HBG2Simulink : User Guide

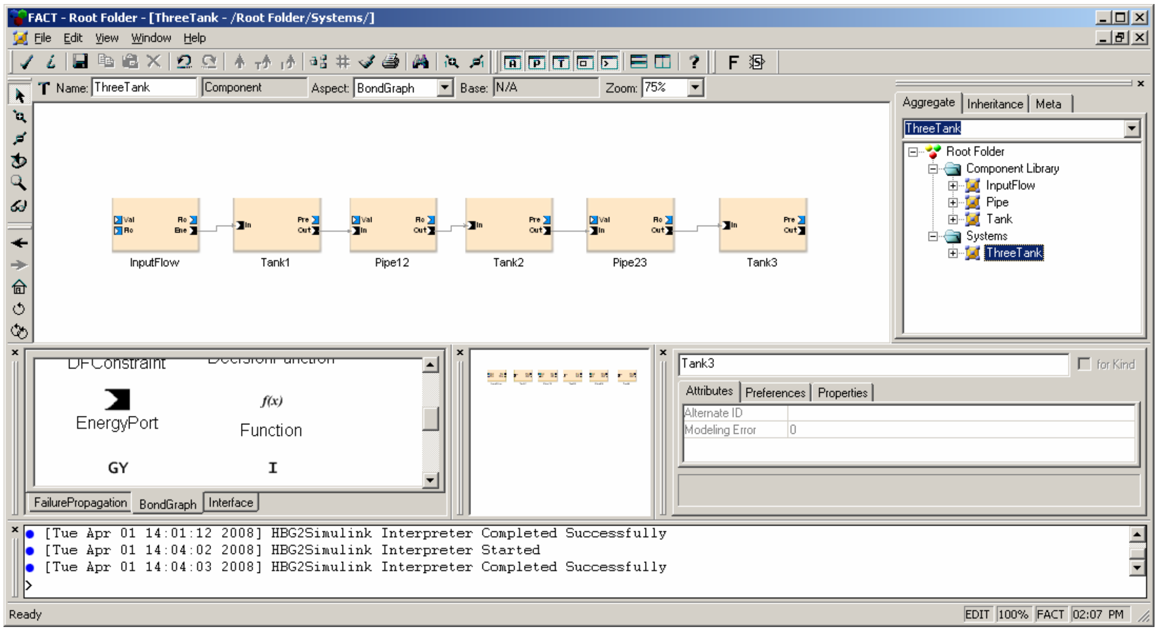
Overview

HBG2Simulink is an interpreter for the FACT paradigm. It transforms a model constructed in GME using the FACT modeling language to a Simulink model. It outputs two files of interest, both Matlab .m files to be executed in the Matlab environment:

1. Build script – calls the functions which construct the Simulink model
2. Data structure script – stores a global flat bond graph structure for model execution

Throughout this document, the interpreter will be demonstrated using the following sample model: FACT/models/ThreeTank/ThreeTank.mga

The GME view of the ThreeTank model is shown below. The interpreter icon is .



```
[Tue Apr 01 14:01:12 2008] HBG2Simulink Interpreter Completed Successfully
[Tue Apr 01 14:04:02 2008] HBG2Simulink Interpreter Started
[Tue Apr 01 14:04:03 2008] HBG2Simulink Interpreter Completed Successfully
>
```

Installation

All the model construction and run-time execution functions are available in the following directory:
FACT/src/HBGSimulinkLibrary
If installing on a machine that does not have FACT installed, this directory must be copied to the machine.

In Matlab, change the working directory to the HBGSimulinkLibrary directory. In the Matlab prompt, run the installer script:

```
>> installHBGSimulink
```

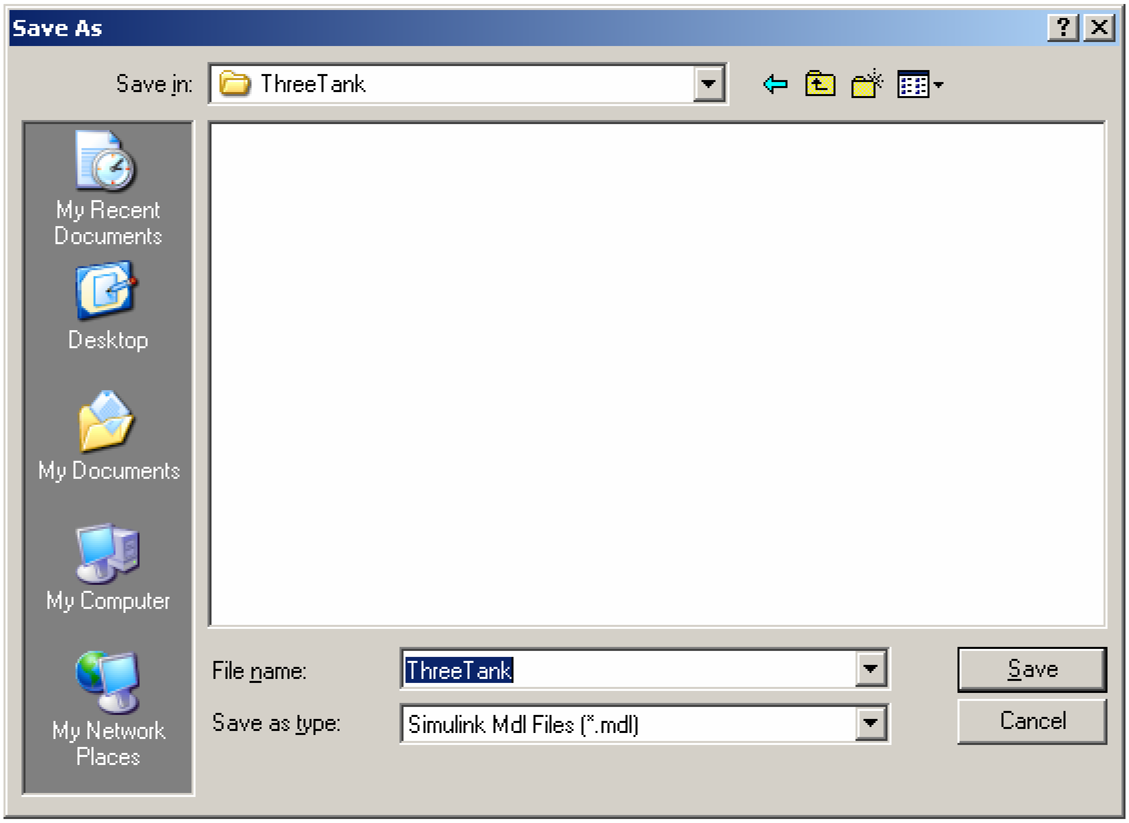
This will add the HBGSimulink directories to your Matlab search path so that the functions are accessible to Matlab. It will also compile the included Simulink S-functions.
To uninstall, run the uninstall script:

```
>> uninstallHBGSimulink
```

This will remove the HBGSimulink directories from your Matlab search path.

Running the Interpreter

Open a FACT model, and open the System or Component you wish to interpret, so that it is visible in the main GME window. Click the HBG2Simulink icon (S). A save file dialog will appear:



Save in: ThreeTank

File name: ThreeTank

Save as type: Simulink Mdl Files (*.mdl)

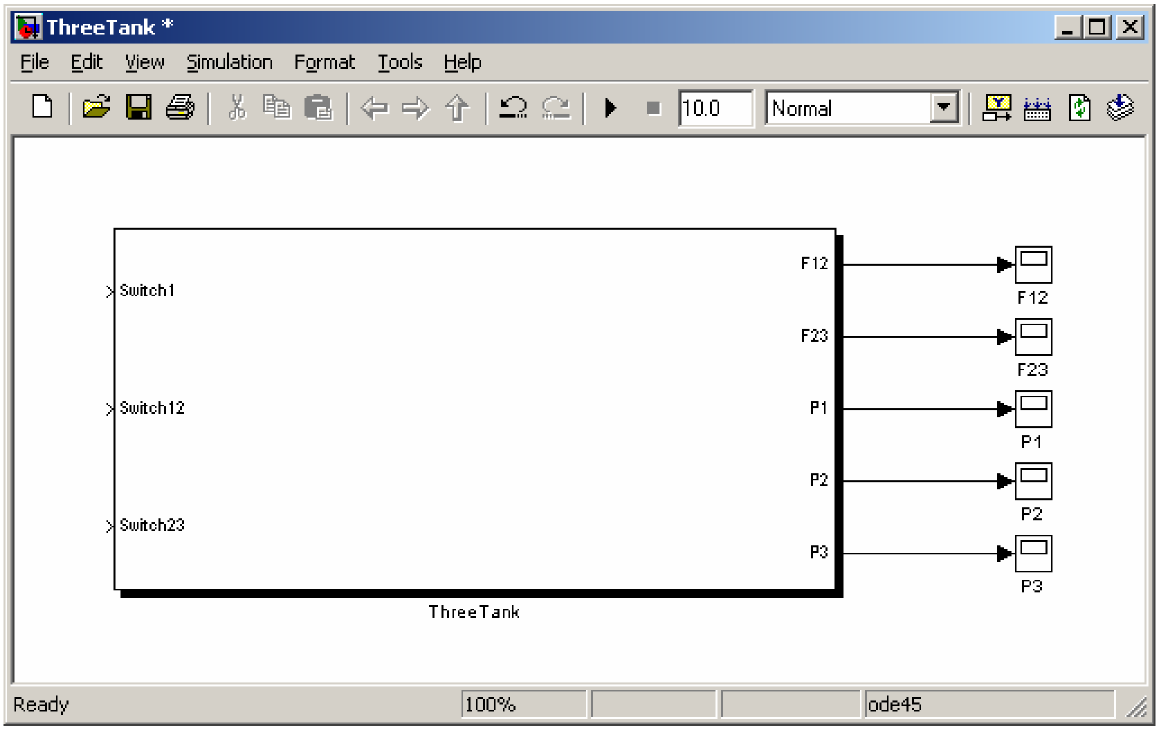
The system or component name is entered as the default. Enter the desired model name (e.g. ThreeTank), and choose save. If you have a modeling error (e.g. a signal port is being used as both an input and an output), the interpreter will quit with a description of the error. If you have a warning (e.g. a parameter value is 0), the interpretation will complete and warnings will be printed to the GME console. If successful, the interpreter will create two files prefixed with the entered file name: a Matlab .m file containing the flat bond graph data structure (e.g. ThreeTankDataStructure.m), and a Matlab .m file which will construct the Simulink model (e.g. ThreeTankBuild.m).

Constructing the Simulink Model

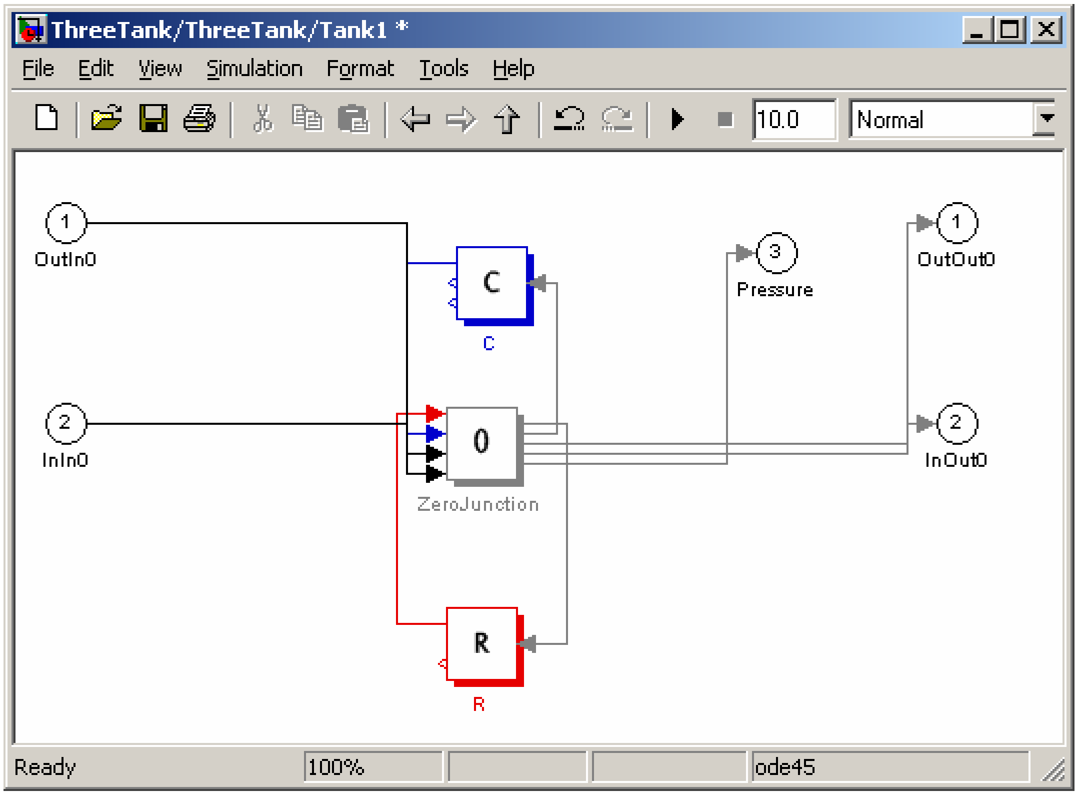
To construct the Simulink model of your system, save the data structure script and build script to the directory you wish your model to be constructed in. In the Matlab prompt, run the build script. For example,

```
>> ThreeTankBuild
```

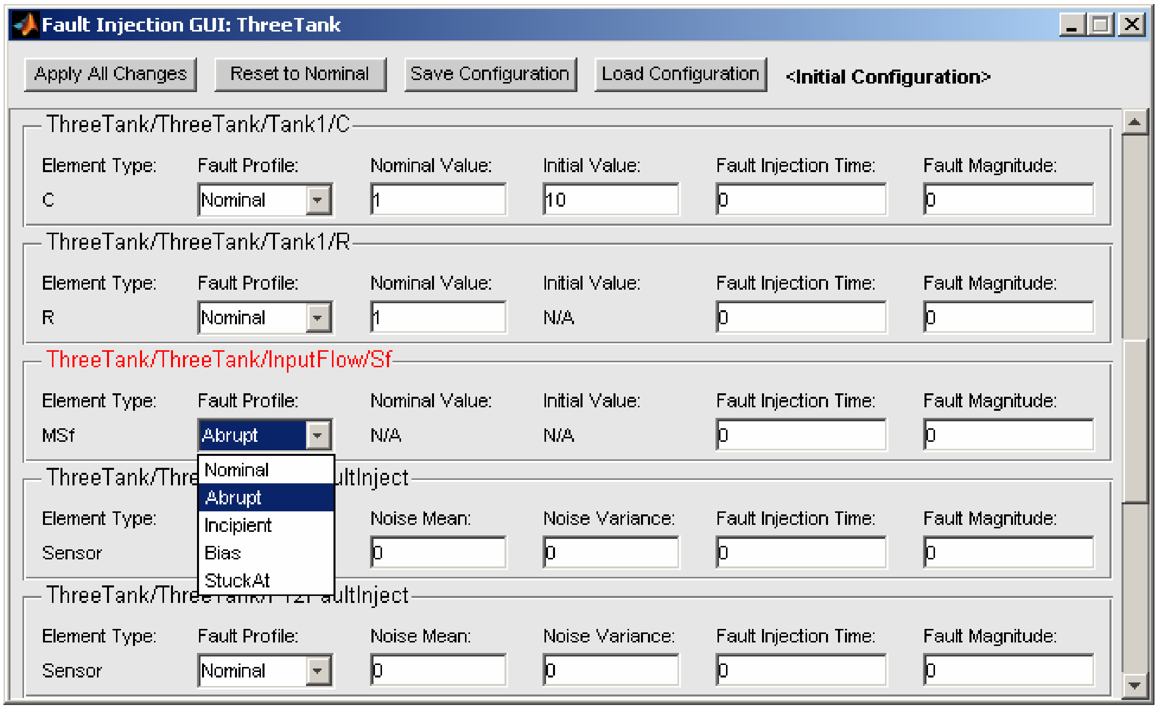
The Simulink model library will be opened, and the model will be constructed and saved as the name initially entered in the save dialog of the interpreter (e.g. ThreeTank.mdl). The model will be wrapped with a subsystem block, with inputs for each PlantInput or top-level input BGSignalPort object, and outputs for each PlantOutput or top-level output BGSignalPort object, with Simulink scopes attached.



No default inputs will be given to the model. The model components will be Simulink instantiations of the elements of the model:



The build script will also construct an interface for changing system parameter values and injecting faults:



<Initial Configuration>

ThreeTank/ThreeTank/Tank1/C

Element Type:	Fault Profile:	Nominal Value:	Initial Value:	Fault Injection Time:	Fault Magnitude:
C	Nominal	1	10	0	0

ThreeTank/ThreeTank/Tank1/R

Element Type:	Fault Profile:	Nominal Value:	Initial Value:	Fault Injection Time:	Fault Magnitude:
R	Nominal	1	N/A	0	0

ThreeTank/ThreeTank/InputFlow/Sf

Element Type:	Fault Profile:	Nominal Value:	Initial Value:	Fault Injection Time:	Fault Magnitude:
MSf	Abrupt	N/A	N/A	0	0

ThreeTank/Thre ultInject

Element Type:		Noise Mean:	Noise Variance:	Fault Injection Time:	Fault Magnitude:
Sensor		0	0	0	0

ThreeTank/Thre ultInject

Element Type:	Fault Profile:	Noise Mean:	Noise Variance:	Fault Injection Time:	Fault Magnitude:
Sensor	Nominal	0	0	0	0

The interface will open when the model is opened, and close when the model is closed.

Running the Simulink Model

To run the Simulink model, simply open the created model and press the run button. The data structure script must be in the same directory as the model for it to execute successfully.

Fault Injection

Faults can be injected into the system by using the fault injection interface, and entering the desired fault information. After entering changes, hit the “Apply All Changes” button to apply the changes to the model. The “Reset to Nominal” button resets all fault profiles to nominal and applies the changes to the model.

The fault profile of a parameter (for Se, Sf, R, C, I, TF, GY, and constants) can be selected as either Nominal, Abrupt, Incipient, Bias, and StuckAt. Bias and StuckAt are available only for Se and Sf.

Select from the drop down list the desired fault profile, and enter injection time and fault magnitude, slope, bias, or stuck value. By default, all parameters are set with a nominal profile.

For sensor faults (for sensors represented using PlantOutputs), you can adjust the mean and variance of the Gaussian noise that is added to the true value. You can also inject “noise faults” which change the variance of the noise.

Fault injection is also possible for junctions controlled by PlantInputs or BGConstants. Corresponding fault injection blocks will be created that can be used to alter the switching input.

The tables below describe the parameters that can be set.

Fault Profile	
Nominal	No fault.
Abrupt	Instantaneous (step) change in the system parameter value.

Incipient	Gradually changing parameter value over time. Approximated as a linearly changing system parameter over time with a constant slope.
Bias	Instantaneous (step) change in the system parameter value.
StuckAt	Parameter or input value becomes stuck at a specified value.
Noise	For sensor faults, multiplies the additive noise by a constant factor.

Fault Value	
Fault Magnitude (for Abrupt Faults)	The multiplicative change in the magnitude of the nominal parameter value (can be positive or negative). The faulty value is the nominal parameter value plus one, multiplied by the entered value.
Fault Magnitude (for Noise Faults)	The multiplicative factor by which the noise is amplified.
Fault Slope (for Incipient Faults)	The rate of change in the magnitude of the nominal parameter value (can be positive or negative) over time. The slope is the entered value.
Fault Bias (for Bias faults)	The additive change in the magnitude of the nominal parameter value (can be positive or negative). The entered value is added to the nominal parameter value.
Fault Stuck Value (for StuckAt faults)	The value the parameter gets stuck at. The nominal parameter value becomes this value and stays at this value.

Fault Injection Time
The time instance when the fault is injected in the system

The fault injection interface also supports saving and loading configurations. Hit the “Save Configuration” button to save the current settings of the interface to a .mat file. Hit the “Load Configuration” button to load previously saved settings. They must be a configurations file from the same model.