % should contain same number of rows as bus array (i.e. all bus areas defined)
2
    % col 1 bus number
3
    % col 2 area number
    area_def = [ ...
5
                 1 1;
6
                 2 1;
7
                 3 1;
                 4 1;
9
                 10 1;
10
                 11 2;
11
                 12 2;
12
                 13 2;
13
                 14 2;
14
                 20 1;
15
                101 1;
16
                110 2;
17
                 120 2];
18
```

It should be noted that rows may not have to be in the same order as the bus array (untested). The area_def array is automatically placed into the global g structure.

```
1  >> g.area
2  ans =
3  area_def: [13x2 double]
4  n_area: 2
5  area: [1x2 struct]
```

Each area currently contains values that may be relevant to AGC calculations. The calcAreaVals function is used to calculate and store such values. An example of what is stored in the g.area.area(x) structure is shown below.

```
>> g.area.area(2)
2    ans =
3         number: 2
4         areaBuses: [6x1 double]
5         macBus: [2x1 double]
```

```
macBusNdx: [3 4]
6
             loadBus: [4x1 double]
7
          loadBusNdx: [8 9 12 13]
              genBus: [2x1 double]
9
           genBusNdx: [6 7]
10
                totH: [1x4063 double]
11
                aveF: [1x4063 double]
12
              totGen: [1x4063 double]
13
             totLoad: []
14
                 icA: [1x4063 double]
15
                 icS: ∏
16
      exportLineNdx: [11 12]
17
       importLineNdx: []
18
            n_export: 2
19
            n_import: []
20
```

It should be noted that icS represents a placeholder for a scheduled interchange value and the totLoad is a field for collected total load. The collection of actual running load values may prove more complicated as the bus array does not seem to be updated every step, only a reduced Y matrix used in the nc_load function called from i_simu.

Line Monitoring

Power flow on a line must be calculated each step as AGC requires actual area interchange for the ACE calculation. The previously existing line_pq function performed this task, but was not fully implemented into the simulation to allow calculation during execution. This minor oversight has been resolved, however the lmon_con array is still used to define monitored lines.

```
% Line Monitoring
% Each value corresponds to an array index in the line array.
% Complex current and power flow on the line will be calculated and logged during simulation
% lmon_con = [5, 6, 13]; % lines between bus 3 and 101, and line between 13 and 120
% lmon_con = [3,10]; % lines to loads
```

Line monitoring data is collected in the g.lmon field of the global variable.

```
>> g.lmon
ans =

lmon_con: [3 10]

n_lmon: 2

busFromTo: [2x2 double]

line: [1x2 struct]
```

Each g.lmon.line entry contains the following fields:

```
>> g.lmon.line(2)
1
    ans =
2
        busArrayNdx: 10
3
            FromBus: 13
4
              ToBus: 14
5
              iFrom: [1x4063 double]
6
                 iTo: [1x4063 double]
               sFrom: [1x4063 double]
8
                 sTo: [1x4063 double]
9
```

Weighted Average Frequency

An average weighted frequency is calculated for the total system and for each area if there are areas detected. The calculation involves a sum of system inertias that may change with generator trips. The current algorithm does not account for tripped generators, but was designed to incorporate this feature in the future.

In a system with N generators, M areas, and N_M generators in area M, the calcaveF function performs the following calculations for each area M:

$$H_{tot_M} = \sum_{i}^{N_M} MV A_{base_i} H_i$$

$$F_{ave_M} = \left(\sum_{i}^{N_M} Mach_{speed_i} MV A_{base_i} H_i\right) / H_{tot_M}$$

Then system total values are calculated as

$$H_{tot} = \sum_{i}^{M} H_{tot_{M}}$$

$$F_{ave} = \left(\sum_{i}^{M} F_{ave_{M}}\right) / M$$

If M=0, calcaveF performs

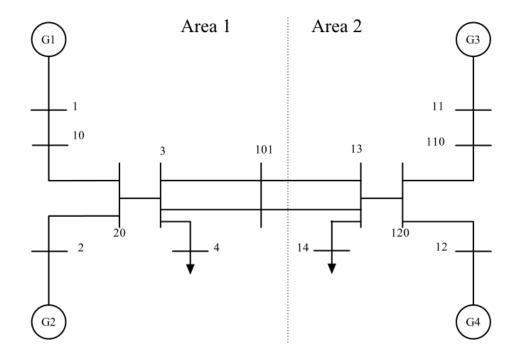
$$H_{tot} = \sum_{i}^{N} MV A_{base_i} H_i$$

$$F_{ave} = \left(\sum_{i}^{N} (Mach_{speed_i} MV A_{base_i} H_i)\right) / H_{tot}$$

Automatic Generation Control

Under development...

General Example Scenario Details



- Kundur 4 machine system packaged with PST
- Constant Z load model
- All machines, exciters, and govs identical
- PSS on gen 1 and 3
- SVC on bus 101
- Event: +50 MW (0.5 PU) step of load on bus 4 at t=1

Initial simulation results (No AGC)

