

PSS[®]E

Graphical Model Builder (GMB)

At a glance

PSS[®]E Version 30.3 introduced the Graphical Model Builder (GMB), a powerful and easy to use dynamics model development tool based on graphic representation of control block diagrams. Accurate power system planning requires that precise models be used in the simulation software. PSS[®]E software is designed for dynamic modeling of complex systems. GMB allows specialized modeling by supporting both standard and non-standard dynamic models.

Advantages of the GMB include:

- Based on Microsoft[®] Visio[®] (graphical block-diagram form)
- Supports a wide range of dynamic models
- Uses common CAD functions plus large symbol library
- Model flexibility using 100+ control blocks

The challenge

Successful operation of a power system depends largely on the engineer's ability to design a system that provides safe, reliable and economical service to the customer. Advanced simulation technologies provide the means for the engineer to design and analyze power systems, and assist him in making key decisions. With powerful simulation software like PSS[®]E, it is 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.

However, as new technologies become available for application in the power system, it is necessary to develop dynamic models to accurately represent their behavior. New models must be developed quickly and accurately and be fully tested prior to use in the power system simulation.

Our solution

GMB is a stand alone model builder and testing environment that can generate dynamic models for use in PSS[®]E. GMB uses Microsoft[®] Visio[®] software to easily create dynamic models (Figure 1). Using Visio, the GMB becomes a drawing tool that is simple and quick for implementing, editing and documenting of dynamic models. PSS[®]E version 30.3 incorporated GMB for:

- Excitation systems (AVRs)
- Turbine governors

In rapid succession, PSS[®]E will be extended to incorporate the following GMB models:

- Power System Stabilizers
- HVDC models
- FACTS models
- Load models
- Transformer models
- New source models (e.g. generic wind models)
- New storage models

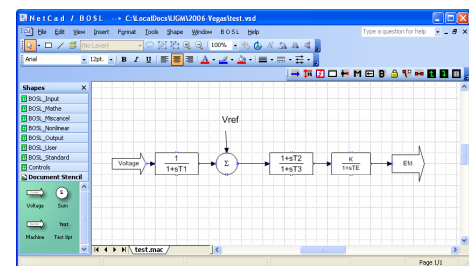


Figure 1: Visio-based Graphical Model Builder

Using GMB, the user can develop a wide variety of dynamic models (i.e. AVR, exciter, FACTS, wind models, etc.) which can be constructed using GMB's built-in graphical block-diagrams coupled with Microsoft[®] Visio[®]. The models can be easily included in PSS[®]E DYRE files without the need for compiling and linking.

During model development, GMB presents a stand-alone model simulation package that allows testing of the independent model. The simulation can be driven using built in signal generators and test points that allow simulating the response of the model over a full range of inputs.

Besides familiar CAD functions, such as copying, shifting, rotating, zooming, etc., the GMB system has a large symbol library which contains more than 100 basic control blocks in the form of symbols. The user develops the model by graphically interconnecting the basic library symbols and editing the properties of each controller element via intuitive dialog boxes. Further, the user may then use this model as a “subsystem” in the symbol library which can then be used in constructing additional models. Library 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 generate any open or closed-loop control structures or logic devices by means of the graphic interface. Besides very simple blocks, such as PID elements, subsequent PSS[®]E releases will provide the ability to incorporate complex “blocks”, such as FFT (Fast Fourier Transformation). Parameter values can be entered and edited individually, or the default values can be used (Figure 2).

Complex open and closed-loop control and protective functions can be implemented with GMB. Besides the open and closed-loop control structure, signal processing structures can be defined by the user. External, user-defined subroutines can also be coupled (open-loop) and there is an interface to real-time applications (closed-loop).

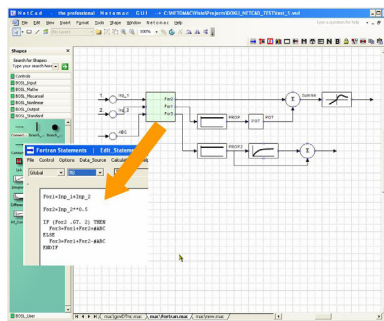


Figure 2: Data input in masks after double clicking

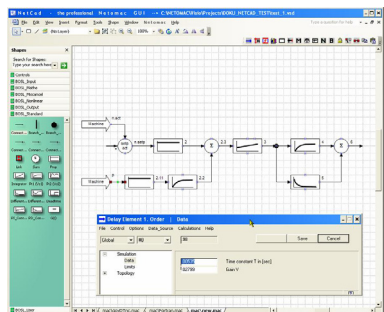


Figure 3: Special user blocks with FORTRAN statements

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 units. 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 printed or plotted.

In working with the graphics, the user can switch between two different block styles: the European DIN symbols; and transfer function diagrams (Figure 4). GMB also offers testing and debugging functionalities in the stand alone mode. A step-function or a sinusoidal signal can be injected at each point of the model and the signal responses can be monitored and plotted. These signals are available as variables for use in PSS[®]E.

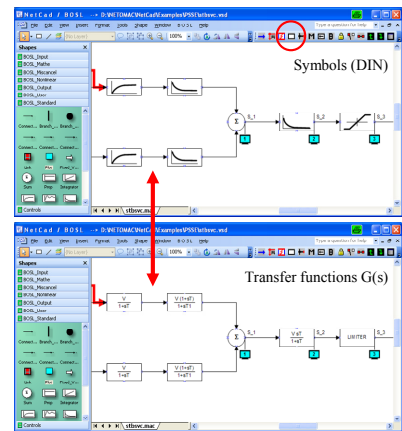


Figure 4: Switching Between DIN-Symbols and Transfer Functions

After finishing the design of the model the DYRE data record with the data of the GMB 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.

Application example

Figure 5 shows the basic voltage control of a DFIG (Doubly Fed Induction Generator) wind machine build with BOSL (example).

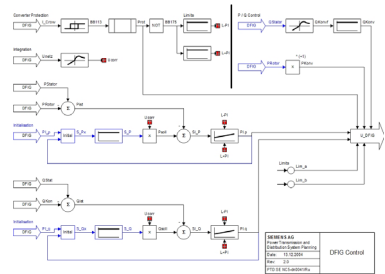


Figure 5: Example: Basic voltage control for a DFIG