

# J-DSP-C, A CONTROL SYSTEMS SIMULATION ENVIRONMENT: LABS AND ASSESSMENT

*Thrassos Thrasyvoulou, Kostas Tsakalis and Andreas Spanias<sup>1</sup>*

**Abstract** – This paper presents various on-line computer laboratory exercises and related assessment results for Java-Digital Signal-Processing-Controls (J-DSP-C). J-DSP-C is an object oriented simulation environment that is intended for control systems simulations. All J-DSP-C component functions appear as graphical blocks that can be connected to form a variety of educational controls simulations. Dynamical systems with various interconnection topologies can be developed and simulated while results can be examined graphically. J-DSP-C has been designed to be easily accessible through the internet with any computer system equipped with a Java-enabled browser.

**Index Terms** – Control systems simulation, education, internet technology, assessment.

## INTRODUCTION

J-DSP-C [1] is a Java-based object-oriented control systems programming environment recently developed at Arizona State University (ASU) for use in undergraduate education. This environment enables the simulation of dynamical control systems from any computer system equipped with an internet browser. It is an improvement on J-DSP [2] in that it has been developed to allow feedback, multiple inputs and outputs, miscellaneous block shapes and most importantly, improved data processing. The software is freely distributed and is designed to enhance distance learning and continuing education with an on-line laboratory experience. For this reason, J-DSP-C is accompanied by servlets that enable electronic report submission and automatic grading. This technology enables the students to submit appropriate graphs and data as well as student remarks, while the instructor can examine the student report in HTML format. J-DSP-C can also be used to add interactive examples into web content for education and demonstration purposes.

A description of J-DSP-C and its capabilities has been provided in [1]. This paper is a significant extension in that it considers evaluation results and comments collected when J-DSP-C was presented for actual laboratory use and evaluation in the undergraduate feedback control systems class at ASU (EEE 480). During this phase, two exercises were assigned using J-DSP-C and the objective was primarily the assessment of the student perception of the software. The two exercises are based on material that the

students typically learn in the first few weeks of the course. Such labs are considered to be fairly important because they help students visualize the application of abstract theoretical concepts. Subsequent labs (currently under development) will address the design of compensators and the evaluation of the resulting feedback system.

The first exercise introduces the students to the basic functionality of J-DSP-C through a series of simple tasks. These include examining step and frequency responses of control systems described by rational transfer functions or state-space representations. Composite systems with feedback are presented and analyzed through the use of basic blocks such as the step signal generator, transfer function and bode plot. The second exercise further elaborates on time and frequency responses. The effects of the damping ratio and transfer function zero of first and second order systems are examined. Overdamped and underdamped responses are examined in detail in relation to the locations of poles and zeros in the system transfer function.

The remaining part of the paper is organized as follows: First, information regarding J-DSP-C's graphical interface is provided followed by a presentation of the set of functions available for simulations. Next, the exercises and their respective evaluation results are presented. We conclude with a summary of our observations.

## THE J-DSP-C ENVIRONMENT

J-DSP-C can be used to simulate dynamical systems with a variety of interconnection topologies, including feedback. In order to accommodate simulation of such systems, the J-DSP-C GUI has been designed to handle blocks with multiple input/output ports. This is in contrast with the standard J-DSP, originally developed solely for sequential simulations encountered in digital signal processing. The J-DSP-C blocks are displayed in shapes that are common in control book notation and are relative to their function. For example, an adder is usually represented with a circle and a gain block with a triangle. Blocks can be rotated, flipped and their interconnections can be edited or modified. Such capabilities are important in achieving a good visual appearance effect and become more essential when the complexity of the system increases. An example of a block diagram with multiple inputs and outputs designed using the new J-DSP-C graphical user interface (GUI) is illustrated in Figure 1. Figure 2 provides a closer look at the capabilities

<sup>1</sup> All authors are with the MIDL, Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-7206. thrassos@asu.edu, tsakalis@asu.edu, spanias@asu.edu

of J-DSP-C in handling complex control system diagrams. All J-DSP-C blocks are associated with dialog windows that appear when the user double-clicks on a block. These dialog windows allow the user to enter or modify the block parameters, e.g., the signal type or the transfer function coefficients or state space description of the system.

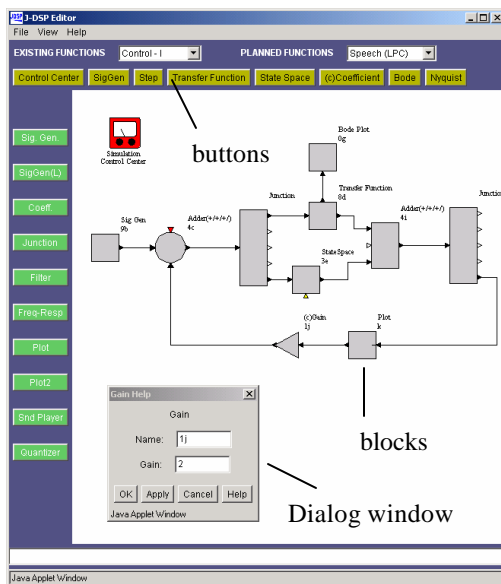


FIGURE 1  
MULTIPLE INPUT/OUTPUT BLOCKS AND J-DSP-C USER INTERFACE .

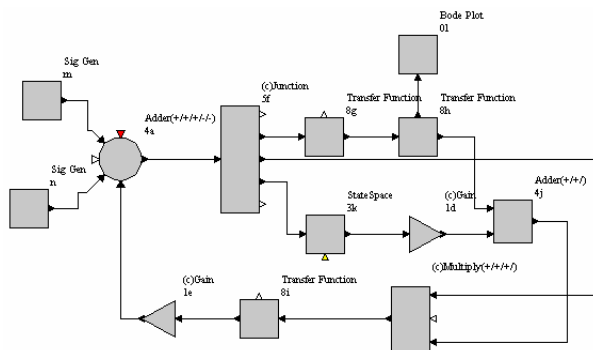


FIGURE 2  
J-DSP-C CAPABILITIES FOR CONTROL SYSTEM DIAGRAMS.

### J-DSP-C FUNCTIONALITY

The entire collection of blocks available in J-DSP-C constitutes a fundamental set of functions that allow the simulation of a variety of systems required for educational purposes. Simulations can be performed by appropriately connecting the necessary J-DSP-C blocks.

J-DSP-C includes two signal generators able to provide a variety of signals. The main signal generator block can provide discrete impulses, sinusoids, sinc functions, random signals with either uniform, Gaussian, or Rayleigh

distributions and exponential signals. If applicable, signals can be made to repeat periodically.

J-DSP-C also includes functions for linear dynamical systems that can be specified either in terms of their transfer function or a state-space representation. More precisely, the transfer function block is used to define a system with transfer function coefficients  $b_i$  and  $a_i$  in the form

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

An alternative and more general way to specify a linear system is with a state-space description. This block implements the equation

$$\begin{aligned} \dot{\underline{x}} &= A\underline{x} + B\underline{u} \\ \underline{y} &= C\underline{x} + D\underline{u} \end{aligned}$$

where  $A$  is an  $n \times n$  matrix,  $B$  is  $n \times m$ ,  $C$  is  $p \times n$ , and  $D$  is  $p \times m$ .  $\underline{x}$  is the state vector of length  $n$ ,  $\underline{u}$  is the system input represented by a vector of length  $m$  and  $\underline{y}$  is the system's output response given by a vector of length  $p$ . The maximum number of states  $n$  is limited to 10. Presently only one input and output are allowed ( $m=1, p=1$ ). However, with small modifications, the software will be able to handle up to 10 inputs and outputs. Figure 3 presents the dialog window of the state-space block that can be used to enter matrices to describe a dynamical system.

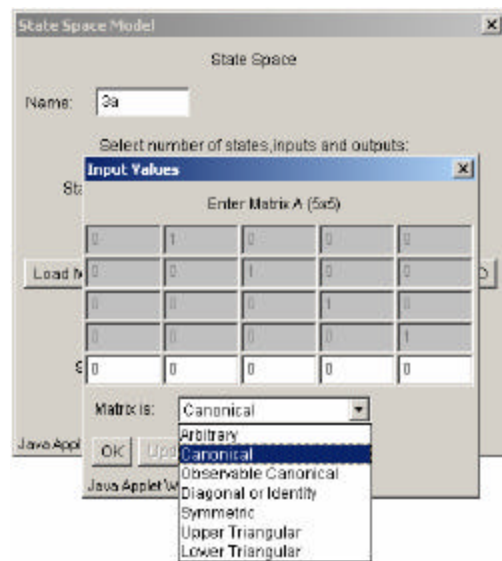


FIGURE 3  
STATE-SPACE DIALOG WINDOW.

Other blocks are designed for mathematical manipulation of signals. This block category includes summation nodes, gains, logarithmic and exponential blocks and multiplication.

Simulation results can be examined using the graphical output capabilities of the plot block which can plot its input in a linear or logarithmic scale. The plot block offers zooming capabilities while at the same time it can provide statistical properties of the displayed signal. Its appearance and dialog window are shown in Figure 4. The Bode and Nyquist plot blocks can also be used to visualize system properties. The Bode plot displays the magnitude and phase of the system transfer function while the Nyquist block plots imaginary versus real parts. The difference between the generic time response block and the Bode/Nyquist blocks is that the latter displays the frequency response immediately upon connecting or modifying the system; the time response is displayed only after the simulation is executed.

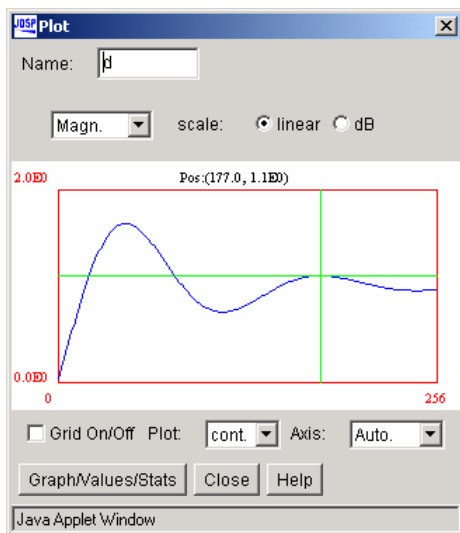


FIGURE 4  
PLOT BLOCK DIALOG WINDOW.

## EVALUATION AND ASSESSMENT

A total of 27 students taking the senior level introductory control class at ASU took part in the evaluation. They were assigned two sets of exercises to be performed using J-DSP-C and were asked to complete them as a part of a homework laboratory assignment. They could do so using their own personal computers or computers made available at the University. Most students who worked at home connected to the internet using a 56.6Kbps dial up modem connection. Others reported using a high bandwidth connection either from their house or the University network. All but one student had no prior experience with either J-DSP or J-DSP-C.

Part of the assigned homework required submitting a report, and evaluating the software. The evaluation questions

were divided into three parts, the first two being exercise-specific. The third part of the questionnaire included a set of general questions. This was composed of three qualitative and seven multiple-choice questions. In answering the multiple-choice questions, the students provided a grade from one (worst) to five (best).

### First exercise set

The first exercise assigned aimed at helping the students visualize the basic ideas behind step and frequency responses. Students were also asked to examine simple composite systems in order to comprehend their behavior. At the same time this exercise was also designed as a tool that introduces students to the use of the J-DSP-C editor as a platform for analysis and design of control systems.

The first task involved the plots of step and frequency responses of simple first and second order systems given in the form of a transfer function. The students had to connect the transfer function block with the Bode block. For the step response, the students had to connect a plot block to the transfer function and introduce a signal generator block to provide the input excitation. The simulation parameters had to be set by the student through each block's dialog windows.

In the next part of the exercise, the students were asked to place the systems used in the first simulation in cascade and then connect them with feedback. No detailed instructions were given other than a general description of the problem. Figure 5 presents a result that students would observe for two particular systems simply by opening the plot block's dialog window. Figure 6 shows the requested setup.

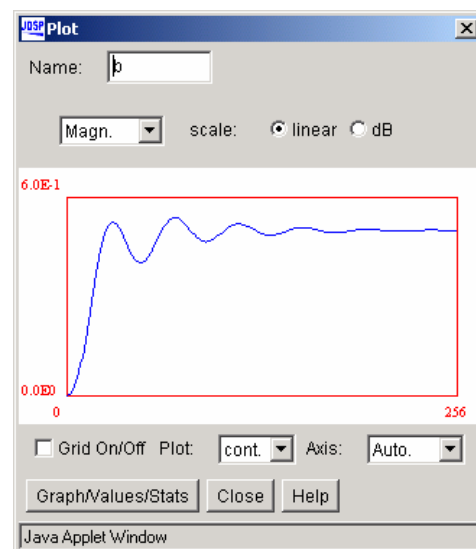


FIGURE 5  
RESULT OF EXERCISE 1.

When students were asked whether the contents of the exercise helped them understand the concepts of step and frequency response of a system, their response was as follows: 3.7% gave a mark of 1, 11.1% gave a mark of 3, 55.65% gave a mark of 4 and 29.6% gave a mark of 5. These results are deemed encouraging. About 85% of the students gave J-DSP-C a mark of 4 or above.

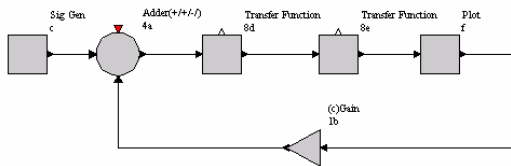


FIGURE 6  
FEEDBACK SYSTEM FOR EXERCISE 1.

**Second exercise set**

The objective of the second lab was to study the time and frequency responses of simple first and second order systems and obtain insight on the key response indicators. For the simple first order case, students would be able to examine stable, minimum-phase systems and observe that such systems can exhibit lead or lag response depending on the sign of the phase of the frequency response. An inverse response is exhibited by non-minimum phase systems. For second order systems, students were also expected to observe that the overshoot of the step response and the peak-magnitude in the frequency response are amplified by zeros near the system bandwidth.

For the first part of this exercise, students were asked to normalize the system transfer functions presented to them as shown below:

$$K \frac{s+b}{s+a} \rightarrow \frac{t_z+1}{s+1} \text{ or } \frac{s+z}{s+1}$$

and

$$K \frac{s+b}{s^2+a_1s+a_0} \rightarrow \frac{t_zs+1}{s^2+2zs+1}$$

For the second part of the exercise, students were requested to use J-DSP-C to study the effect of the transfer function zero  $1/t_z$  on the frequency and step response of various systems. They were asked to plot the step responses, frequency response magnitudes and phases (Bode plots) for three representative cases of  $t_z$  that would illustrate the different types of response for a first order case. Moreover, second-order systems exhibit an overdamped or underdamped response depending on the value of the damping ratio  $z$ . Therefore students were asked to use J-DSP-C in an attempt to determine the effect of the damping

ratio on the system response. While there are many useful ways to view the results, students were asked to focus on bandwidth and overshoot. Figure 7 presents a Bode plot that students would use to visualize the results.

After concluding both exercises, students were required to comment on the truthfulness of several statements. The questions/statements posed to them during the assessment part are listed below:

- Q1: You now understand better the effect of  $t_z$  on the step and frequency response of a system (1: worst – 5: best).
- Q2: The effect of the damping ratio  $z$  is now clearer (1: worst – 5: best).
- Q3: The simulations allowed you to better comprehend the overall parameters affecting the overshoot (1: worst – 5: best).

The student responses are shown in Figures 8 to 10, one for each question above. Students in general enjoyed using J-DSP-C and felt they learned something more by using it.

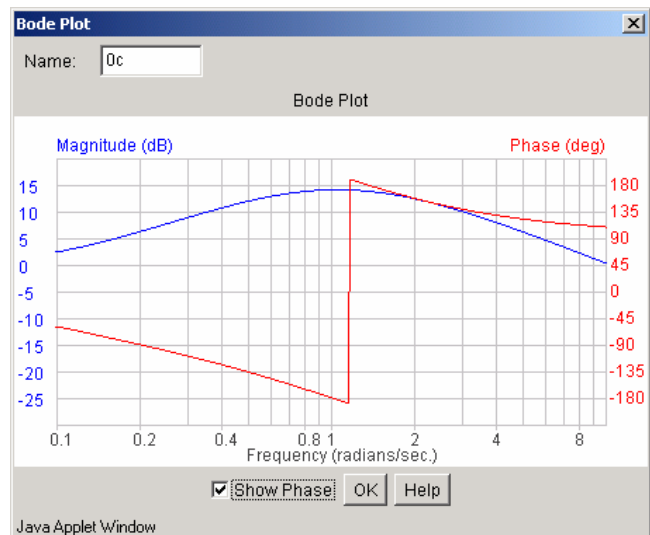


FIGURE 7  
BODE PLOT OF AN UNDERDAMPED SYSTEM

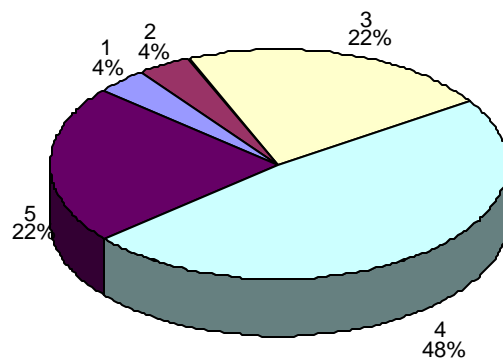


FIGURE 8  
STUDENT ANSWERS TO QUESTION Q1.

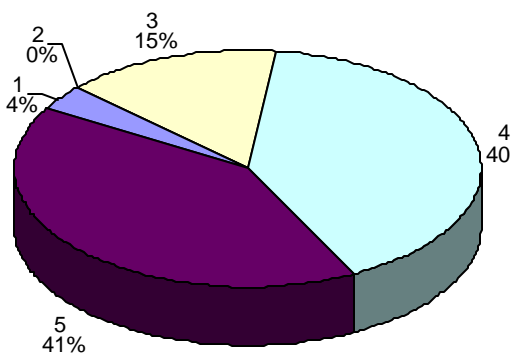


FIGURE 9  
STUDENT ANSWERS TO QUESTION Q2.

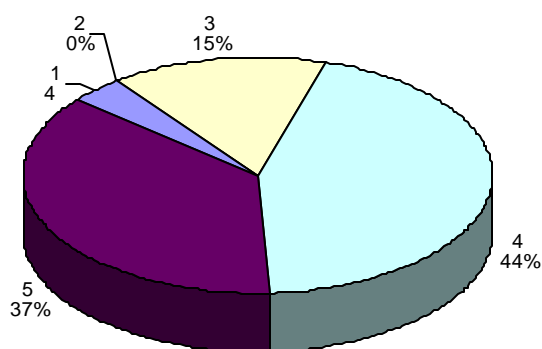


FIGURE 10  
STUDENT ANSWERS TO QUESTION Q3.

### General Evaluation

The first part of the general questionnaire asked students what they would change in J-DSP-C to make it more user-friendly. Most students suggested extending the documentation in the help dialog window of each block. Some asked that the help dialog windows be further enriched with pictures and examples. Apparently, most of the students looked in the dialog windows for help, instead of considering the introductory material and J-DSP-C manuals available to them. Some of the students explicitly mentioned in their comments that all the problems they faced were easily solved as soon as they skimmed through the introductory J-DSP-C material. In future versions of J-DSP-C, an attempt will be made to satisfy this student request for better help screens, even though the dialog windows were originally meant to be a short and concise description of the block and only a complement to the manual. However, J-DSP-C has proven how user friendly and intuitive it is, because students were able to complete their reports without any significant problems and in a short period of time. Other student answers reported that J-DSP-C was already a useful tool and did not provide requests for any changes. Others requested modifications in relation to the ability to save systems on their hard disk, and to be able to print the block diagram or a plot directly from J-DSP-C. Unfortunately

these problems arise from the fact that J-DSP-C runs as a Java applet on the user's computer and does not have access to essential computer resources that allow such functionality. Solutions to this problem are currently being considered.

While reporting any errors or bugs they encountered, most of the students cited minor delay problems, especially when running J-DSP-C using a dial-up connection. Students running J-DSP-C from computers at the ASU computer labs did not face such problems. Very few students had difficulty running their first simulation because they did not know they should use the control block to set general simulation parameters. This block controls the behavior of J-DSP-C and is necessary to run a simulation. Again, this problem was traced to the general practice of ignoring the manuals or the introduction to J-DSP-C where this information was included. Once students actually read the manual they did not report any major problems. (Nevertheless, we plan to take this issue under consideration in the future versions of J-DSP-C.) A few other students provided information regarding minor browser compatibility issues.

When asked for general comments and suggestions, the majority of the students said they were satisfied with J-DSP-C and that they found it as a nice and convenient tool. Some compared it favorably with major packages in common use today, especially when it is to be used for small tasks in the way it is designed. Students enjoyed the simplicity of the software and the fact that it is entirely free and that they were able to run it from any computer without having to worry about installation, copyrights and software licenses. They also repeated their request for more on-line documentation to be included with the help dialog windows. Most students found that the block diagrams were very intuitive. In general, J-DSP-C was deemed a good tool for use in controls classes.

The last part of the general evaluation involved seven questions/statements. The students were asked to grade J-DSP-C with a mark from one to five. The questions/statements were:

Q1: How would you rate the J-DSP-C concept (1: worst – 5: best)?

Q2: Establishing and connecting blocks is easy (1: worst – 5: best).

Q3: Do you like the idea of an Internet based simulation tool such as J-DSP-C (1: worst – 5: best)?

Q4: Did you experience any problems while starting the J-DSP-C editor (1: Yes, 5:No)?

Q5: Changing the options in the blocks is easy and convenient (1: worst – 5: best).

Q6: The graphical interface of J-DSP-C is intuitive and user-friendly (1: worst – 5: best).

Q7: Should the J-DSP-C editor be established as a full-fledged tool (1: worst – 5: best)?

The student responses were very encouraging. In the first question almost 90% of the marks obtained were either 4 or 5. The second question obtained 80% fours and fives. When asked whether the students enjoyed the basic idea behind J-

DSP-C the majority answered positively. A complete list of student answers is shown in Table I.

TABLE I  
GENERAL EVALUATION ANSWERS.

	Evaluation mark (%)				
	1	2	3	4	5
Q1	3.7	7.4	0.0	40.7	48.1
Q2	3.7	3.7	11.1	18.5	63.0
Q3	3.7	7.4	11.1	22.2	55.6
Q4	40.7	N/A	N/A	N/A	59.3
Q5	3.7	3.7	14.8	55.6	22.2
Q6	0.0	14.8	25.9	33.3	25.9
Q7	3.7	11.1	14.8	37.0	33.3

### CONCLUSIONS

This paper presented J-DSP-C and recently conducted evaluation results obtained from a sample of undergraduate control students. Concept specific assessment revealed that students gained knowledge on the specific subjects examined and were left with a good visual impression of the material. The results of the general evaluation indicate that students had a positive view of J-DSP-C and that they would be willing to use if for educational tasks. The lack of some features found on major commercial packages was noted, but did not present a major problem. Instead, many students seemed to enjoy the simplicity and visualization introduced in the simulations by J-DSP-C.

### ACKNOWLEDGMENT

The JDSP-C and the original JDSP [1]-[4] projects are funded in part by the NSF CCLI program grant DUE 0089075.

### REFERENCES

- [1] Thrasyvoulou, T., Tsakalis K., and Spanias A., "J-DSP-Control: A control systems simulation environment", *Proc. 22nd IASTED Intl. Conf. on Modeling, Identification and Control*, Innsbruck, Austria, Feb. 10-13, 2003.
- [2] Spanias, A. et al., "Development and Evaluation of a Web-Based Signal and Speech Processing Laboratory for Distance Learning", *ASEE Computers in Education Journal*, Vol. X, No. 2., April-June 2000, pp. 21-26.
- [3] Spanias A. et al, "On-line laboratories for speech and image processing and for communication systems using J-DSP", *2nd DSP-Education workshop*, Pine Mountain, GA, Oct. 13-16, 2002.
- [4] J-DSP on-line help, <http://jdsp.asu.edu>