Simulink: Useful Things to Know

George Hines
CDS 101/110a

1 What is Simulink?

Simulink is a modeling environment in which systems are represented as block diagrams, which are most often a convenient way to show process actions and interactions. In this course, we require only a basic ability to construct block diagrams, but if you choose to explore Simulink's capabilities further, it can also interface with C, Fortran, and Matlab m-file scripts (at this stage this is neither necessary nor recommended). The computation underlying Simulink models is handled by the set of solvers included in the Matlab package.

2 Useful Blocks

The Simulink Library contains all the blocks that are available with which to build your models. The library should open by default when you open Simulink, but if it is not visible for any reason, it can be brought up from any open Simulink window from the View menu. To create a new model, select New→Model from the File menu of any open Simulink (or Matlab) window. The following table contains a list of many blocks that you will find useful in this course. It is by no means comprehensive, but it will get you started.

Block Name	**Function**	**Library Location**
Constant	generates a constant reference of the specified value	Sources/Commonly Used Blocks (CUB)
Unit Step	generates a step function of the specified magnitude at the specified time	Sources
Sine Wave	generates a sinusoid reference of the specified amplitude and period	Sources
Ramp	generates a linearly increasing reference with the specified slope at the specified rise time	Sources
In1	used in a subsystem; creates an input port on the subsystem's block in the parent model	Sources/CUB
Scope	plots its reference input against time	Sinks/CUB
Out1	used in a subsystem; creates an output port on the subsystem's block in the parent model	Sinks/CUB
Sum	performs addition or subtraction (specified) on the reference signals	Math Operations/CUB
Gain	multiplies the reference signal by the specified value	Math Operations/CUB
Add/Substract	alternate form of Sum	Math Operations

Multiply/Divide	performs scalar or matrix multiplication or division (specified) of the reference signals	Math Operations/CUB
Sin	calculates the specified trigonometric function of the reference signal	Math Operations
Unary Minus	negates the reference signal	Math Operations
Derivative	differentiates the reference signal (continuous)	Continuous
Integrator	integrates the reference signal (continuous)	Continuous/CUB
State Space	computes the system response of a model represented in state-space form	Continuous
Transfer Fcn	computes the system response of a model represented as a transfer function	Continuous
Subsystem	allows creation of a model that can be used as a block in a different model	Ports and Subsystems/CUB
Manual Switch	selects between reference signals by mouse click	Signal Routing

3 Specifying Parameters

In the above list of useful blocks, any value that can be "specified" is set in the Block Parameters dialog box for that block. The block parameter dialog is opened by double-clicking on the block. Parameters that may not be adjusted are grayed out. As an example, see Fig. 1. Note that all parameters of this block are currently adjustible. To specify parameters for the entire simulation, such as the duration of the simulation,

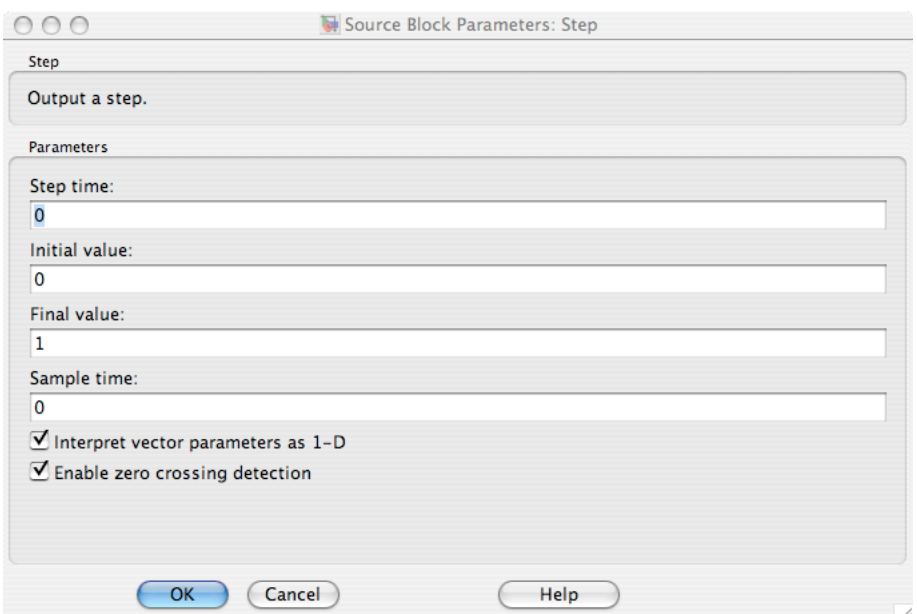

Figure 1: Parameter dialog box for the Unit Step block

the solver type, or the step size, open the Configuration Parameters dialog (Fig. 2) in the Simulation drop-down menu.

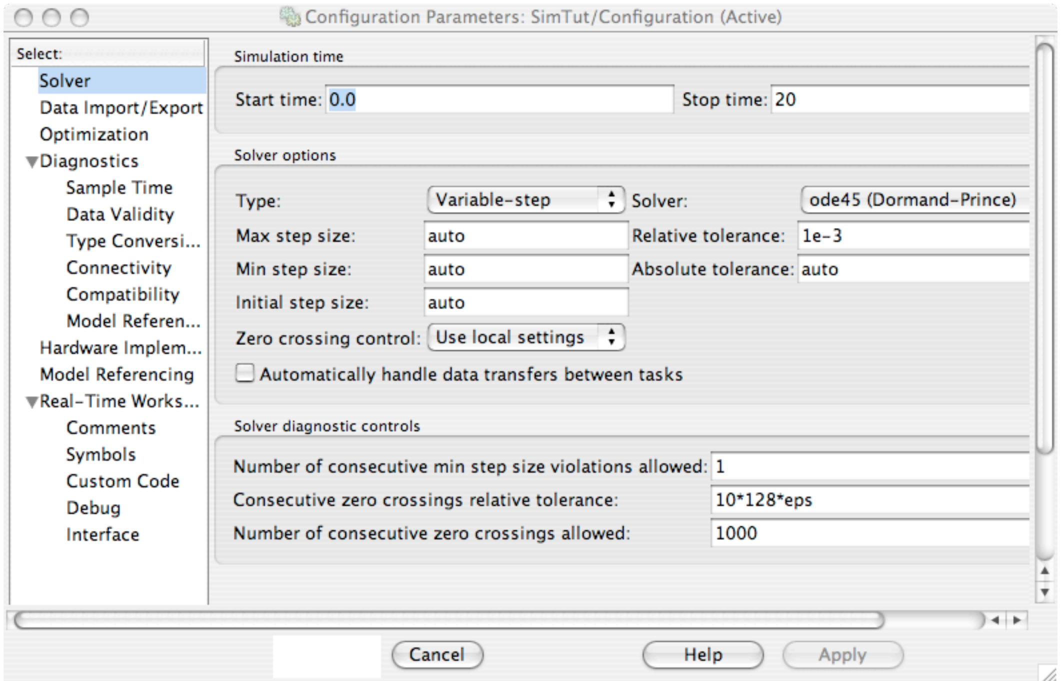

Figure 2: Configuration Parameters dialog box

4 Style

It is imperative to make block diagrams resemble the logical flow of information through the actual system. A few general guidelines are:

- DON'T cross connections
- DON'T change the dominant direction of a connection unless it's a feedback path
- DO name connections with the signal they carry
- DO make subsystems for more complicated components

To help avoid changing the dominant direction of a connection, Simulink lets you flip blocks so that the input is on the right and the output is on the left. You may also rotate them by 90 degrees left or right in the unlikely event that you need a vertical connection to a block. Don't overuse these tools...it's been my experience that they're rarely needed in basic models.

Generally, for the purpose of making subsystems, a "complicated component" is one that involves a lot of arithmetic that is essentially clutter at the systems level. For instance, if you have a nonlinear plant with the process equations laid out in block form, make a subsystem so that in the parent model there is one block called "Plant". (Of course, turn in printouts of all the subsystems if asked to show your diagram.)