Dynamic Modeling Using the Graphical Model Builder (GMB)

Olaf Ruhle
Senior Consultant
olaf.ruhe@siemens.com

Abstract
Successful operation of a power system depends largely on the engineer's ability to provide safe, reliable and economic service to the customer. Advanced simulation technologies provide useful means to the engineer for the design and analysis of the power system, and assisting them in making reasonable decisions. Due to powerful software like PSS™E, it became possible to simulate the dynamic behavior of very large power systems and to verify the performance of these complex systems in a fast and accurate manner. The simulation covers all stages of development and operation of a power system, such as planning, design, test and also during operation. This paper describes the new module “Graphical Model Builder” of PSS™E-30.3. Using this Graphical Model Builder every dynamic model like AVR, exciter, FACTS, wind models, etc. can be built in graphical block-diagram form using Microsoft VISIO. The models can be easily included in PSS™E dyr files without compiling and linking. For Rev 30.3, only AVRs and Governors will be supported in the Graphical Model Builder. Support for additional model types will be rolled out in subsequent releases.

The old Matlab/Simulink interface to PSS™E was plagued with problems: 1. Matlab/Simulink was expensive; 2. Upgrades of Matlab/Simulink libraries tended to break existing code. In PSS™E-30.3, a new Microsoft VISIO based Graphical Model Builder (GMB) is integrated as a stand-alone application. This Graphical Model Builder uses the powerful VISIO interface to create easily dynamic models (Figure 1). It is a drawing tool that is simple and quick to operate for implementing, editing and documenting of dynamic models:

- Excitation systems (AVRs)
- Turbine Governors
- Power System Stabilizers
- HVDC Models
- FACTS Models
- Load Models
- Transformer Models
- New Sources Models (e.g. Generic Wind Models)
- New Storages Models
Besides familiar CAD functions, such as copying, shifting, rotating, zooming, etc., the system has a large symbol library which contains more than 100 different control blocks in the form of symbols. The user establishes system diagrams and the block diagram by graphical connection of library symbols. The data is input via masks that are object-related and have abbreviated aid texts in addition to detailed aid texts. It is also possible to combine groups of linked symbols to form independent new symbols as macro models and to add these to the symbol library or to the user’s own library. On the basis of this hierarchical structuring capability (“Subsystems”), the system makes it possible to decide, according to requirements and for the same database, in how complex or simplified a way a system can be illustrated. Individual components can be activated and deactivated and connected to any desired part of the system.

The symbol library “BOSL” (Block-Oriented Simulation Language) contains more than 100 different function blocks. These blocks can be combined to any open or closed-loop control structures or evaluation devices by means of the graphic interface. Besides very simple blocks, such as PID elements, there are also complex “blocks”, such as FFT (Fast Fourier transformation). The controllers can be stored as subsystems in a library so as to link them quickly to a system. Parameterizing can be input individually and changed, or the default values can be used (Figure 2). Optionally complex open and closed-loop control and protective functions can be implemented with the block-oriented simulation language. Besides the open and closed-loop control structure, signal processing structures can also be defined by the user (evaluation devices). External, user-defined subroutines can also be coupled (open-loop) and there is an interface to real-time applications (closed-loop). The block-oriented structures can be combined with FORTRAN-like terms (Figure 3), such as mathematical functions, logical terms or instructions, e.g. IF/THEN/ELSE and GOTO/CONTINUE. Input variables are available to the controllers in all sizes. In addition, the variables from other closed and open-loop controllers or the evaluation structures can be used as input variables. All inputs and outputs of blocks can be output.

The user can switch between 2 different block styles: 1. The European DIN symbols and 2. The transfer functions (Figure 4). The Graphical Model Builder also offers testing and debugging functionalities like in Matlab/Simulink. A step-function or a sinusoidal signal can be injected at each point of the structure (Figure 5, red arrow). Also at each point of the structure the block signals can be plotted (Figure 5, small monitors). These signals are available as VARs in PSS™E.
After finishing the design of the model the DYRE data record with the data of the BOSL model can be created:

'USRMBL' 'MODEL' 101 '1' 0.02 205.1 0.0035 3.0 -4.0 1.0 1.3
0.0382 0.23 1.22 3.45 0.03 4.6 0.096 0.00 1000. /

'Model' is the model name and is also the name of the file that contains the model block diagram. DYRE does not generate any connection routines (hence no need for compiling and linking). The user can now start PSS®E and the dynamic simulation.

Figure 6 shows the basic voltage control of a DFIG (Doubly Fed Induction Generator) wind machine build with BOSL (example).

For the usage of the Graphical Model Builder Siemens PTI offers Trainings, Seminars and Service Hotline. For more information please contact Dr. Olaf Ruhle (Olaf.Ruhle@siemens.com).

Figure 2 - Data Input in Masks after Double Clicking
Figure 3 - Special User Blocks with FORTRAN Statements

```fortran
Foc1=Imp_1+Imp_2
Foc2=Imp_2**6.5
IF (Foc2 .LT. 2) THEN
  Foc3=Foc1+Foc2*68888
ELSE
  Foc3=Foc1+Foc2*888
ENDIF
```
Figure 4 - Switching Between DIN-symbols and Transfer Functions
Figure 5 - Testing and Debugging Functionality

Figure 6 - Example: Basic Voltage Control for a DFIG