INTELLIGENT ELECTRICAL POWER GRIDS

MODELLING OF GENERATOR CONTROLLERS

1. OBJECTIVES AND PREPARATION

This tutorial is meant to give an insight into power system controls. This tutorial builds on the last tutorial whereby uncontrolled generators were modelled in the IEEE 9 bus system.

The objective of this lab session is to develop dynamic models of generators, which serves to give an insight into power system control with OpenModelica. AVR and Governor models are added to generator systems. This will show how to control dynamic response of systems in event of disturbances. At the end of this tutorial, the user will have learned and have had experience with power system modeling and control.

2. MACHINE CONTROLLERS : AUTOMATIC VOLTAGE REGULATORS

The system that was created for the previous exercise is an uncontrolled system. The effect of sudden introduction of load on the system voltage was seen. To reduce the voltage imbalance in the system, we have to introduce a control mechanism. Power plants generally employ Automatic Voltage Regulators (AVR) to control generator voltage in event of voltage changes. An AVR is a feedback control system that measures the output voltage of the generator, compares that output to a set point, and generates an error signal that is used to adjust the excitation of the generator. As the excitation current in the field winding of the generator increases, its terminal voltage will increase.

To introduce the AVR, we will need to make some additions to our existing generator model.

1. Open the generator workspace.
2. Locate *AVRTypeII* model in the OpenIPSL library at **OpenIPSL → Electrical → Controls → PSAT → AVR → AVRTypeII** and drag them to the generator workspace.
3. Drag two ***Constant*** blocks from **Modelica → Blocks → Sources → Constant** for AVR module**.** Name them const1 and const2.
4. Delete the existing connection between Generator.vf0 to Generator.vf.
5. Make the following connections:
    a. Generator.v → AVR.v
    b. const1.y → AVR.vref
    c. AVR.vf → Generator.vf

d. const2.y → AVR.vf0

6. Your final model for each generator should look like that in Figure 1.

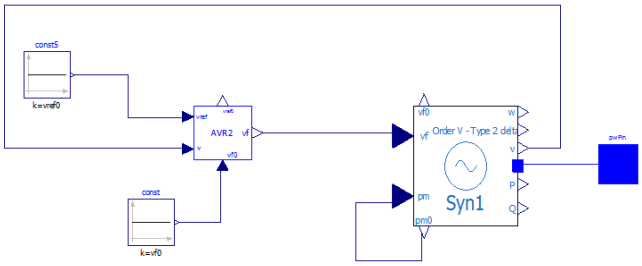

Figure 1: Final Generator + AVR configuration

7. Similar to how we introduced the generator parameters in the equations mode for generalizing the generator system, we will do the same with AVR. Copy the following code in the text view of the generator model workspace, before the equation section.

```
  parameter Real vf0=1.122656195484139 "Initial field voltage"
annotation (Dialog(group="AVR parameters"));
  parameter Real vref0=1.065622531687790 "Reference voltage
AVR" annotation (Dialog(group="AVR parameters"));
  parameter Real Ta=0.02 annotation (Dialog(group="AVR
parameters"));
  parameter Real Kf=0.002 annotation (Dialog(group="AVR
parameters"));
  parameter Real Tf=1 annotation (Dialog(group="AVR
parameters"));
  parameter Real Ke=1 annotation (Dialog(group="AVR
parameters"));
  parameter Real Te=0.2 annotation (Dialog(group="AVR
parameters"));
  parameter Real Tr=0.001 annotation (Dialog(group="AVR
parameters"));
  parameter Real vrmax=7.32 annotation (Dialog(group="AVR
parameters"));
  parameter Real vrmin=0 annotation (Dialog(group="AVR
parameters"));
  parameter Real Ka=200 annotation (Dialog(group="AVR
parameters"));
```

8. In the diagram view of the generator workspace, open the AVR dialog and fill the text spaces next to parameter names as shown in Figure 2. ***Note that parameter v0 is filled with V_0.***

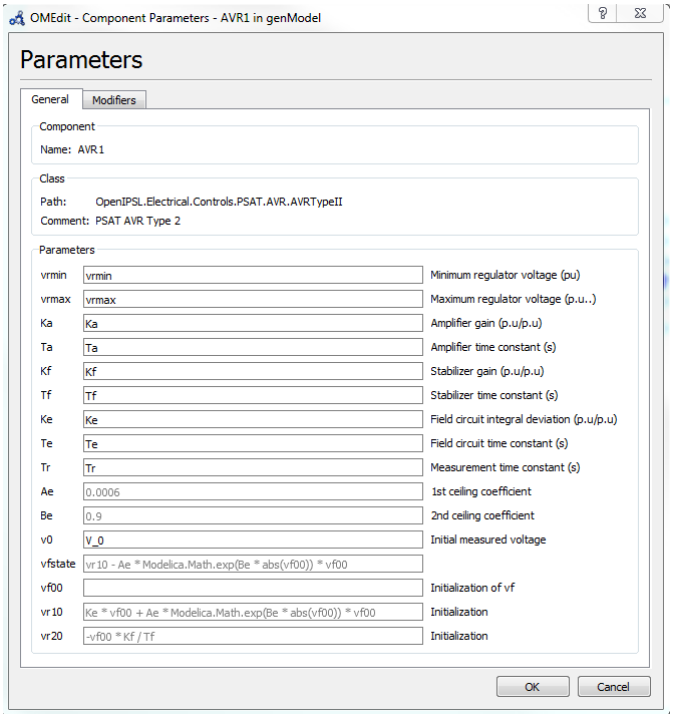

Figure 2: AVR parameters

9. Open the *const1* box and name the parameter k as vref0.

10. Set k as vf0 in *const2* block.

After generator models have been undated, save the model, and go to network model workspace. Open any generator properties dialog box. It should now contain options to set AVR parameters as well. The generator properties box opened from network model workspace should look like that in Figure *3*. These values can now be populated for each AVR. The values for each generator are given in **Table *1***.

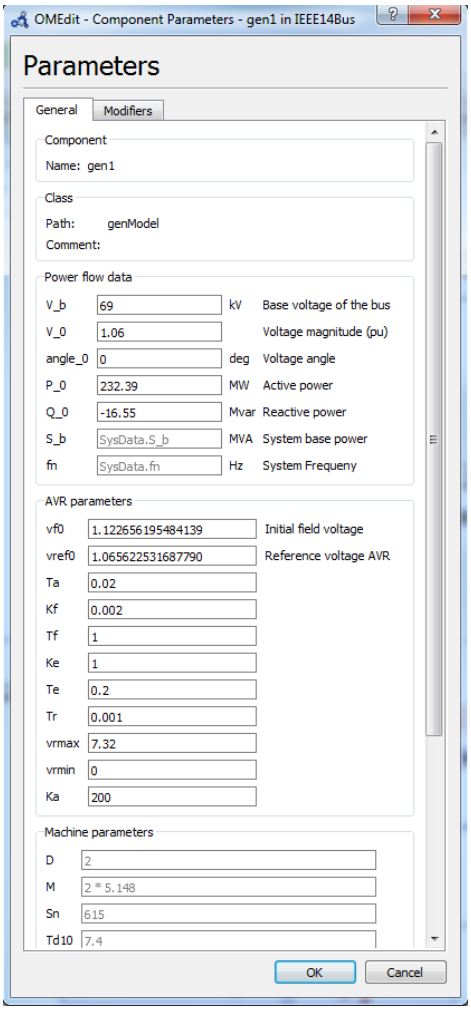

Figure 3: Generator Dialog box from Network Model Workspace. You can now set AVR parameters in addition to machine parameters.

Modelling tip: Your network may stretch the boundaries of your work area in Modelica. You can resize your workspace by right clicking on work area and changing *Properties*. Here you can change the *top, bottom, right, left* parameters to stretch workspace.

Table 1: AVR Parameters

	Ae	Be	Ka	Ke	Kf	Ta	Te	Tf	Tr	vf0	vref0	v0	vrmax	vrmin
AVR1	0.0039	1.555	50	1	0.063	0.2	0.314	0.35	0.001	1.0821	1.062	1.04	5	-5
AVR2	0.0039	1.555	50	1	0.063	0.2	0.314	0.35	0.001	1.7893	1.06	1.025	5	-5
AVR3	0.0039	1.555	50	1	0.063	0.2	0.314	0.35	0.001	1.4029	1.05	1.025	5	-5

3. WORKFLOW

After adding the AVR, it is now time to observe it in action. To observe the effect of adding AVR on system voltage, we introduce a new event: load step event. Follow the instructions to create a load step event:

- In the network model workspace, delete the fault component, if present
- Also delete the load on bus 7.
- Add a new load by adding it from **DelMod → Loads → scaleLoad**.
- Connect the load to bus 7.
- Initialize the load (P, Q, V_0, angle_0) with the exact same values as for previous load.
- Add a Step block next to the load from **Modelica → Blocks → Sources → Step**.
- Connect it to the input of load as shown in Figure 4.
- Set height = 0.2, offset = 1, startTime = 50
- Simulate the system for 150s.

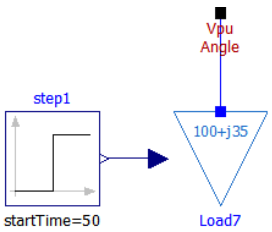

Figure 4: Creating load step event at Load 7

Sub-task 3: Perform the above load step event for the uncontrolled system as well (Remove the fault element before adding the load step). Compare the result of uncontrolled with the controlled case.

4. MACHINE CONTROLLERS : GOVERNOR SYSTEMS

Frequency of the system is dependent on active power balance. As frequency is a common factor throughout the system, a change of active power demand at any point (a generator outage, or in this case, a load step) will affect the system frequency. In the previous case, if generator frequency (or speed, ω) is plotted, you will observe how the frequency is affected by a load change.

Because we have three generators in the system, there must be provided a way to allocate change in demand to the generators. A speed governor on each generating unit provides primary speed control function, while supplementary/secondary control originating at central control center allocates generation. Frequency control was detailed in the lecture earlier, so this guide will not go into details of frequency control theory. The governor model used in this session is given below in Figure 6.

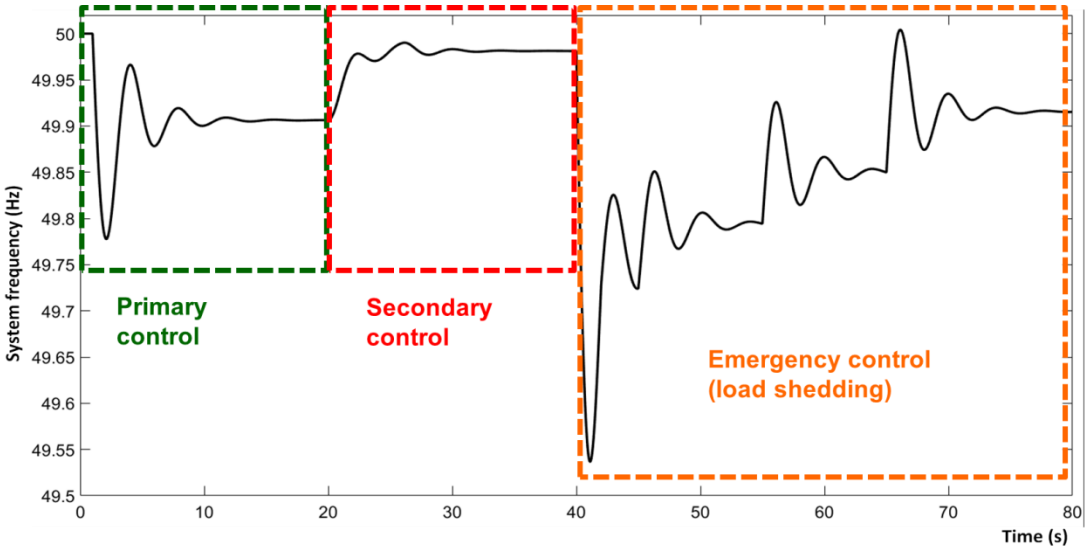

Figure 5: Frequency control

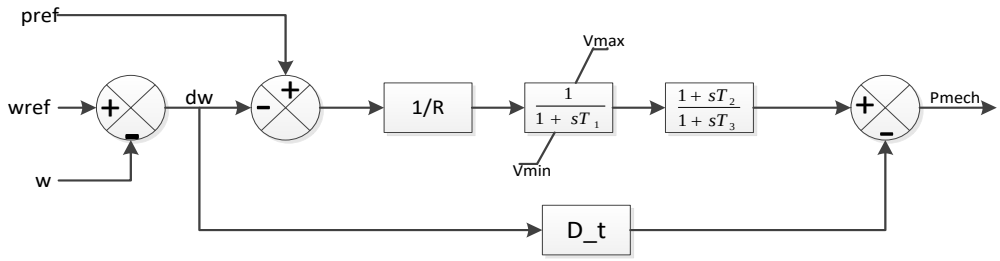

Figure 6: TGOV1 Governor model used in the system

It is advised to create a copy of the model file that was created during last session (IEEE 9 bus with AVR). Rename this newly copied file and load it in the OpenModelica environment along with the relevant libraries as detailed in the last session. We will now add governor modules to the generator systems.

1. Open the generator workspace.
2. Drag a constant block into the workspace (name it as const1) and set its value at 0.
3. Locate *GOV* model in the DelMod library **DelMod → Control → TGOV1**.
4. Drag it into the generator workspace. Make the following connections:
    a. **TGOV1.PMECH → gen.pm**

b. **gen.pm0 → TGOV1.PMECH0**

c. **const1.y → TGOV1.AGCInput**

d. **Generator Speed (w) → Governor Speed (SPEED)**

5. Copy the following code in the equation view of the generator model workspace, before the equation section.

```
parameter Real R = 0.05 annotation (Dialog(group="GOV
parameters"));
parameter Real T_1 = 0.04 annotation (Dialog(group="GOV
parameters"));
parameter Real T_2 = 2.1 annotation (Dialog(group="GOV
parameters"));
parameter Real T_3 = 7 annotation (Dialog(group="GOV
parameters"));
parameter Real V_MAX = 3 annotation (Dialog(group="GOV
parameters"));
parameter Real V_MIN = 0 annotation (Dialog(group="GOV
parameters"));
parameter Real K_11 = 1 annotation (Dialog(group="GOV
parameters"));
```

Note: In case your parameter box shows K_2, replace the K_11 in the code to K_2 . Ensure that all the K_II are replaced with K_2 for proper functioning of the model.

6. In the diagram view of the generator workspace, open the Governor dialog and fill the text spaces next to parameter names as shown in Figure 7.

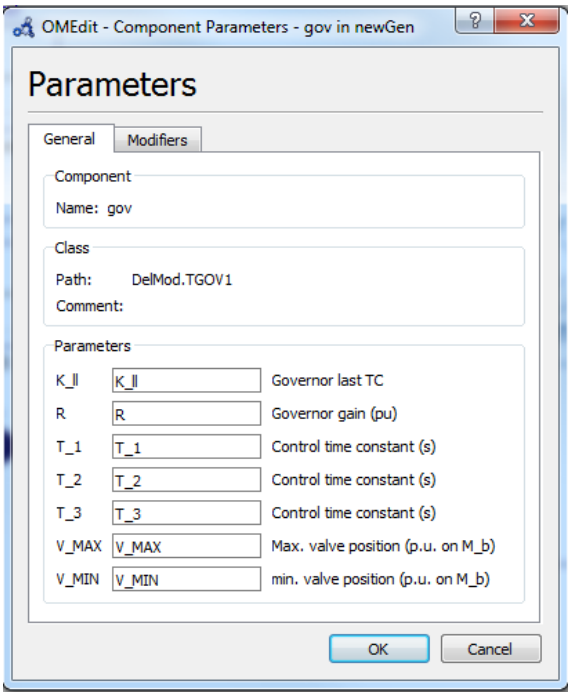

Figure 7: GOV parameters

After generator models have been updated, go to network model workspace. Open any generator properties dialog box. It should now contain options to set GOV parameters. The generator properties box opened from network model workspace should look like that in Figure 8. These values can now be populated for each GOV. The values for each generator are given in Table 2

Table 2: Governor Parameters

Parameter	GOV1	GOV2	GOV3
K_ll	1	1	1
R	0.01	0.01	0.01
T_1	0.04	0.04	0.04
T_2	2.1	2.1	2.1
T_3	7	7	7
V_MAX	3	3	3
V_MIN	0	0	0

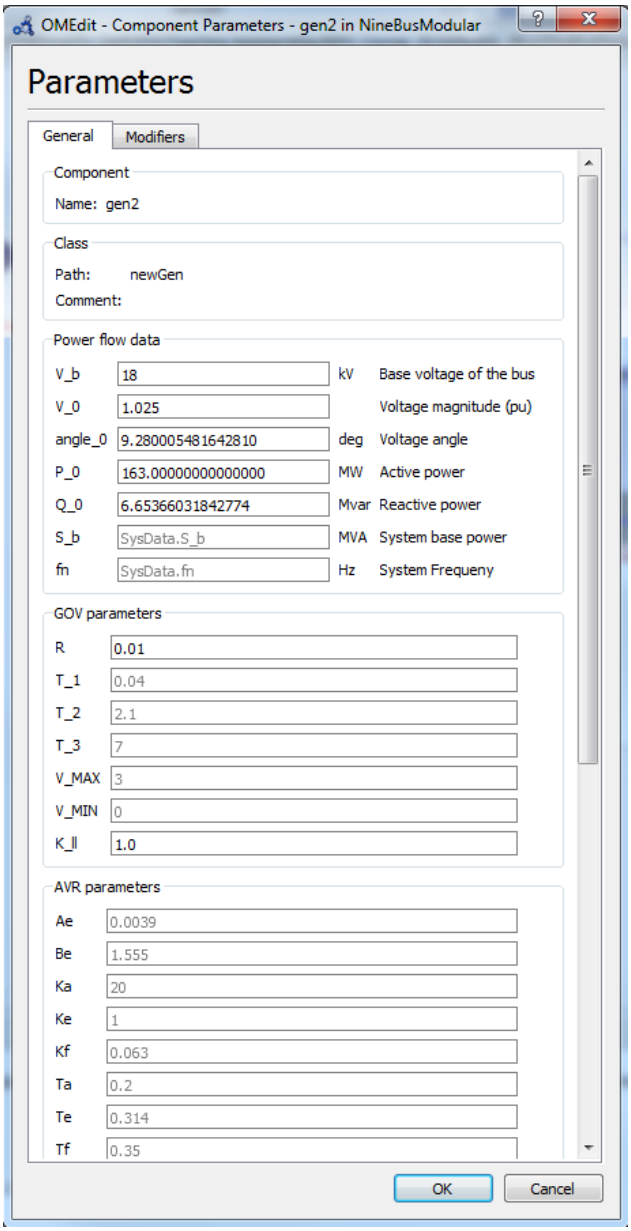

Figure 8: GOV parameters are now reflected in network model workspace

5. WORKFLOW

After adding the GOV, we will now observe its effect. To observe the effect of adding GOV on system frequency, use the same load event as previously in last section with AVR.

Sub-task 4: Compare the frequency of the generators in the uncontrolled case, AVR case, and GOV case. What conclusions can you draw from these observations?