Experience with Automatic Generation Control (AGC) Dynamic Simulation in PSS®E

Introduction

This article is written to benefit PSS®E customers with their application of Automatic Generation Control (AGC) dynamic simulation in PSS®E. Since the publication of the authors’ article “Automatic Generation Control (AGC) Dynamic Simulation in PSS®E” [1] and the authors’ paper “Extended Term Dynamic Simulation for AGC with Smart Grids” [2], the authors have been contacted by researchers, consultants and engineers worldwide for answers to some how-to questions. In this article, the authors share their experience in AGC dynamic simulation with PSS®E customers.

User-written AGC Model

The extended-term dynamic simulation module in PSS®E is used by the authors. The authors wrote a user-written AGC model which works very well. In a user-written model, many internal variables and arrays are accessible by the model writers. If you are familiar with PSS®E internal variables and arrays, you know that user-written models are preferred. The authors’ user-written AGC model is a generator model with zero MW and Mvar injections. All the AGC functions are implemented in the generator model. Although PSS®E does not convert a zero injection generator to its Norton equivalent circuit, it is suggested that the generator internal impedance (ZR and ZX) in the raw file must be very big so that the generator has no side effect on the system. A sample generator record is shown below:

```
101,'GC', 0.000, 0.000, 0.000, 0.000, 1.02500, 0, 100.000, 1.00000E10, 3.00000E10, 0.00000E+0, 0.00000E+0, 1.00000, 1, 100.0, 90.000, 0.000, 1, 1.0000
```

If the study system is small, you can use state space dynamic simulation in PSS®E. It is required that the integration time step must be small, e.g., 0.008333s. For a large power system and long study period, a small integration time step makes the simulation process very long. In this case, extended term dynamic simulation is preferred.

In case you are not familiar with PSS®E internal variables and arrays, the Python AGC model is recommended. PSS®E provides various Python APIs for you to implement the AGC model. However, the model writers are subject to the limitations of available Python APIs.

In this article, only the user-written model and extended term dynamic simulation are discussed.

Supervisory Control

The authors used governor model GGOV3 in the AGC dynamic simulation. Model GGOV3 is not included in the PSS®E standard library yet. However, it is similar to GGOV1. The block diagram of GGOV1 is shown in Figure 1. The supervisory control input is Pmwset which is stored in VAR(L+6). Updating VAR(L+6) to the setpoint calculated by AGC implements the supervisory control of the governor.

If you use other types of governor models, turbine load controller model LCFB1 has to be used. The supervisory control input Pmwset is stored in array LCREF. Updating LCREF to the setpoint calculated by AGC implements the supervisory control of the governor.
Integration Time Step and Simulation Period

An issue with a small integration time step is related to single precision float number. Single precision float numbers are used in PSS®E to define some internal variables and arrays. There is no issue at all if the simulation period is short. However, for long-term dynamic simulation, issues may arise. In PSS®E dynamic simulation, most quantities (voltages, state variables, etc.) have reasonable value limits. There are two exceptions: simulation period and generator angles (discussed later). Assume \( t_i \) is the cumulative time value. Consider the following update:

\[
t'_{i+1} = t_i + \Delta t
\]

Where \( \Delta t \) is the integration time step. Both \( t_i \) and \( \Delta t \) are stored in single precision float numbers. The above update has no issue if the simulation period is short. If the simulation period is long, however, the round-off error in (1) will accumulate and may be intolerable when \( i \) reaches a big number. As an example, assume \( i \) starts from zero and the half cycle integration time step 0.008333s is used (\( \Delta t = 0.008333s \)); the results of update in (1) are shown in Figure 2.
Cumulative Time (Time Step = 0.008333s)

It can be seen that the cumulative time has intolerable round-off error at the early stage of simulation. Actually, the cumulative time has round-off error when $i \geq 2$. The round off error accumulates very quickly. After $i = 24986956$ (time = 208224s), $\Delta t$ loses significance in (1) and $t_i$ becomes flat.

In order to eliminate the round-off error, the following integration time steps are suggested (in seconds): 2.0, 1.0, 0.5, 0.25, 0.125, 0.0625, 0.03125, 0.015625, 0.0078125 and 0.00390625. As an example, assume $i$ starts from zero and the integration time step 0.0078125s is used ($\Delta t = 0.0078125$s); the results of update in (1) are shown in Figure 3.
It can be seen that update in (1) has no round-off error at all until \( i = 16777216 \) (time = 131072s). After \( i = 16777216 \) (time = 131072s), \( \Delta t \) loses significance in (1) and \( t_i \) becomes flat. So the maximum simulation period is 131072 seconds or about 36.4 hours. Table 1 shows the maximum simulation period for different integration time steps.

<table>
<thead>
<tr>
<th>Integration Time Step (seconds)</th>
<th>Maximum Simulation Period (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9320</td>
</tr>
<tr>
<td>1</td>
<td>4660</td>
</tr>
<tr>
<td>0.5</td>
<td>2330</td>
</tr>
<tr>
<td>0.25</td>
<td>1165</td>
</tr>
<tr>
<td>0.125</td>
<td>582</td>
</tr>
<tr>
<td>0.0625</td>
<td>291</td>
</tr>
<tr>
<td>0.03125</td>
<td>145</td>
</tr>
<tr>
<td>0.015625</td>
<td>72</td>
</tr>
<tr>
<td>0.0078125</td>
<td>36</td>
</tr>
<tr>
<td>0.00390625</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 1 - Maximum Simulation Period for Different Integration Time Steps
If the simulation period is within the maximum simulation period for the chosen integration time step, there is no round-off error in cumulative time.

The authors prefer 0.25 second and larger as the integration time step. Using 0.25 second, the maximum simulation period is greater than 1165 hours (which is more than enough for AGC dynamic simulation) and also the simulation process is reasonably fast.

Aside from numerical integration stability requirement, system dynamic behavior requirement, SCADA cycle and AGC cycle, is there any other upper limit on integration time step? The answer is YES. But why and how? The readers are encouraged to answer the questions. Another article (or paper) addressing this is planned by the authors.

**Generator Angle Adjustment**

Generator angles in PSS®E are stored in single precision float arrays. Consider the following update:

\[ \delta_{i+1} = \delta_i + \Delta\delta_i \]  

Where \( \delta_i \) and \( \Delta\delta_i \) are generator angle and angle correction respectively, and both are stored in single precision float numbers. Update in (2) will have a problem if \( |\delta_i| >> |\Delta\delta_i| \). This may happen when the system is under frequency (or over frequency) for a long time and then becomes normal. If a small integration time step \( \Delta t \) is used, the situation is even worse because \( |\Delta\delta_i| \) is much smaller for small \( \Delta t \) even with dramatic system disturbances. A small integration time step makes the simulation process much longer. AGC dynamic simulation is generally used to simulate slowly changing power systems, so a large integration time step is preferred in order to speed up the simulation process.

Even with a large integration time step, update in (2) may still cause a problem because there is no limit on \( |\delta_i| \). Fortunately, adding 360 degrees to \( \delta_i \) or subtracting 360 degrees from \( \delta_i \) doesn’t affect simulation results in PSS®E. So it is suggested to keep \( \delta_i \) between -200 degrees and 200 degrees. If \( \delta_i > 200 \), subtract 360 from \( \delta_i \). If \( \delta_i < -200 \), add 360 to \( \delta_i \). It is a little tricky to do so because a generator angle is generally a state variable: several internal arrays should be updated accordingly.

As an example to show the results from generator angle adjustment, a three-bus system (a regular generator, a wind farm and a load) is used to run AGC dynamic simulation. Figure 4 shows the angle of the regular generator with and without angle adjustment. Figure 5 shows the MW output of the regular generator with and without angle adjustment. Figure 6 shows ACE (Area Control Error) (one hour time span from 10000s to 13600s) with and without generator angle adjustment. Figure 7 shows CPS1 (Control Performance Standard 1) with and without generator angle adjustment. The simulation results show that the generator angle adjustment is correct. Although the results with angle adjustment and the results without angle adjustment are matched in the simulation, it can be concluded that simulation results with angle adjustment are more accurate if there are mismatches.
Figure 4 - Generator Angle

Figure 5 – Generator MW
Conclusions
PSS®E provides customers with the capability to simulate AGC dynamically under the condition of intermittent generation and varying load. This simulation can help the power system planner to study and understand AGC behavior when a renewable energy resource is planned. The information provided in this article clarifies some confusion and will help PSS®E customers to do their AGC dynamic simulation or long-term dynamic simulation.

Reference