MATLAB TRAINING SESSION V

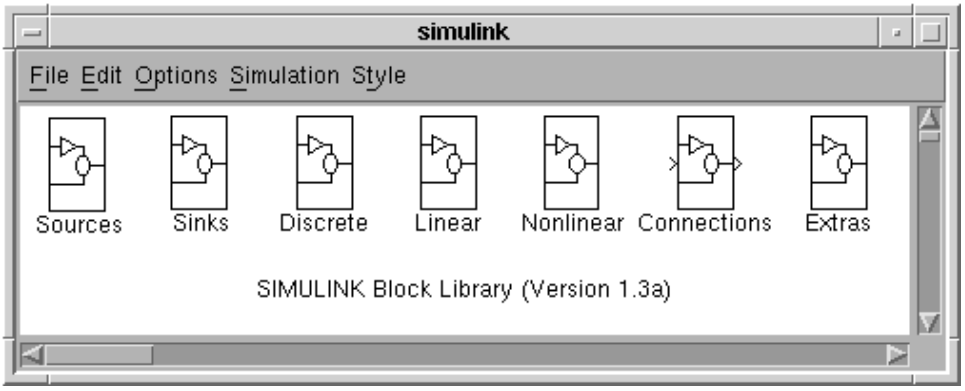
SIMULINK

SIMULINK™ is a program for simulating dynamic systems.As an extension to MATLAB, SIMULINK adds many features specific to dynamic systems while retaining all of MATLAB's general purpose functionality. SIMULINK has two phases of use: model definition and model analysis. A typical session starts by either defining a model or retrieving a previously defined model, and then proceeds to analysis of that model. These two steps are often performed iteratively until the model achieves the desired behavior.

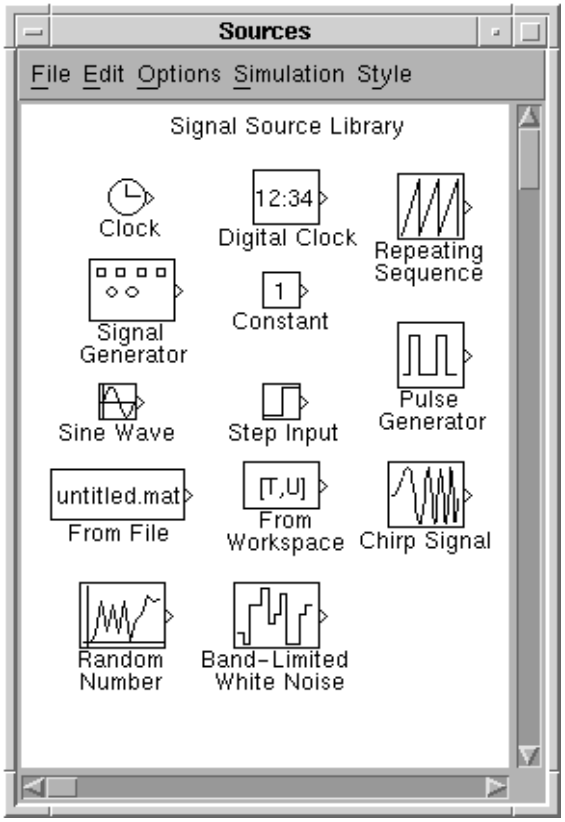
To facilitate model definition, SIMULINK adds a new class of windows called *block diagram* windows. In these windows, models are created and edited principally by mouse driven commands. Part of mastering SIMULINK is to become familiar with the manipulation of model components within these windows. After you define a model, you can analyze it either by choosing options from the SIMULINK menus or by entering commands in MATLAB's command window. Built-in analysis tools include various simulation algorithms, `linmod`, a tool for extracting linear models of systems, and `trim`, a tool for finding equilibrium points.

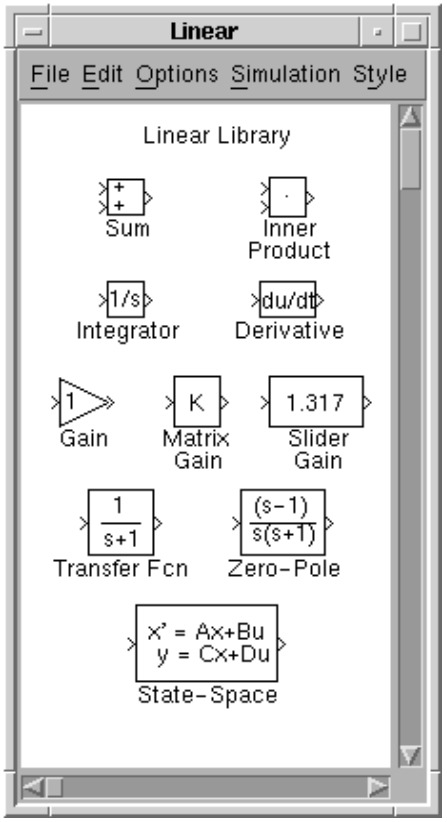
Constructing a Simple Model:

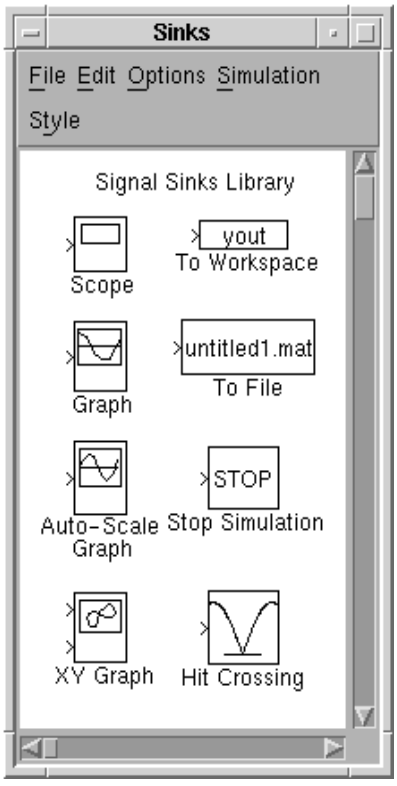
At the MATLAB prompt type >>simulink. This command displays a new window containing icons for the subsystem blocks that make up the standard library. These subsystems can be opened (by double-clicking) to produce the windows containing the prototype blocks to be copied into your models. Click on File and New, then move the window to a comfortable position.

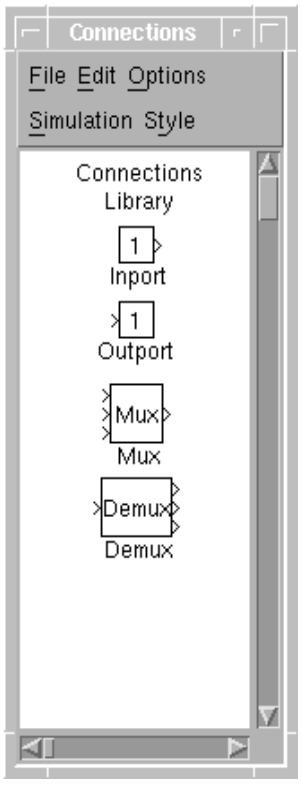


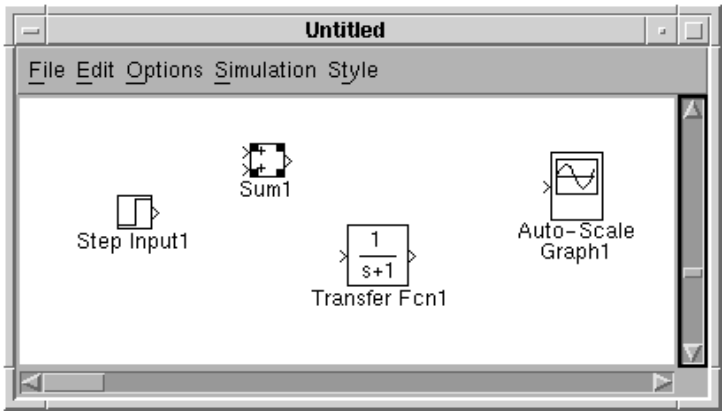
Open `Sources`, `Sinks`, `Linear`, and `Connections` by double-clicking on the icon with the left mouse button. Move the windows to a comfortable position. Blocks can be copied from one window to another by dragging them from the original location to the new location by holding down the left mouse button. Assemble the following diagram in your working window (page 3).
[Step Input -> Sources; Sum & Transfer Fcn -> Linear; Auto-Scale Graph -> Sinks]

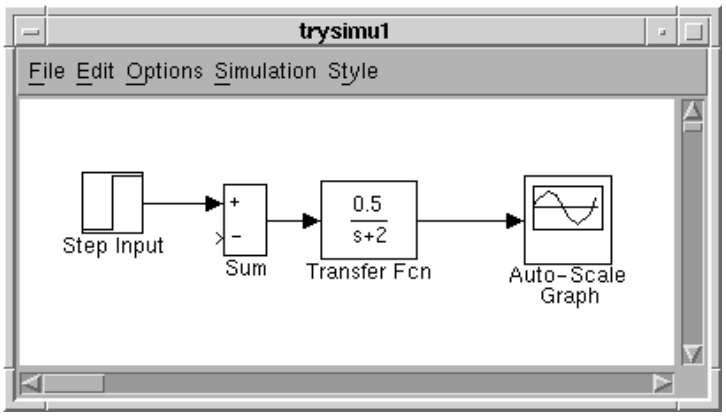

Sources
File Edit Options Simulation Style
Signal Source Library
Clock
12:34
Digital Clock
Repeating Sequence
Signal Generator
1
Constant
Pulse Generator
Sine Wave
Step Input
untitled.mat
From File
[T,U]
From Workspace
Chirp Signal
Random Number
Band-Limited White Noise

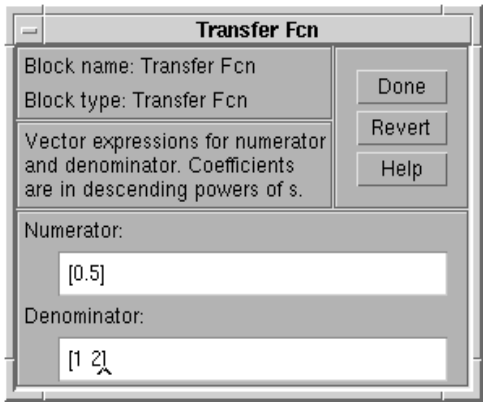

Linear
File Edit Options Simulation Style
Linear Library
Sum
Inner Product
1/s
Integrator
du/dt
Derivative
1
Gain
K
Matrix Gain
1.317
Slider Gain
1
s+1
Transfer Fcn
(s-1)
s(s+1)
Zero-Pole
x' = Ax+Bu
y = Cx+Du
State-Space


Sinks
File Edit Options Simulation Style
Signal Sinks Library
Scope
yout
To Workspace
Graph
untitled1.mat
To File
Auto-Scale Graph
STOP
Stop Simulation
XY Graph
Hit Crossing

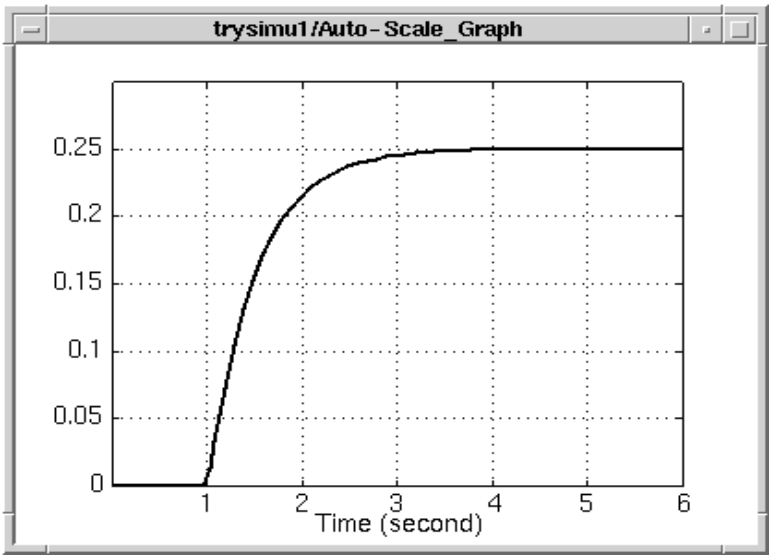

Connections
File Edit Options Simulation Style
Connections Library
1
Inport
1
Outport
Mux
Mux
Demux
Demux



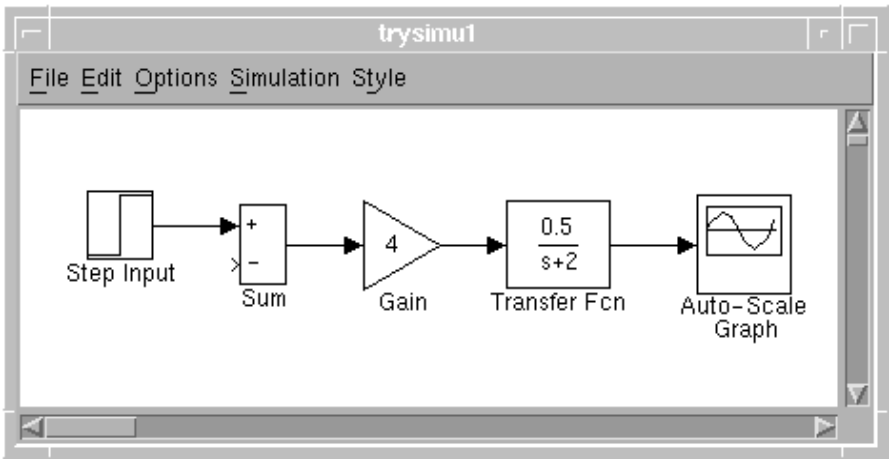


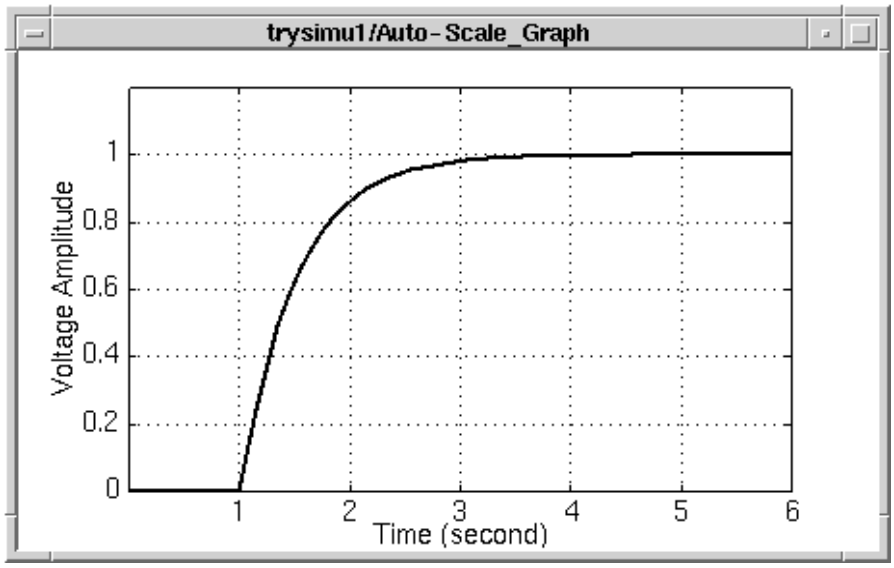


(1) Enlarge the blocks, arrange into order, and wire them together as shown. Blocks are expandeded by dragging the corner indicators outward with the mouse; To wire a block, drag the block's output arrow to the input arrow of the next icon using the left mouse button.
(2) Double click on the Transfer Fcn block and change the coefficients
(3) Click on File, then Save and name your model trysimu1
(4) Click on Simulation, then Parameters and set Stop Time to 6 and Max Step Size to 0.1.
(5) Click on Simulation, then Start and observe the trace on the Auto-Scale Graph.
*Notice the exponential response with maximum value at 0.25 (is this what we expect?).

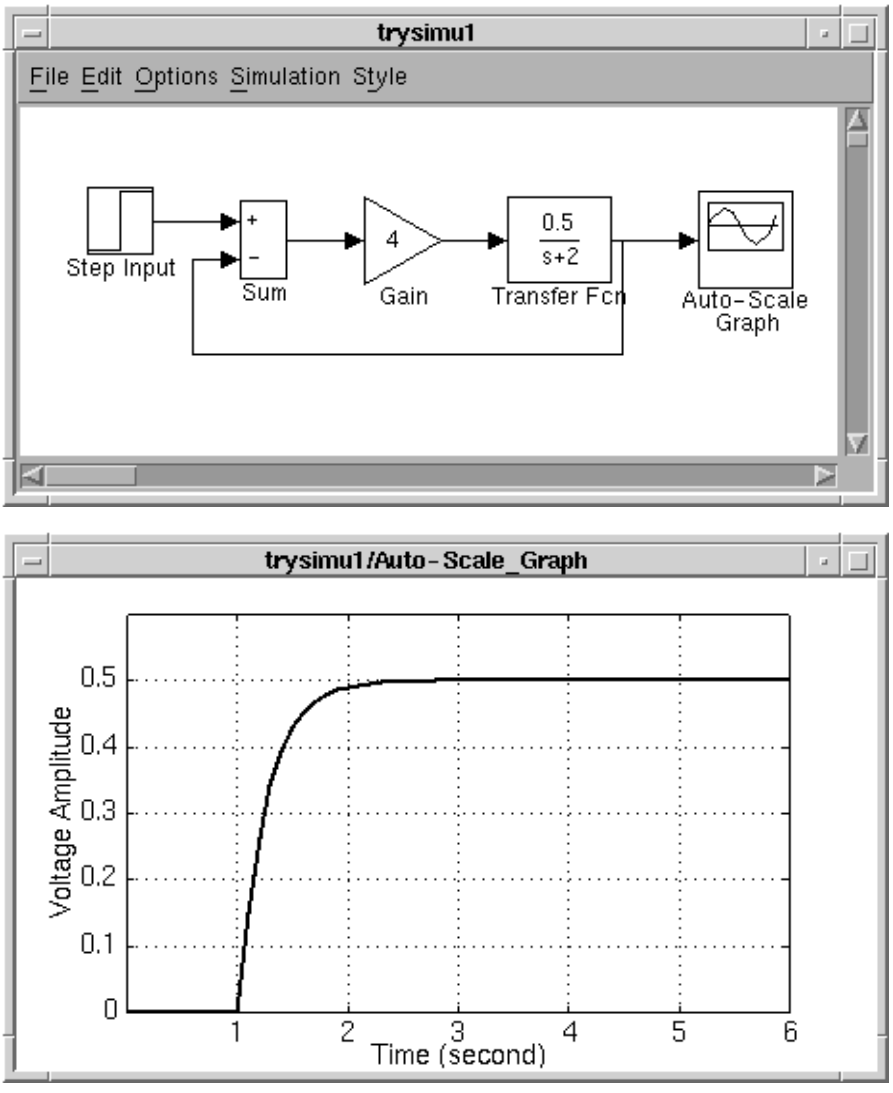


(1) Now modify the model to the one shown below.
(2) Double click on the Gain block and set the gain to 4.
(3) Click on Simulation, then Start and observe the trace on the Graph window.
Notice the exponential response with maximum value at 1.0.
(4) Save the model.





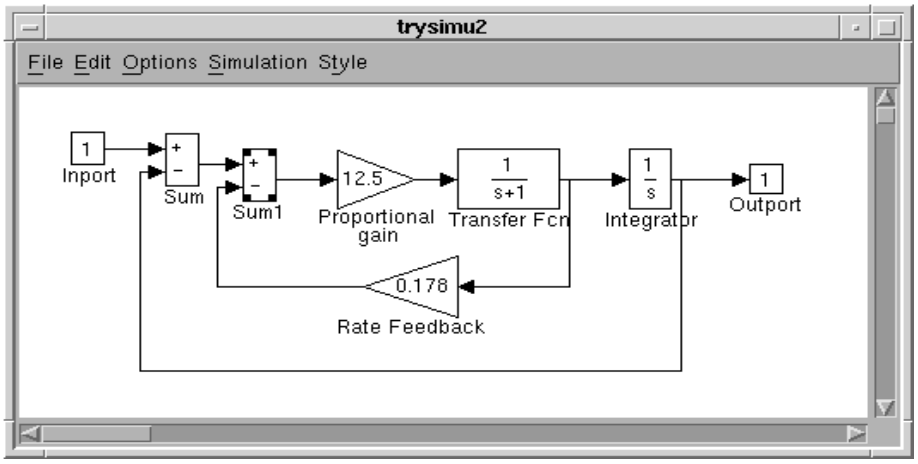
(1) Now modify the model to the closed-loop feedback control system shown below.
(2) Double click on the Sum block and set the second symbol to - (minus sign).
(3) Wire the feedback:
Get a tap off the Transfer Fcn wire with a right-click-hold-drag and pull it down below the icon. Point left-click-hold-drag to add a new wire to the head of the feedback arrow and move left under the Sum block. Repeat this procedure to go up then right into the - input of the sum.
(4) Click on Simulation, then Start and observe the trace on the Graph window.
*Notice that with feedback we have increased the speed of the response
(5) Save the model.



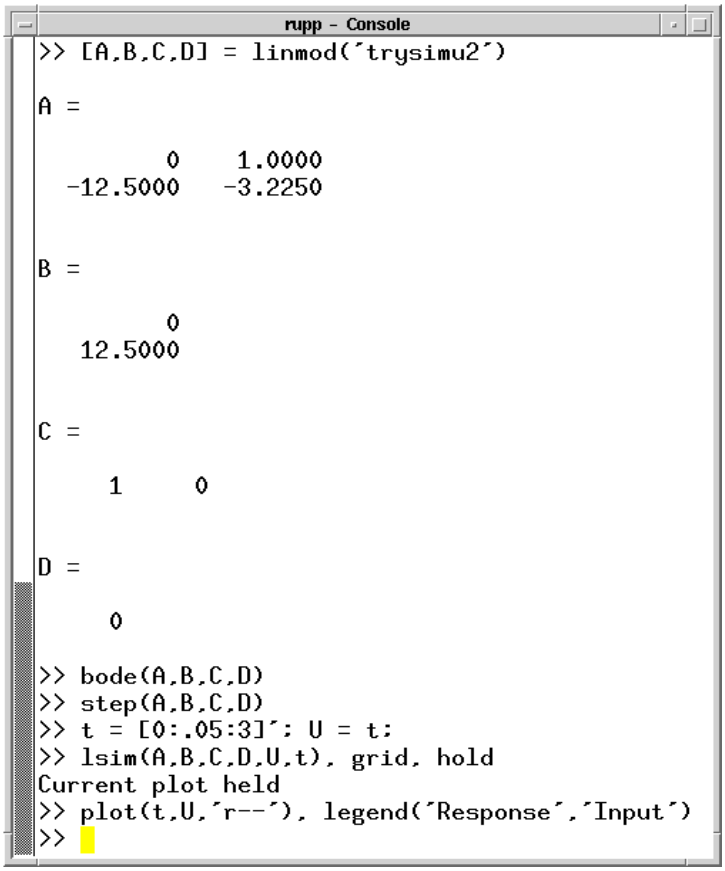
The model is saved , we can return to it another day by:
(1) Click on File, then Quit MATLAB.
(2) Restart MATLAB
(3) At the MATLAB prompt type >>trysimu1: Start a simulation as was done above.
(4) Close trysimu1 and start simulink for the next example: `>>simulink`.

Using the MATLAB Commands:

Suppose we are interested in investigating the frequency response characteristics and the step and ramp response of the following servo-system.



(1) Build the model shown above.
Change block names with a click on the block name, type the new name, and finally click on white space.
Click to highlight the Rate Feedback gain block and use the Flip command from the Options menu.
(2) Save the model as **trysimu2** and then close the file.
(3) Build a state space model of the system at the MATLAB command window using `linmod`
(4) Determine the frequency response characteristics using `bode`
(5) Inspect the step response of the system using `step`
(6) Inspect the ramp response of the system using a ramp input and `lsim`

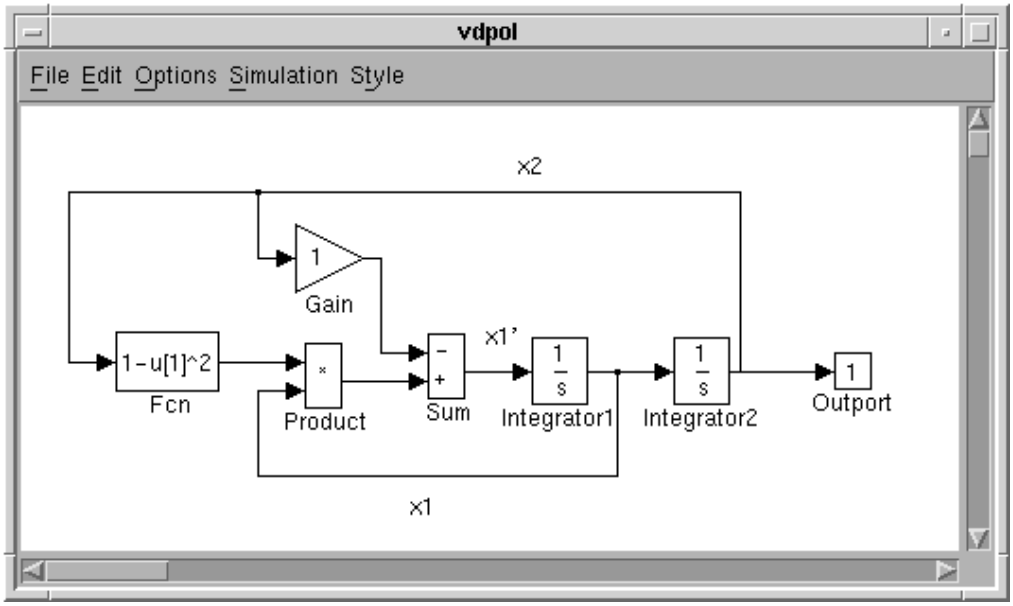


A Nonlinear Model:

$$Consider \qquad \ddot{x} + (x^2 - 1)\dot{x} + x = 0$$

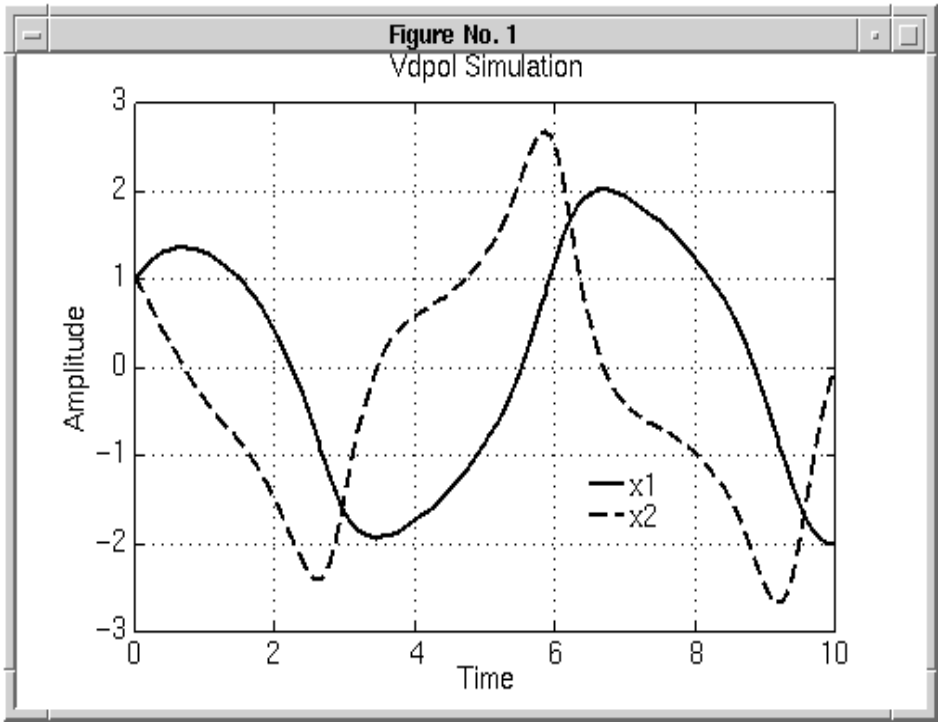
$$Let \qquad x_1 = \dot{x} \qquad x_2 = x$$

$$Then \qquad \dot{x}_1 = x_1(1 - x_2^2) - x_2 \qquad and \qquad \dot{x}_2 = x_1$$



(1) Build the model shown above.
Get the Integrators, Gain, and Sum blocks from the Linear library
Get the Outport block from the Connections library
Get the Fcn and Product blocks from the Nonlinear library
Double-click and change Fcn to 1-u[1]^2 and Sum to -+
Get a tap off existing wires by point-click-dragging with the right mouse button
Labels are added by point-clicking on the text location and then typing the text
(2) Save the model as vdpol
(3) We will use the built-in `*linsim` integrator to simulate the response for this system
*Other options include rk23, rk45, adams, gear, and euler
(4) At the MATLAB command prompt:

```
rupp - Console
>> tol = 1e-3;
>> minstep = 1e-5; maxstep = 1;
>> options = [tol, minstep, maxstep];
>> tf = 10; x0 = [1;1];
>> [t,x] = linsim('vdpol',tf,x0,options);
>> plot(t,x(:,1),t,x(:,2),'--'), grid
>>
```



PRINTING THE MODEL:

A graphics file of the block diagram model can be created using the same print command used with graphs.

`print -s[`*`model_name`*`] -d[graphics_format]` *`file_name`*

1. The model name should be the current model window you want to print.
   The print command creates a graphics file specified with the default format being PostScript
2. The user sends the graphics file to the printer from the operating system prompt

Printing Example:

(1) In MATLAB type

```
>>print -svdpol vdpol
```

(2) In Unix type

```
SunOS/boswell 3% lp -deb119_ps1 vdpol.ps
```

Importing into FrameMaker:

(1) In MATLAB type

```
>>print -svdpol -deps vdpol
```

(2) In FrameMaker use

File -> Import -> File -> vdpol.eps -> Set