Introduction to J-DSPC and Control blocks

J-DSP is an object-oriented Java™ tool, where J-DSP stands for Java Digital Signal Processing. J-DSP has been developed at Arizona State University (ASU) and is written as a platform-independent Java applet that resides either on a server or on a local hard-drive. It is accessible through the use of a web browser at http://jdsp.asu.edu. J-DSP has a rich suite of signal processing functions that facilitate interactive on-line simulations of modern statistical signal and spectral analysis algorithms, filter design tools, QMF banks, and state-of-the-art vocoders. J-DSP has recently been modified to allow control system simulations which are presented in this text. Due to the nature of control simulations however, the J-DSP blocks have been developed in a stand alone J-DSP version called J-DSPControl (J-DSPC).

J-DSPC provides a user-friendly environment through Java's graphical capabilities. Its highly intuitive graphical user interface (GUI) is easy to understand and use. All functions in J-DSPC appear as graphical blocks that are divided into groups according to their functionality. Selecting and establishing individual blocks can be done by a drag-and-drop-process. Connecting the blocks can also be done easily by dragging the mouse from one output pin to an input one. Figure 1 shows the J-DSPC editor's graphical environment.

Each J-DSPC block is linked to a control function. By connecting blocks together, a variety of control systems can be simulated while at the same time signals at any point of a simulation can be examined through the appropriate blocks. Blocks can be edited through dialog windows, allowing the user to change the corresponding function's parameters to desired values and/or to view results. Dialog windows appear when the user double clicks on the center of a block. Blocks can easily be manipulated (i.e. edit, move, delete and connect) within the specified drawing area, using the mouse.

For the original J-DSP version, all system execution is dynamic, which means that any change at any point of a system will automatically take effect in all related blocks. Any number of block dialog windows can be left open to enable viewing results at more than one point in the system. However, due to the nature of control simulations which most often requires feedback, all J-DSPC control blocks have been designed not to execute dynamically. Instead, the simulation can only be executed if the user places the distinct ***Control*** block shown in figure 2 on the workspace. By double-clicking on the ***Control*** block its dialog window appears along with the [Start Simulation] button. The user needs to press the button, after he/she has correctly created the control system to be simulated in J-DSPC. Once the button is pressed, the simulation executes and results can be viewed by means of blocks like the ***Plot*** block. Remember, the Plot block has to be connected prior to running the simulation otherwise it will not display anything. In adition, the user needs to make sure to always press the update button (present in all dialog windows) every time he/she modifies the simulation parameters so that these can take effect.

Here is the suggested process for creating and running a control simulation:

1. Place all the blocks on the workspace and arrange them to facilitate the connection process in step 2.
2. Connect all blocks as necessary. Modify the connections as needed in order to achieve a nice diagram.
3. Open the dialog boxes of the blocks where necessary and enter the required parameters. Press the [Update] button to enable them!
4. Create a ***Control*** block and double click on it in order to open its dialog window. Setup the simulation parameters as needed.
5. Press on the [Start Simulation] button to start the simulation.
6. Observe the simulation results, usually through the ***Plot*** block.
7. If needed to run a new simulation, simply change the block parameters or add/remove blocks and connections in the system. Repeat from step 5.

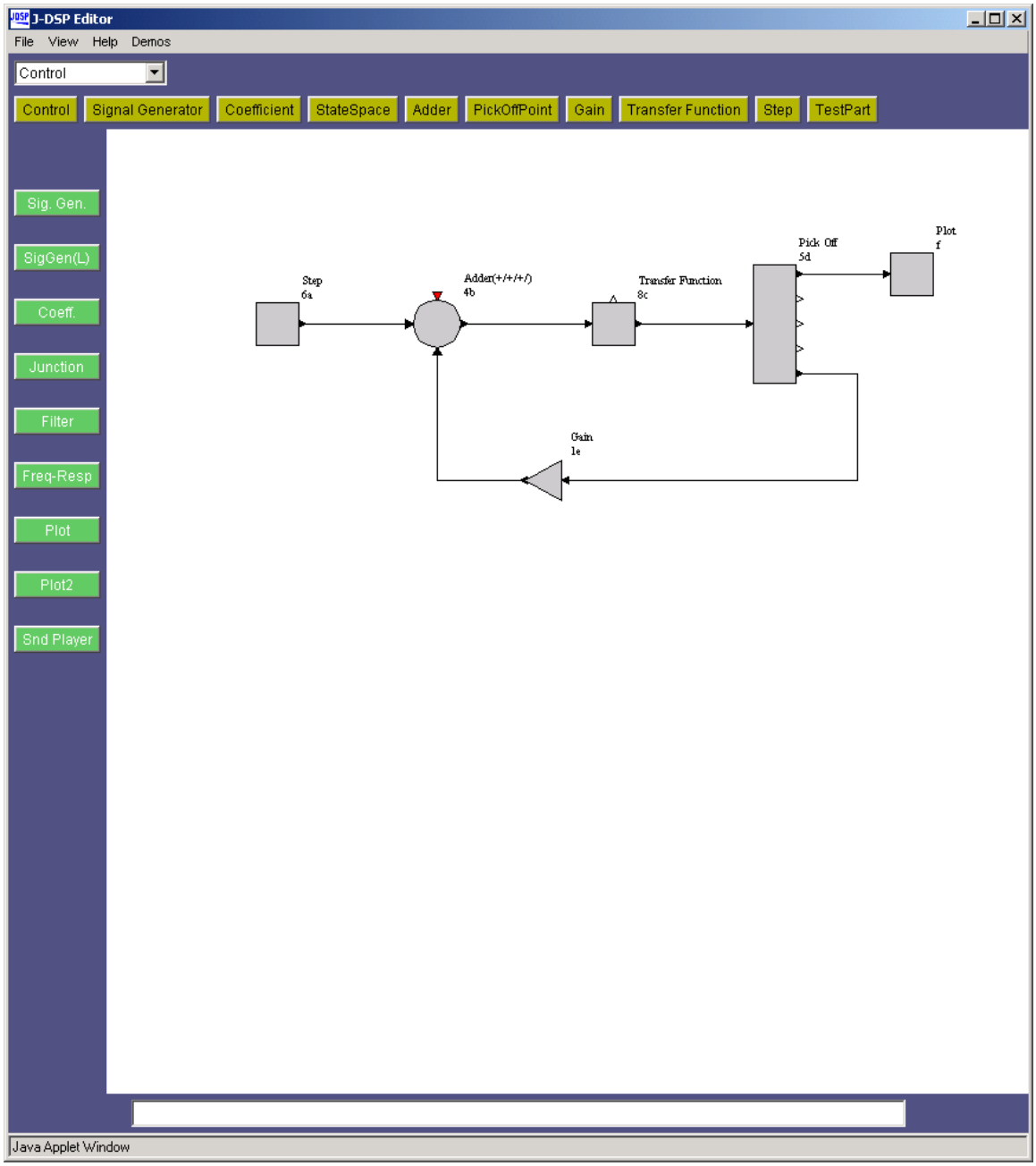

Figure 1: J-DSP Editor environment

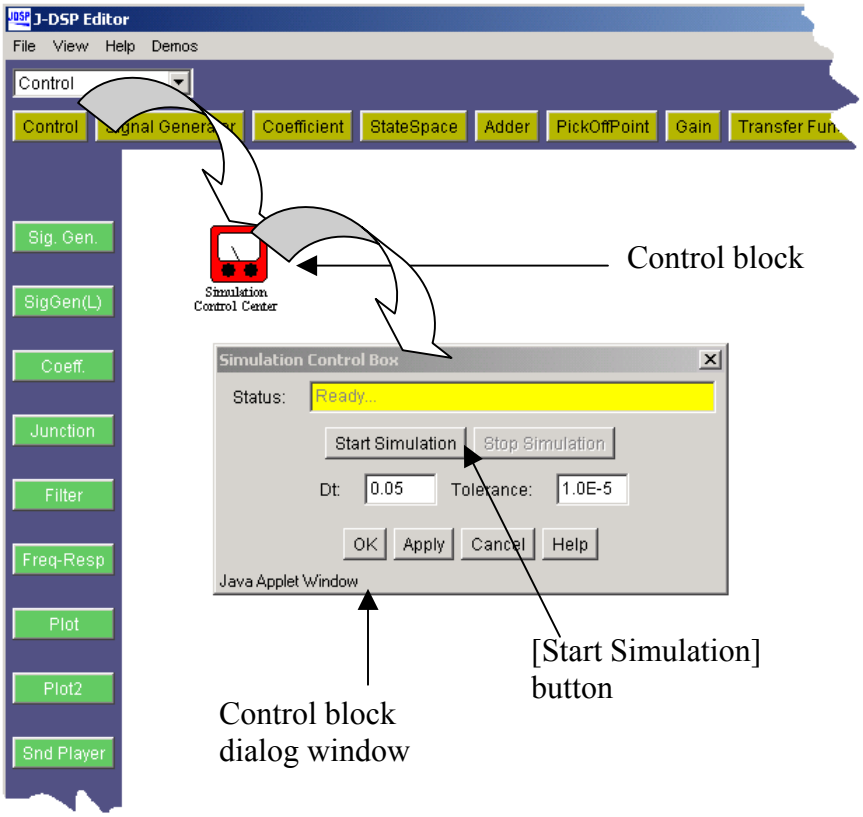

Figure 2: The ***Control*** block

In the new J-DSPC environment, the user has more access on manipulating blocks and connections, specifically developed with systems and controls in mind. Blocks can now be rotated by pressing the Ctrl-R combination of keys. Blocks can be flipped vertically by pressing Ctrl-T and horizontally by pressing Ctrl-E. A connection can be modified simply by dragging a part of it with the mouse. In order to create better looking diagrams, it is suggested that you first place the blocks at the correct location, rotate or flip them as needed while leaving the connections for last. J-DSP editor performs better if the diagrams are created in this order. In case a connection becomes messy during your attempt to modify it, it is also suggested that you delete it and create it again from scratch. Figure 3 shows a case where modifying a connection results in an improved diagram.

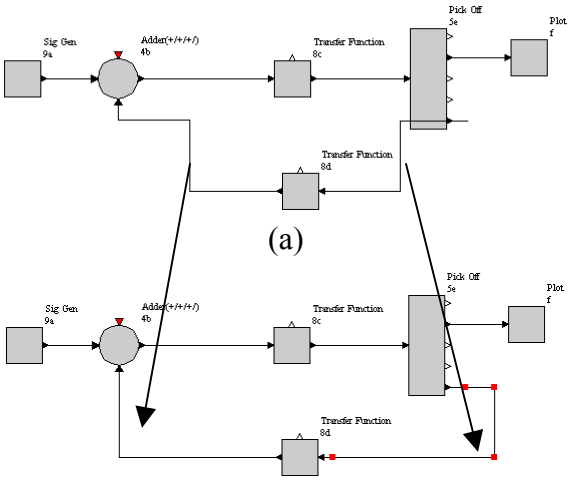

Figure 3*:* (a) Prior to modifying a connection (b) After modification

For more information on the basics of J-DSP, the student is invited to read the "Introduction to J-DSP" manual located at the J-DSP Editor's web site at http://jdsp.asu.edu/jdsp_manual.html.

Getting acquainted with J-DSPC: A first control systems simulation

Introduction

The best way to understand J-DSPC is through an example. Start J-DSPC and follow the instructions step by step to create your first simulation. **Remember to always press the [Update] button every time you modify the simulation parameters so that they can take effect!**

Experiment 1

This example simulates the step response of a system with transfer function

$$H_1(s)\frac{1}{s^2+0.3s+1}$$

Step 1: Place the ***Step***, ***Transfer Function*** and ***Plot*** blocks on the workspace. Connect them as shown in figure 1. Note that the ***Plot*** block is located in the group of permanent blocks to the left side of the J-DSPC window.

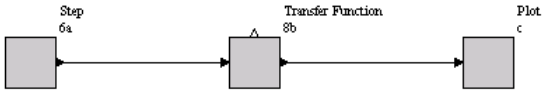

Figure 1: Setting up and connecting the blocks

Step 2: Double-click on the ***Transfer Function*** block to open its dialog and set the order to 2. Enter the transfer function coefficients as shown in figure 2. Press the [OK] button in order to set the new transfer function to the part and close the dialog window.

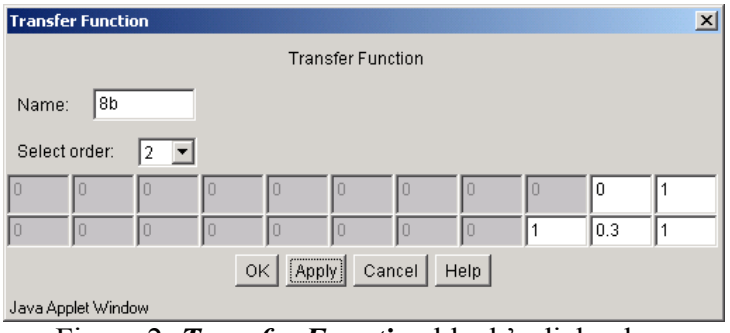

Figure 2: ***Transfer Function*** block's dialog box

Step 3: Double-click on the ***Plot*** block in order to open its dialog box which will display the simulation results.

Step 4: Create a ***Control*** block and place it on the workspace. Double click to open its dialog box. Your J-DSPC workspace should now be similar to figure 3.

Step 5: Set the *Dt* field in the ***Control*** block's dialog window to 0.1 and press the [Start Simulation] button. The simulation now executes and results are displayed in the ***Plot***

block's dialog box as shown in figure 4. You can experiment by varying the *Dt* field and re-running the simulation. Observe that the Dt field has an effect on the resolution of the output plot.

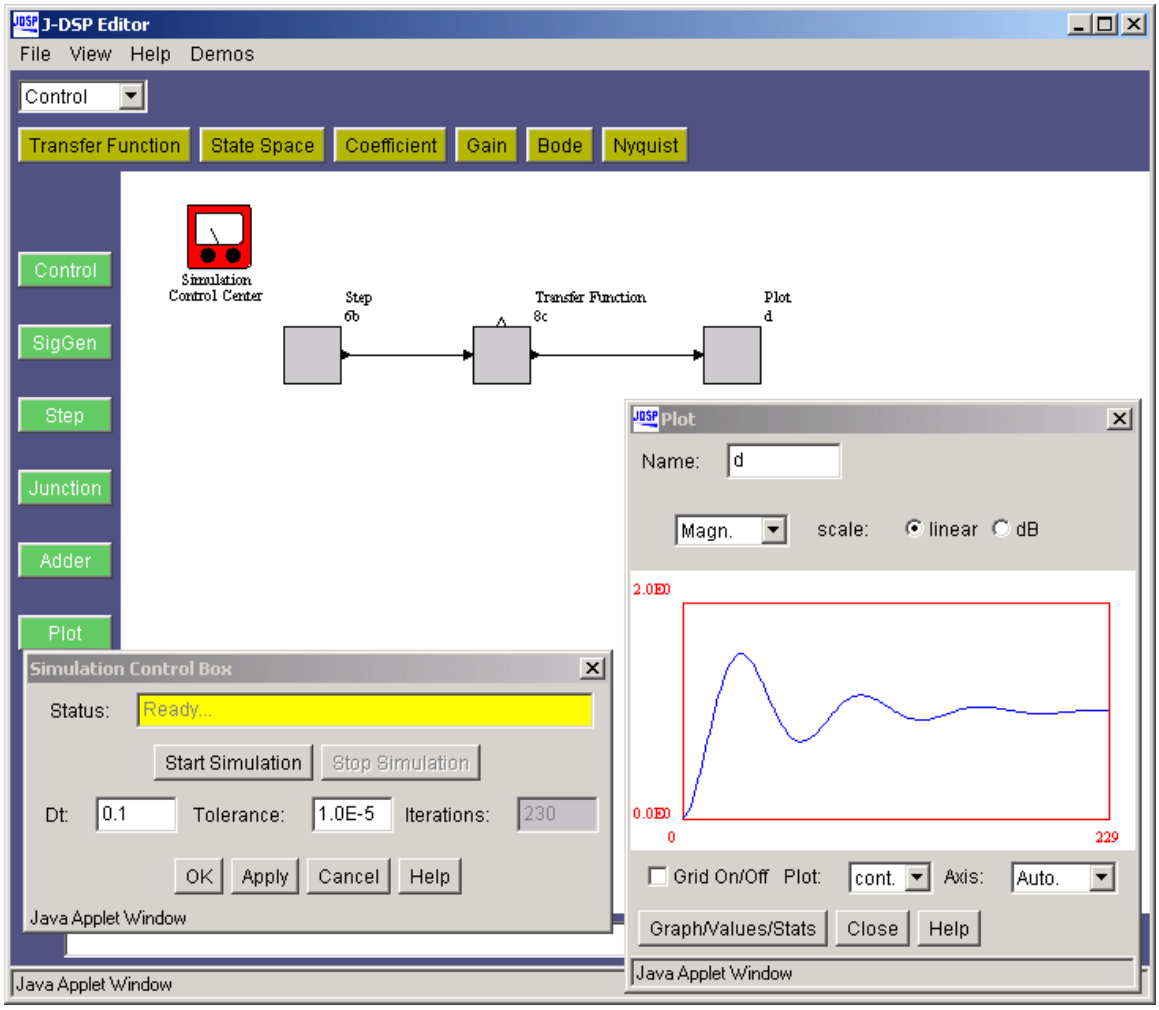

Figure 3: J-DSPC workspace after step 4.

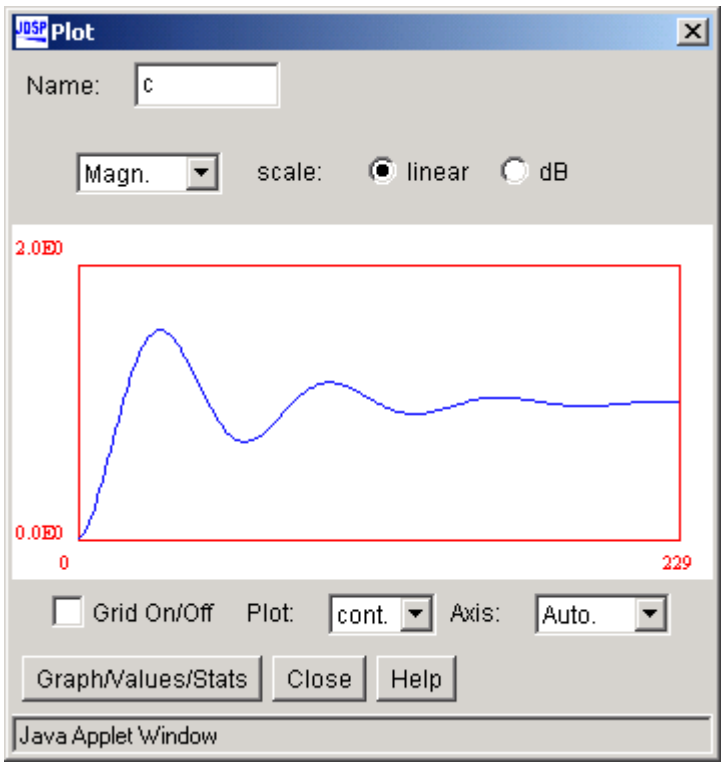

Figure 4: Simulation results with Dt = 0.1

You have now performed your first J-DSPC simulation. You can always experiment by changing the transfer function, adding more blocks or varying the simulation parameters. Here is another transfer function you might wish to try:

$$H_2(s) = \frac{1}{5s+1}$$

Experiment 2

This simulation uses two systems in cascade. Using similar procedures as in experiment 1, connect two ***Transfer Function*** blocks in series. Do not forget to include the ***Step*** and ***Plot*** blocks in your simulation.

Note: You can select *File* | *New* from the J-DSPC editor's main menu to clear up your workspace and create new blocks, or you can simply modify the existing system from exercise 1.

Create the flowgram for the simulation and enter the following transfer functions into the first and second transfer function blocks respectively. Your workspace should now be similar to figure 5.

$$H_1(s)\frac{1}{s^2+0.3s+1}\ , \quad H_2(s) = \frac{1}{5s+1}$$

Set the Dt fileld to 0.1 and run the simulation. Use the ***Plot*** block to examine the result which is shown here in figure 6.

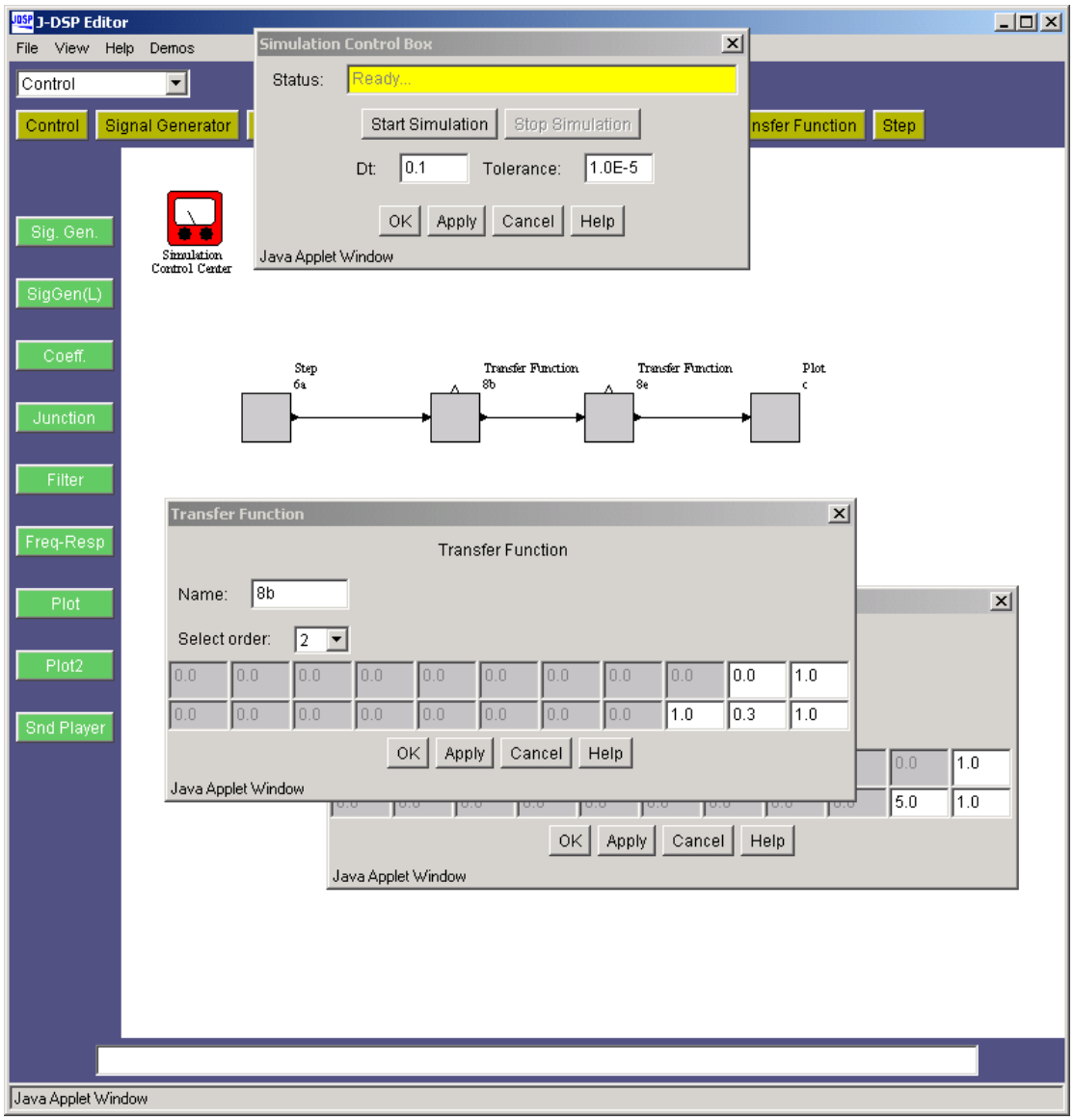

Figure 5: Workspace for experiment 2.

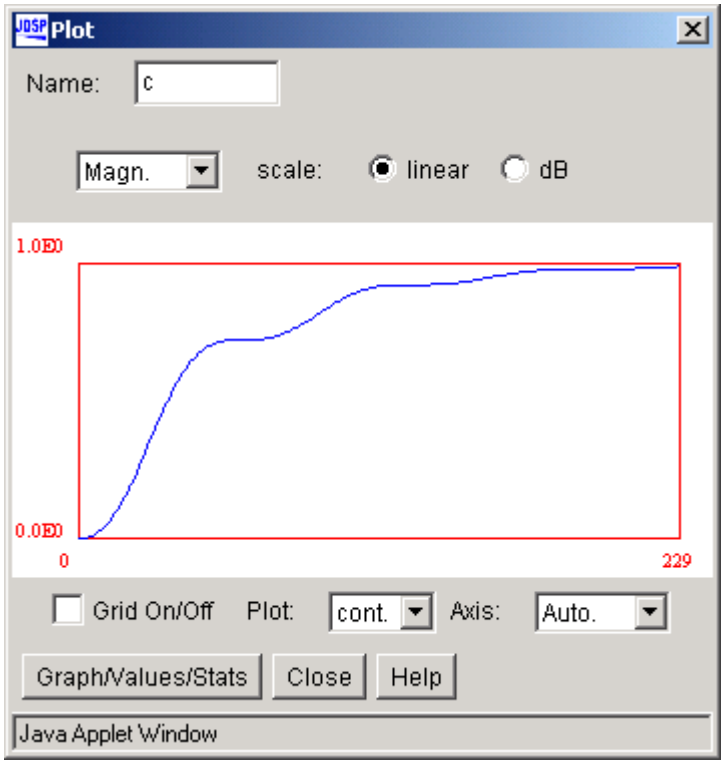

Figure 6: Simulation results for experiment 2.

Experiment 3

Here, the two systems used in experiment 2 remain in cascade, but unity negative feedback is also applied. Use an ***Adder*** and a ***Pick Off Point*** to setup the flowgram as shown in figure 7. Remember to set the feedback input of the ***Adder*** block to a negative value, otherwise the systems will become unstable. For this experiment, a good setting for *Dt* is 0.15.

Note: If the feedback connection is not drawn in an optimal way, remember that you can modify it by simply dragging it as desired.

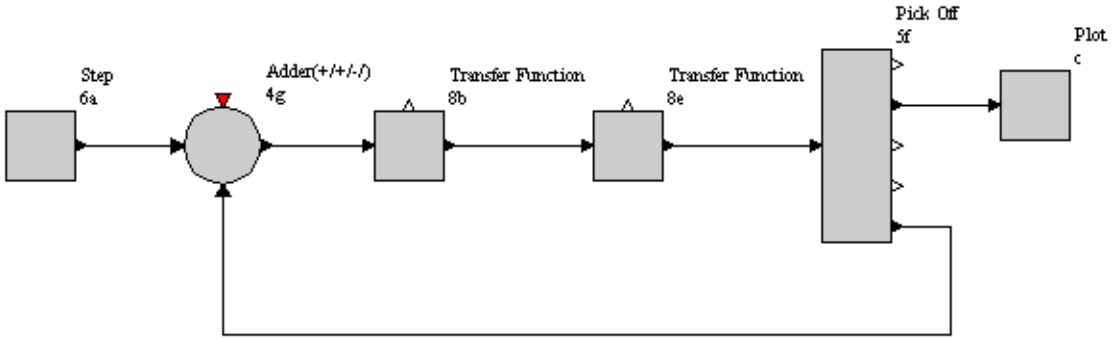

Figure 7: Unity feedback applied

Run the simulation after setting the *Dt* field in the ***Control*** block's dialog box to 0.15.The result is displayed in figure 8.

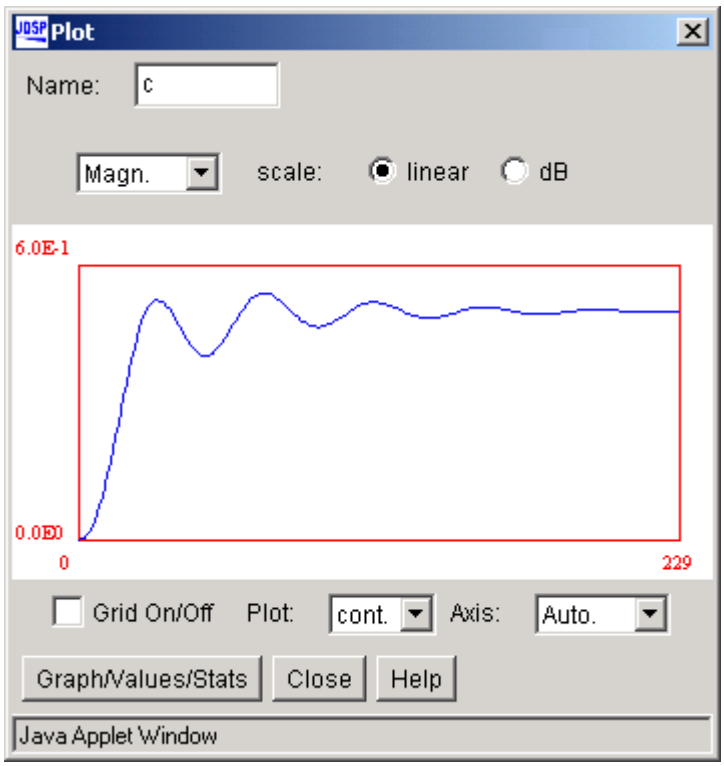

Figure 8: Simulation results for experiment 3

J-DSP Control Blocks

Block Name	Description
• ***Control***	Controls the simulation and its parameters
• ***Signal Generator***	Signal generator specific to control blocks
• ***State Space***	Simulates a system in state space form
• ***Coefficient***	Passes transfer function coefficients to the state space part
• ***Adder***	Adds or subtracts input signals
• ***Pick Off Point***	Propagates its input at its outputs
• ***Gain***	Multiplies a signal with a constant
• ***Transfer Function***	Simulates a system given a transfer function
• ***Step***	Generates a step sequence
• ***Bode***	Bode plot of a system

Control

Description: The ***Control*** block is a fundamental block used in all control simulations. It is not to be connected to any other blocks but is only to be used for setting up simulation parameters and starting the simulation.

Figure 1: ***Control*** block

Inputs: None

Outputs: None

Algorithm/Equation implemented: N/A

Dialog Box: By double-clicking on the ***Control*** block a dialog window appears as shown in figure 2. The following is a list of items found on this dialog box:

1. Status field: Indicates the simulation status.
2. [Start Simulation] button: Starts the simulation.
3. [Stop Simulation] button: Stops the simulation (Not functional yet).
4. Dt: Discretization time used in the simulation.
5. Tolerance: Set the error tolerance used to check convergence of the simulation.
6. [OK] button: Closes the dialog box and passes the parameters to the simulation.
7. [Apply] button: Passes the parameters to the simulation.
8. [Cancel] button: Closes the dialog box ignoring any changes made.
9. [Help] button: Displays the Help dialog box for the ***Control*** block.

Figure 2: ***Control*** block's dialog box

Signal Generator

Description: Generates a variety of time domain signals. It supports: pulses, triangular, delta, exponential, sinusoid, sinc, random, and user defined signals. The length of each signal, "pulse width" can be set, along with the "gain", the amplitude of the signal. The base of the exponential can also be varied. Random signals can have uniform, normal and Rayleigh distributions with variable mean and variance.

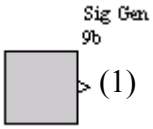

Figure 3: ***Sigal Generator*** block

Inputs: None

Outputs: (1) Generated signal (time domain)

Algorithm/Equation implemented: N/A

Dialog Box: By double-clicking on the ***Signal Generator*** block a dialog window appears as shown in figure 4. The following is a list of items found on this dialog box:

1. Name: Change the name of the block.
2. Signal: Select signal type.
3. Freq: Signal frequency.
4. Gain: Signal amplitude.
5. Periodic checkbox: select periodic or not.
6. Period: The period of a periodic signal.
7. Time Shift: The signal's time shift.
8. [Close] button: Close the window.
9. [Update] button: Pass the selected parameters to the part.
10. [Help] button: Displays the Help dialog box for the ***Signal Generator*** block.

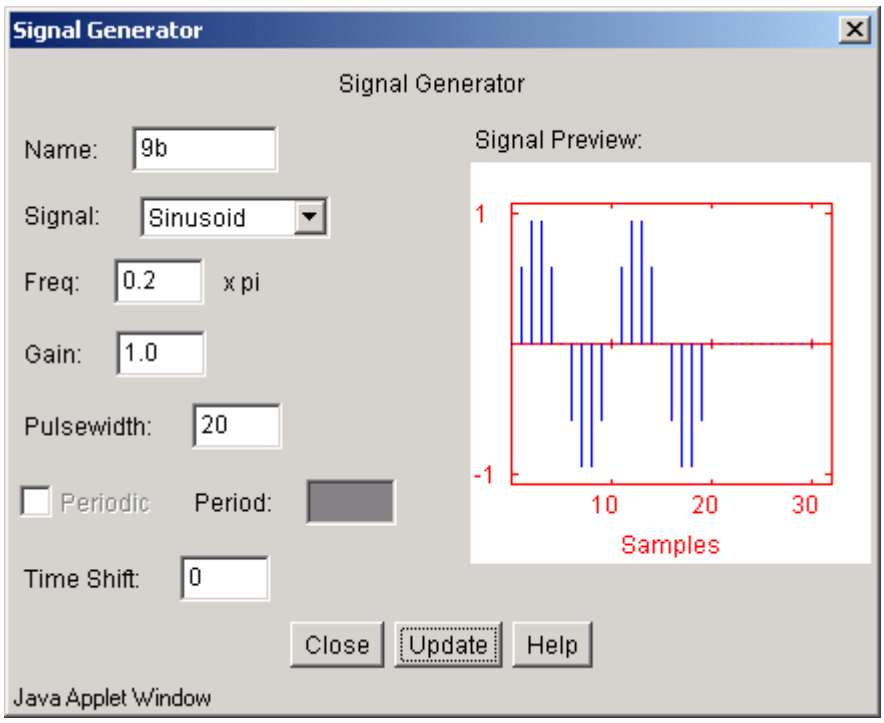

Figure 4: ***Signal Generator*** block's dialog box

State Space

Description: The State Space block simulates a system given a state space representation in the form of four matrices A, B, C, D. There is only need for input (1) and output (3) to be connected in order for the block to operate properly. Input (2) is optional. See the ***Coefficient*** block that follows for more information.

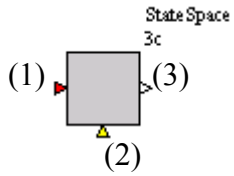

Figure 5: ***State Space*** block

Inputs: (1) Any signal (time domain), (2) Transfer function (coefficients).

Outputs: (3) System's response to input (1) (time domain).

Algorithm/Equation implemented:

$$\dot{\underline{x}} = A\underline{x} + Bu$$
$$y = C\underline{x} + Du$$

where $\underline{x}$ is the states vector, u is the input at pin (1) and y is the output at pin (3)

Dialog Box: By double-clicking on the ***State Space*** block a dialog window appears as shown in figure 6. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. States: The number of states in the system
3. Inputs: The number of inputs in the system. Currently only one input is supported.
4. Outputs: The number of outputs in the system. Currently only one output is supported.
5. [Load Matrix A], [Load Matrix B], [Load Matrix C], [Load Matrix D] buttons: Open up a new dialog box for loading the corresponding matrices into the system as shown in figure 3.
6. [Load Initial Conditions] button: Load initial conditions for the system. The default value is zero.
7. Select differentiation method: Provides a choice between forward Euler differentiation and backward Euler differentiation.
8. [OK] button: Close the window and pass the parameters to the part
9. [Apply] button: Pass the parameters to the part without closing the window
10. [Clear All] button: Clear all the matrices in the system
11. [Cancel] button: Close the window ignoring any changes
12. [Help] button: Displays the Help dialog box for the ***State Space*** block

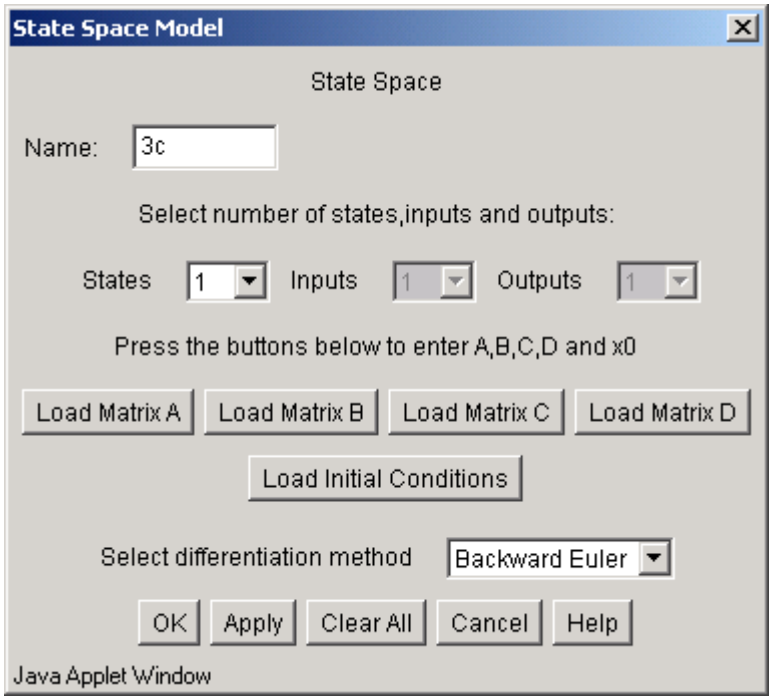

Figure 6: ***State Space*** block's dialog box

Matrix Input Dialog Box: By double-clicking on one of the four [Load Matrix X] buttons, a new dialog window appears as shown in figure 7. This dialog window is used for entering the matrix elements of the respective matrix X. You can choose the matrix type from a drop-down list of matrix types in order to simplify the matrix loading procedure. Press the [OK] button to save the matrix or the [Cancel] button to ignore changes. The [Update] button works only when the *Symmetric* option is selected from the drop down box and its function is to copy the matrix coefficients to their symmetric location

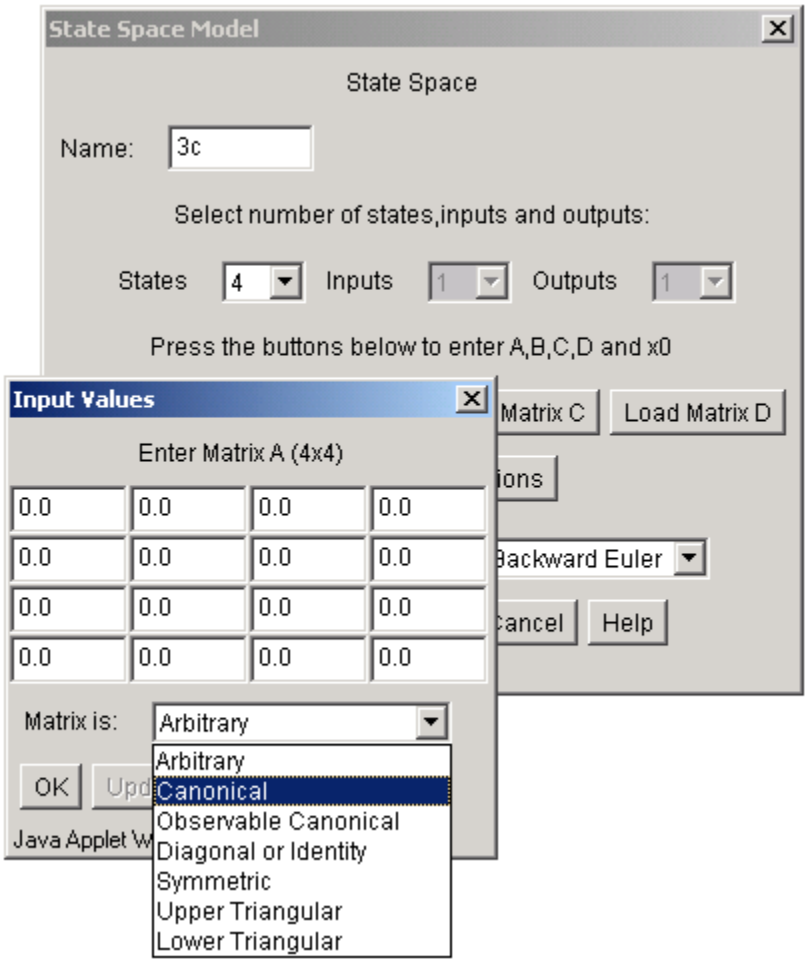

Figure 7: ***State Space*** block's dialog box with matrix input dialog box

Coefficient

Description: The State Space ***Coefficient*** block when connected to the ***State Space*** block's bottom input, serves to transform it into a regular transfer function block.

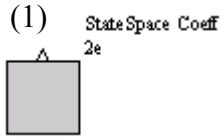

Figure 8: State Space ***Coefficient*** block

Inputs: None

Outputs: (1) Transfer function (coefficients).

Algorithm/Equation implemented (thought the ***State Space*** block):

$$H(s) = \frac{\sum_{i=0}^{L} b_i s^i}{\sum_{i=0}^{M} a_i s^i}$$

Dialog Box: By double-clicking on the State Space ***Coefficient*** block a dialog window appears as shown in figure 9. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. Select order: The order of the system to be passed to the ***State Space*** block
3. Input boxes: Enter the transfer function coefficients as they would appear in an equation of the form:

$$H(s) = \frac{\sum_{i=0}^{L} b_i s^i}{\sum_{i=0}^{M} a_i s^i}$$

4. [OK] button: Close the window and pass the parameters to the part.
5. [Apply] button: Pass the parameters to the part without closing the window
6. [Cancel] button: Close the window ignoring any changes
7. [Help] button: Displays the Help dialog box for the State Space ***Coefficient*** block

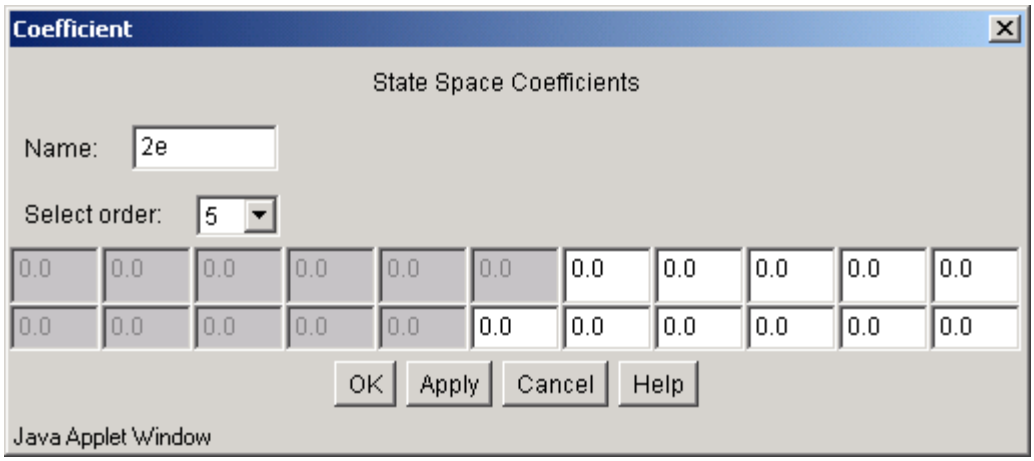

Figure 9: State Space ***Coefficients*** block's dialog box

Example: The following transfer function would be entered as shown in figure 10:

$$H(s) = \frac{2s^2 + 4s + 4}{s^3 + 3s^2 + s + 2}$$

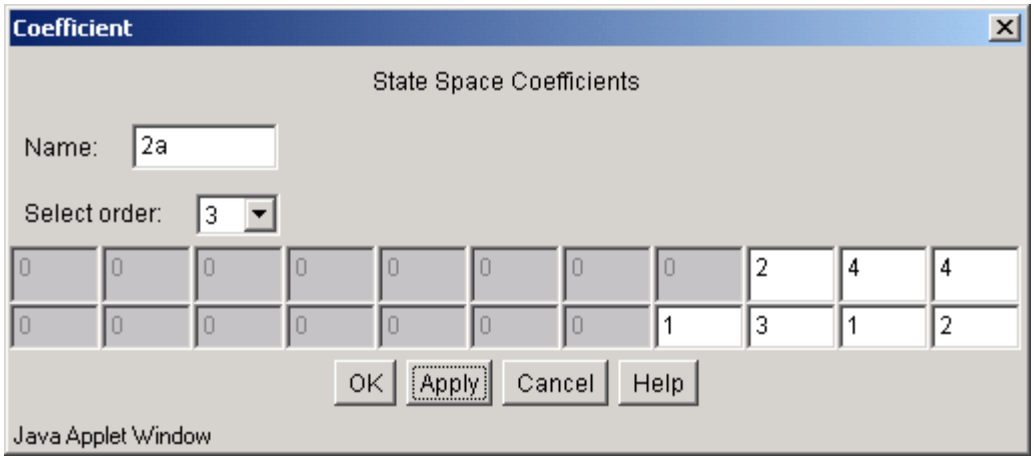

Figure 10: State Space ***Coefficients*** block example

Adder

Description: The ***Adder*** block adds or subtracts its input signals depending on the selection made in the part's dialog window. The block can change shape and number of inputs.

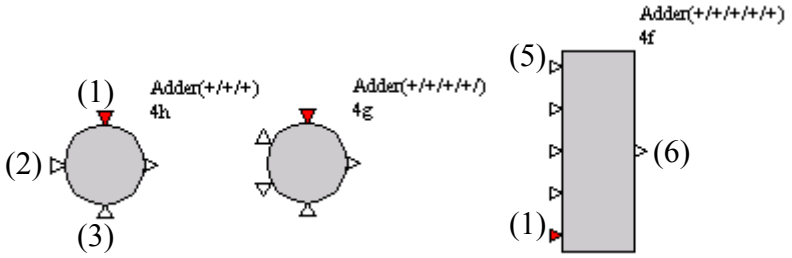

Figure 11: ***Adder*** block in various forms

Inputs: (1)… (5) Any signal (time domain).

Outputs: (6) Sum of input signals (time domain).

Algorithm/Equation implemented:

$$y = a_1 x_1 + a_2 x_2 + ... + a_5 x_5$$

Where x_1 to x_5 are the inputs, y is the output and $a_i = \begin{cases} +1 & for + \\ -1 & for - \\ 0 & no\_input \end{cases}$

Dialog Box: By double-clicking on the ***Adder*** block a dialog window appears as shown in figure 12. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. Input 1 to 5: The sign of the input.
3. [OK] button: Close the window and pass the parameters to the part
4. [Apply] button: Pass the parameters to the part without closing the window
5. [Cancel] button: Close the window ignoring any changes
6. [Help] button: Displays the Help dialog box for the ***Adder*** block
7. [Rotate] button: Rotate the block
8. [Flip Horiz] button: Flip the block horizontally
9. [Flip Vert] button: Flip the block vertically
10. Shape selection: Select between a square or a round block
11. Number of Inputs: Select the number of inputs for the block.

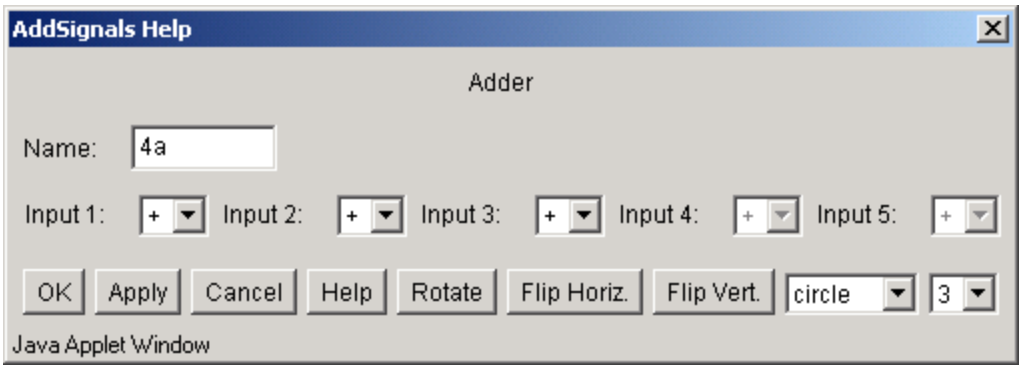

Figure 12: ***Adder*** block's dialog box

Note that the inputs are counted anticlockwise from the top when the adder has a circular shape, or from bottom to top when the adder has a rectangular shape.

Pick Off Point

Description: The ***Pick Off Point*** block simply passes its input to the outputs.

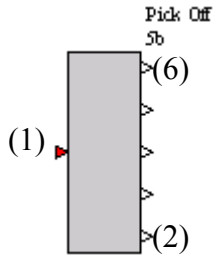

Figure 13: ***Pick Off Point*** block

Inputs: (1) Any signal (time domain).

Outputs: (2)… (6) Input signal (time domain).

Algorithm/Equation implemented:

$$y_i = x \text{ , } i \text{ from 2 to 6.}$$

where y is the output of pins (2) to (6) and x is the input of pin (1)

Dialog Box: By double-clicking on the ***Pick Off Point*** block a dialog window appears as shown in figure 14. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. [OK] button: Close the window and pass the parameters to the part
3. [Apply] button: Pass the parameters to the part without closing the window
4. [Cancel] button: Close the window ignoring any changes
5. [Help] button: Displays the Help dialog box for the ***Pick Off Point*** block

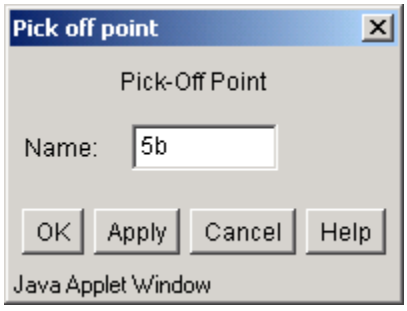

Figure 14 ***Pick Off Point*** block's dialog box

Gain

Description: The ***Gain*** block multiplies its input with a constant and passes it to the output.

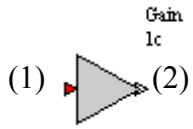

Figure 15: ***Gain*** block

Inputs: (1) Time Domain

Outputs: (2) Time Domain

Algorithm/Equation implemented:

$$y = ax$$

where x is the input, y is the output and a is the gain value as entered in the block's dialog box.

Dialog Box: By double-clicking on the ***Gain*** block a dialog window appears as shown on figure 16. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. Gain: Multiplication factor
3. [OK] button: Close the window and pass the parameters to the part
4. [Apply] button: Pass the parameters to the part without closing the window
5. [Cancel] button: Close the window ignoring any changes
6. [Help] button: Displays the Help dialog box for the ***Gain*** block

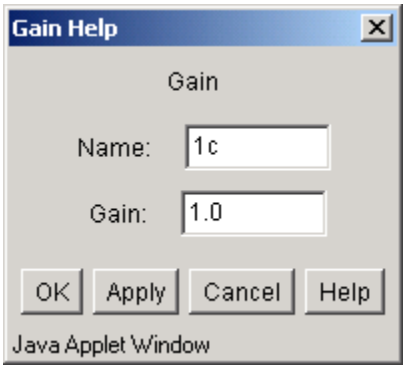

Figure 16: ***Gain*** block's dialog box

Transfer Function

Description: The ***Transfer Function*** block simulates a system given a transfer function in the form shown below. The top output can be connected to a ***Bode*** block in order to examine the system's frequency response.

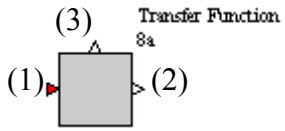

Figure 17: ***Transfer Function*** block

Inputs: (1) Any signal (time domain).

Outputs: (2) System's response to input signal (time domain), (3) Frequency Response (coefficients).

Algorithm/Equation implemented:

$$H(s) = \frac{\sum_{i=0}^{L} b_i s^i}{\sum_{i=0}^{M} a_i s^i}$$

where b_i and a_i are respectively entered as the top and bottom coefficients in the dialog box shown in figure 18

Dialog Box: By double-clicking on the ***Transfer Function*** block a dialog window appears as shown on figure 18. The following is a list of items found on this dialog box:

1. Name: Change the name of the block.
2. Select Order: Order of the system.
3. Input boxes: The coefficients a_i and b_i of the transfer function.
4. [OK] button: Close the window and pass the parameters to the part
5. [Apply] button: Pass the parameters to the part without closing the window
6. [Cancel] button: Close the window ignoring any changes
7. [Help] button: Displays the Help dialog box for the ***Transfer Function*** block

The top boxes correspond to the numerator coefficients, while the bottom boxes correspond to the denominator coefficients. See the ***Coefficient*** block's example which also applies here.

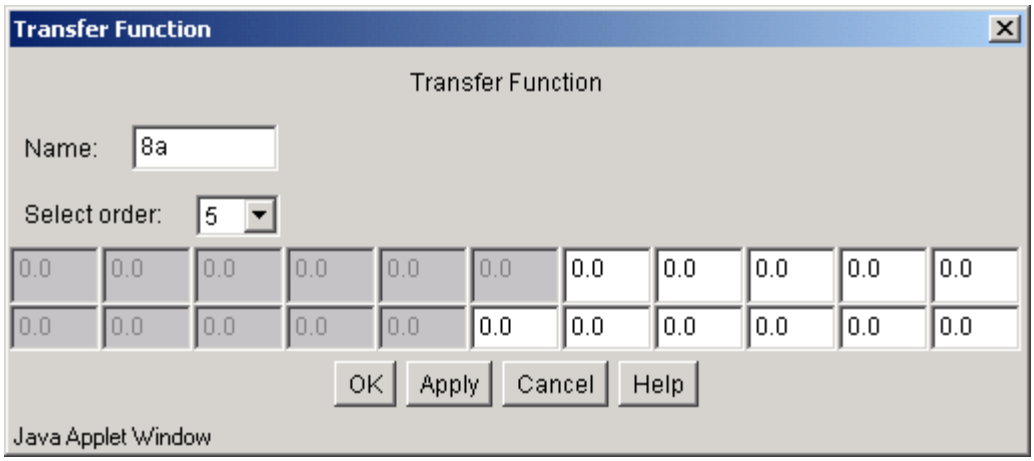

Figure 18: ***Transfer Function*** block's dialog box

Step

Description: The ***Step*** block generates a step sequence for the entire length of the simulation.

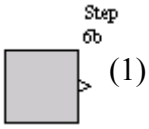

Figure 19: ***Step*** block

Inputs: None

Outputs: (1) Step sequence (time domain)

Algorithm/Equation implemented:

$$y_i = 1 \text{ for all } i$$

where y_i is the output of pin (1)

Dialog Box: By double-clicking on the ***Step*** block a dialog window appears as shown on figure 19. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. [OK] button: Close the window and pass the parameters to the part
3. [Apply] button: Pass the parameters to the part without closing the window
4. [Cancel] button: Close the window ignoring any changes
5. [Help] button: Displays the Help dialog box for the ***Step*** block

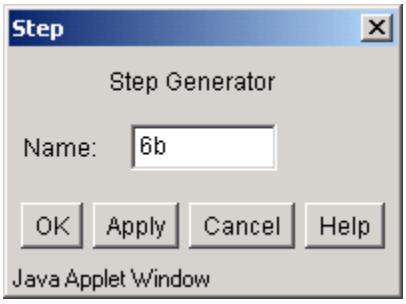

Figure 20: ***Step*** block's dialog box

Bode

Description: The ***Bode*** block generates a bode plot of a system defined by a transfer function loaded from the ***Transfer Function*** block's top output pin

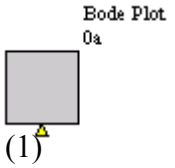

Figure 21: ***Bode*** block

Inputs: (1) Frequency Response (Coefficients)

Outputs: None.

Algorithm/Equation implemented:

N/A

Dialog Box: By double-clicking on the ***Bode*** block a dialog window appears as shown on figure 21. The following is a list of items found on this dialog box:

1. Name: Change the name of the block
2. [OK] button: Close the window
3. [Help] button: Displays the Help dialog box for the ***Bode*** block

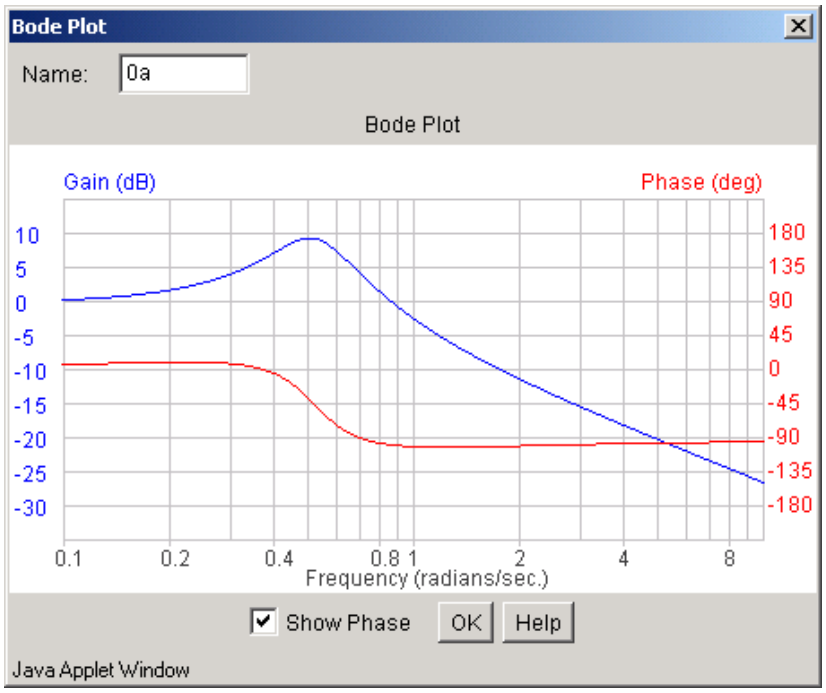

Figure 22: ***Bode*** block's dialog box

J-DSP Figures and Plots

This version of J-DSP allows you to save your work by taking a snapshot of the screen or a dialog window and copying it into a Microsoft Word, PowerPoint or any other program as needed.

In order to do this, you need to use the following buttons or button combinations:

PrntScrn in order to capture the whole screen
Or
Alt- PrntScrn in order to capture only the active window

Here are two examples on how to save your work in an Ms Word file:

Example 1: Saving a plot using *Alt- PrntScrn.*

- Step1: Create a plot in J-DSP by running a simulation. Select the plot's dialog window.

- Step2: Press *Alt- PrntScrn* (*Alt* and *PrntScrn* buttons simultaneously) to copy the dialog window into the clipboard as an image.

- Step3: At the Ms Word file that you wish to save your work in, place the cursor where you need your work to appear and press *Ctrl-V* (or right-click and Paste). The result is shown in figure 1:

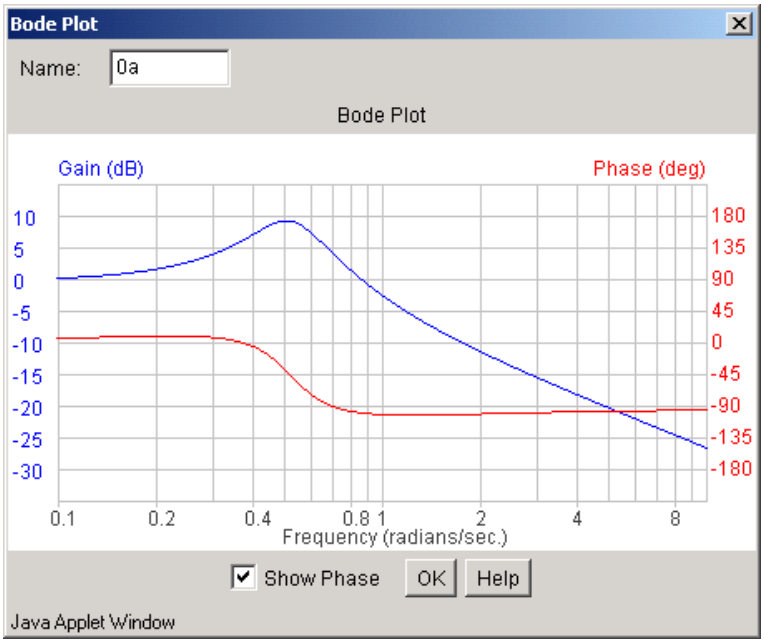

Figure 1: A bode plot saved using example 1

Example 2: Saving a flowgram using *PrntScrn*.

- Step1: Select the J-DSP Editor's window and make sure your flowgram is visible and not covered by other dialog windows.
- Step2: Press PrntScrn to copy the monitor's contents into the clipboard.
- Step3: In windows, open the program Paint which is usually found in the Accessories menu (you can use any other image editing program as you like).
- Step 4: In Paint, select Edit and then Paste (or press *Ctrl-V*). This will transfer the contents of the clipboard to the program as shown in figure 2:

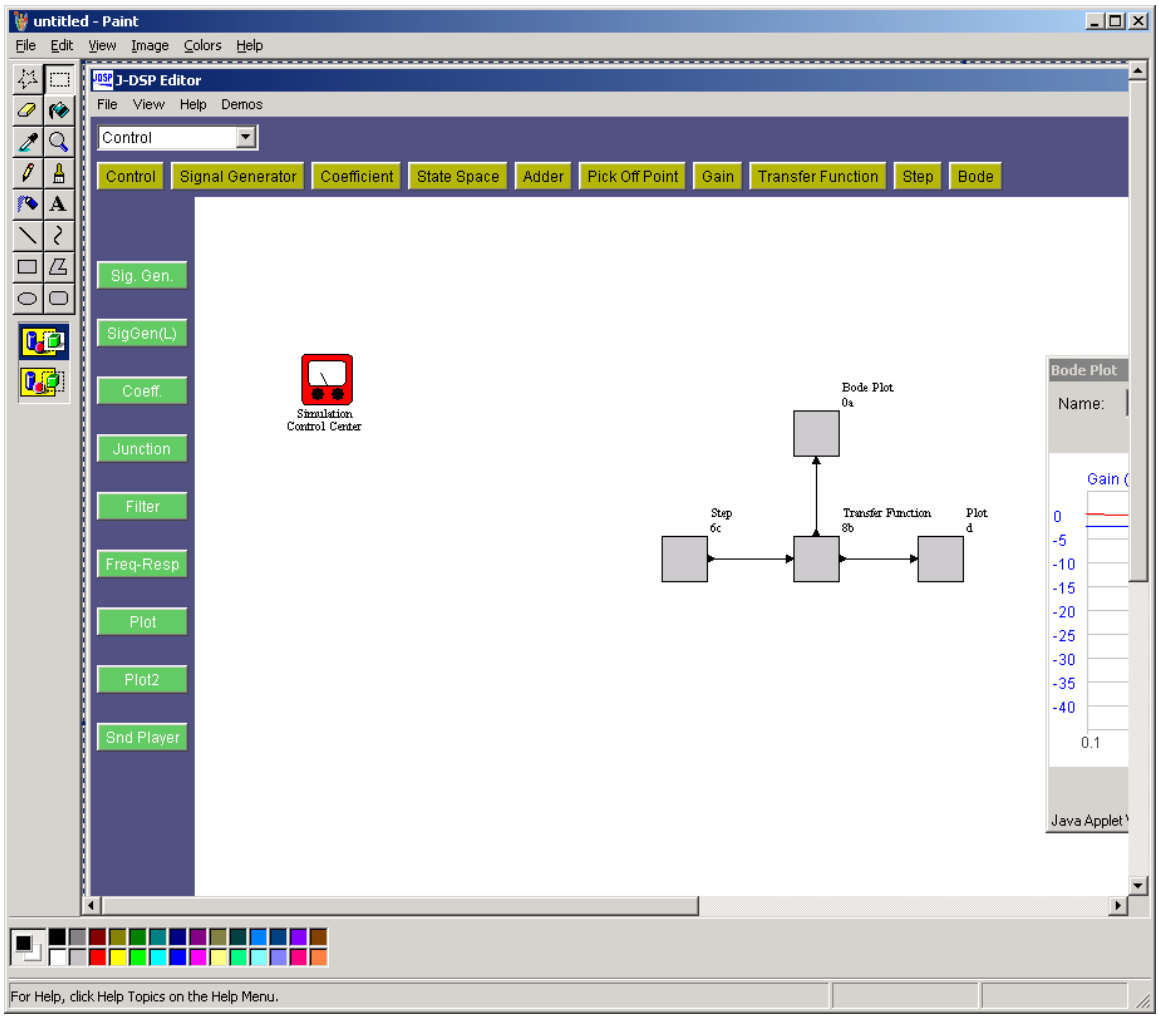

Figure 2: Using Paint to save a flowgram

- Step 5: Use the select tool in Paint to select the desired area over the flowgram and press *Ctrl-C* (or Edit and then Copy).

- Step6: At the Ms Word file that you wish to save your work in, place the cursor where you need your work to appear and press *Ctrl-V* (or right-click and Paste). The result is shown in figure 3:

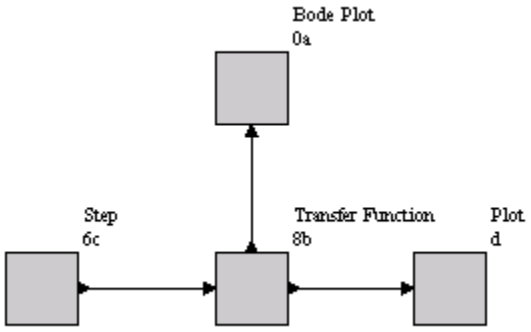

Figure 3: Saved flowgram using Paint

Note: There are many programs that make this task easier, so look on-line for screen capturing software which may help you simplify the above procedure. In the future, J-DSPC will be fitted with the ability to export flowgrams and plots without the need to resort to other programs.

EEE 480 LAB EXPERIMENTS

K. Tsakalis

July 10, 2002

1 LAB1

1.1 Scope

The objective of this Lab is to familiarize the student with the use of the J-DSPC Editor as an introductory platform for analysis and design of control systems. (Also see "Introduction to J-DSPC" and "Control blocks manual" on the web). If instructed, present all your plots in a Word file.

1.2 Assignment

1. **Using help**: Each J-DSPC block has a dialog window which appears by double-clicking on the block. There, you will find a [Help] button which when pressed brings out useful information about the specific part.

2. **Plots**: Plot the function y(t) = sin(t) for 30 time samples (i.e. set the *pulsewidth* to 30). Use the ***Sig. Gen.*** and ***Plot*** blocks.

3. **Working with Figures**: Present the result of Experiment 2 as a figure. Copy and paste the figure in a Word or PowerPoint document.

4. **System Responses**: Use the ***Step***, ***Transfer Function, Bode*** and ***Plot*** blocks to generate the step and frequency responses of the systems with transfer functions:

$$G_1(s) = \frac{1}{5s+1},\ G_2(s) = \frac{1}{s^2 + 0.3s + 1}$$

   Use *pulsewidth* = 4000 and *D*t = 0.15.

5. **Composite Systems**: Compute the step response of cascade (series) connection of the systems defined in Experiment 4. Also compute the step response of the feedback system

$$y(s) = G_2(s)G_1(s)e,\ \ e = r - y$$

6. **State-space representations**: Generate state-space representations of the systems defined in Experiments 4 and 5. Use the State Space ***Coefficient*** clock connected at the bottom input of the ***State Space*** block. Open the Dialog window of the ***State Space*** block and press the [Load Matrix X] button for matrices A, B, C, D to see the respective State Space representation which has been loaded automatically. Note that it will be in canonical form. You can also do this by hand.

2 LAB2

2.1 Scope

The objective of this Lab is to study the time and frequency responses of simple systems and obtain insight on the key response indicators.

2.2 Assignment

1. **Normalization**: In this type of study, it is convenient to normalize the system transfer functions and eliminate the degrees of freedom that have an easily understood effect on the system response. Such normalizations are output scaling and time scaling that correspond to changes in the units of the output and time.

   #1: Show that after normalization, the transfer function of a stable first-order system can be written as:

   $$K\frac{s+b}{s+a} \to \frac{\tau_z+1}{s+1}, \text{ or } \frac{s+z}{s+1}$$

   The first is applicable to systems that have non-zero DC-gain, and the second to systems that have non-zero direct throughput.

   #2: Similarly, stable second-order systems with no direct throughput can be written as

   $$K\frac{s+b}{s^2+a_1s+a_0} \to \frac{\tau_z s+1}{s^2+2\zeta s+1}, \text{ or } \frac{s+z}{s^2+2\zeta s+1}$$

2. **First-order s**ystems have particularly simple responses. Stable and minimum-phase systems can exhibit lead or lag response depending on the sign of the phase of the frequency response. Non-minimum phase systems (with RHP zeros) exhibit inverse response. Study the effect of the transfer function zero ($1/\tau_z$) on the frequency response and step response of the system.

   #3: Plot the step responses, frequency response magnitudes and phases (bode plots) for three representative cases of τ_z that illustrate the different types of response.

   Remarks:
   1. Critical points for τ_z are 0 and 1.
   2. The step response may be discontinuous at 0.
   3. The limit values of the step response at 0^+, ∞ can be computed using the Laplace limit theorems.
   4. To better view the results, experiment with the *D*t value in the ***Control*** block.

3. **Second-order systems** may exhibit overdamped or underdamped response depending on the value of the damping ratio ζ. The overshoot of the step response and the peak-magnitude in the frequency response are amplified by zeros near the system bandwidth. Non-minimum-phase zeros result in inverse response. Study the effect of the transfer

function zero ($1/\tau_z$) and the damping ratio ζ on the frequency response and step response of the system.

While there are many useful ways to view the results, here we focus on bandwidth and overshoot:

#4: Let ζ = 0.5 and τ_z range in the interval [-10, 10] and plot the percent-overshoot and bandwidth as functions of τ_z (use a numerical computation of these quantities). Try *Dt*=0.05

#5: Repeat for ζ = 1.

4. **Third-order systems** may exhibit even more complicated behavior. Here, we are interested in systems that can be written as a cascade of a complex conjugate pole-pair and a first order lead or lag transfer function. Such systems arise often in simple compensator design problems where the dominant closed-loop dynamics are second order with the addition of a pole and a zero near each other and within the bandwidth of the dominant pair.

   In such cases the system response is primarily dictated by the dominant pole-pair with a relatively small perturbation caused. In general, first order lead elements amplify the overshoot of the step response and the peak-magnitude in the frequency response; on the other hand, first order lag elements attenuate both of these measures. The amount of amplification or attenuation depends on the distance between the pole and the zero and their relative location with respect to the dominant pair. Study the effect of the pole and zero of the first-order element and the dominant pair damping ratio on the frequency response and step response of the system.

   Again, there are many useful ways to view the results. One possibility is to fix ζ and τ_z and plot the overshoot (or any other measure of interest) as a function of the "distance" between the pole and the zero.

   #6: For ζ = 0:5 generate a plot showing the overshoot as a function of τ_p / τ_z, for different values of τ_z. Select a few values of τ_z in the interval [0:2; 10] and $\tau_p = \tau_z$ in the interval [0:7; 1:4].

   #7: Based on this study, determine the pole locations p such that the transfer function

$$\frac{s+0.3}{(s+p)(s^2+0.894s+0.8)}$$

   exhibits less than 20% overshoot. Verify with a simulation.

Lab Specific Evaluation Forms

Please select 1-5, 5 being the best possible mark.

Lab1

a. The contents of this exercise helped you understand the concepts of step and frequency response of a system.
1 – 2 – 3 – 4 – 5

b. Performing this exercises helped you grasp the relationship between a transfer function and a state space representation. (Answer only if lab 1, part 6 was assigned)
1 – 2 – 3 – 4 – 5

Lab2

a. You now understand better the effect of τ_z on the step and frequency response of a system.
1 – 2 – 3 – 4 – 5

b. The effect of the damping ratio ζ is now clearer
1 – 2 – 3 – 4 – 5

c. The simulations allowed you to better comprehend the overall parameters affecting the overshoot.
1 – 2 – 3 – 4 – 5

General J-DSPC Evaluation Forms:

a. How would you rate the J-DSPC concept?
1 – 2 – 3 – 4 – 5

b. Establishing and connecting blocks is easy
1 – 2 – 3 – 4 – 5

c. Do you like the idea of an Internet based simulation tool such as J-DSPC?
1 – 2 – 3 – 4 – 5

d. Did you experience any problems while starting the J-DSPC editor?
Yes – No

e. Changing the options in the blocks is easy and convenient.
1 – 2 – 3 – 4 – 5

f. The graphical interface of J-DSPC is intuitive and user-friendly
1 – 2 – 3 – 4 – 5

g. Should the J-DSPC editor be established as a full-fledged tool?
1 – 2 – 3 – 4 – 5

h. What would you change in the J-DSPC program to make it more useful and user-friendly?

i. Please describe any errors or bugs you encountered.

j. Do you have any general comments and suggestions?