AN INTERACTIVE HYBRID PROGRAMMING APPROACH TO SIGNALS AND SYSTEMS LABORATORY

N. Kehtarnavaz, P. Loizou, and M. Rahman

Department of Electrical Engineering, University of Texas at Dallas

ABSTRACT

Signals and systems lab courses that are currently offered at many universities are mostly based on text-based programming languages and environments, in particular MATLAB. This paper presents an alternative programming approach in these courses by combining textual with graphical programming. This hybrid programming approach offers code interactivity in a time-efficient manner. Example labs are presented to demonstrate how this approach allows students to interact and experiment with their codes to gain a better understanding of the signal processing concepts.

Index Terms **–** Education, education technology, signals and systems laboratory, hybrid programming, interactive approach to lab courses.

1. INTRODUCTION

Signals and systems is normally the very first course in the electrical engineering undergraduate curriculum where students get exposed to the signal processing concepts such as convolution, Fourier series, Fourier transform and filtering. When teaching signals and systems lab courses, we have made the observation that students spend a fair amount of time establishing interactivity with their text-based codes. Among different text-based codes, MATLAB is the most widely utilized coding in signals and systems lab courses. In many cases during lab sessions, students are left with little time to gain a better grasp of the signal processing concepts. Although code writing is an important aspect of these lab courses, students' understanding of the signal processing concepts can be enhanced if they get more involved in system building and analysis in an interactive manner that is time-efficient.

In this paper, we discuss an alternative approach to teaching signals and systems lab courses based on hybrid programming. Hybrid programming constitutes a combination of textual and graphical programming which has been previously used for educational purposes, for example [1]. Textual and graphical programming each has its own merits and demerits from a programming point of view. In general, math operations are easier to code in textual mode. For instance, MATLAB provides a rich set of built-in functions for performing signal processing math operations involving vectors and matrices. On the other hand, graphical programming offers an easy-to-build interactive and visualization environment towards building signal processing systems.

In an effort to bring together the preferred features of textual and graphical programming, we have redesigned the labs in a typical signals and systems laboratory course by incorporating MATLAB codes into the LabVIEW graphical programming environment. This way although the coding is done in MATLAB, the interactivity and visualization is achieved in LabVIEW by just having some basic knowledge of LabVIEW. Such a basic knowledge can be easily gained via an introductory lab to LabVIEW in addition to an introductory lab to MATLAB at the beginning of the course. It should be noted that it is not required to have a detailed extensive knowledge of LabVIEW to build an interactive system around a MATLAB code. This hybrid programming approach allows students to build an interactive system in a time-efficient manner. As a result, they can gain a better understanding of the signal processing concepts by examining various parameters and asking "what if?" questions interactively.

In addition to the signal processing concepts, an application is covered in each lab for students to relate the concepts learned to an actual real-world application. The applications covered span a wide variety of signal processing application areas including speech processing, biomedical signal processing, telecommunications and digital music synthesis. Though these applications are rather simple, they provide motivation for students to stay engaged in the labs.

Section 2 gives a brief overview of the hybrid programming environment. Section 3 provides an outline of the redesigned signals and systems labs, while section 4 includes two sample design applications, with the conclusions stated in section 5.

2. HYBRID PROGRAMMING: LABVIEW AND MATLAB

The LabVIEW environment consists of two major components: Front Panel (FP) and Block Diagram (BD). An FP provides the graphical-user-interface (GUI) while a BD contains the building blocks of a system and resembles a flowchart. LabVIEW systems are called Virtual Instruments (VIs) and their FPs resemble an instrument panel consisting of various controls and displays. For details on the LabVIEW programming environment, the interested reader may refer to [2]. Unlike LabVIEW, MATLAB offers a text-based programming environment with a rich set of built-in signal processing functions but with a relatively involved GUI capability.

A hybrid code can be put together by using so called MathScript nodes or MATLAB Script nodes within LabVIEW [3]. A BD in LabVIEW is capable of executing MathScript or MATLAB Script nodes that contain MATLAB codes inside them. Inputs and outputs can be added to such nodes which appear as controls and indicators and are easily accessed from the FP.

The concept of adding GUI features to textual coding is not new. For example, Vicente et al. [1] designed EASYSP with MATLAB/Octave functions using XML plugins to obtain GUI support integrated systems. However, in terms of building a system or modifying it, the ease with which one can switch between LabVIEW FP and BD is a clear advantage over previous hybrid approaches.

At this point, it is worth mentioning that it is possible to perform hybrid programming through other means, for example, by using MATLAB within the SIMULINK environment [4]. In our approach, we have chosen LabVIEW over SIMULINK since as per the study done in [5], students preferred the interactivity and visualization capabilities of LabVIEW over SIMULINK.

3. SIGNALS AND SYSTEMS LABS

This section briefly describes the signal and systems labs that are redesigned in a hybrid mode where MATLAB codes are embedded in the LabVIEW graphical programming environment [6].

I. **Introduction to MATLAB**: This lab is included for students to gain some familiarity with MATLAB if not previously been exposed to it. Often students are already familiar with MATLAB when taking signals and systems courses.

II. **Introduction to LabVIEW**: This lab is included for students to gain some basic understanding of LabVIEW as how to use controls, indicators and other LabVIEW features to interact with MATLAB codes.

III. **Convolution and linear time invariant (LTI) systems**: This lab is redesigned to allow students to interact with their convolution codes and examine LTI systems. Due to the discrete-time nature of programming, it is required to have an approximation of the convolution integral. The lab shows how to perform numerical approximation of convolution. Furthermore, the convolution properties are examined. Two applications of convolution consisting of an RLC circuit analysis and an echo cancellation system are then discussed.

IV. **Fourier series (FS) and its applications**: In this lab, the representation of periodic analog signals using Fourier series (FS) is examined. The decomposition and reconstruction of periodic signals using a finite number of Fourier coefficients are studied. As an application, an RLC circuit analysis is performed using periodic input signals.

V. **Continuous-time Fourier transform (CTFT) and its applications**: In this lab, the transform CTFT is introduced and its properties are examined. Amplitude modulation and high frequency noise removal are covered as applications of CTFT.

VI. **Digital signals and their transforms:** This lab deals with the transforms associated with digital signals. Analog to digital conversion and related issues including sampling, aliasing are examined during the first part of this lab. In the second part, the transformations consisting of discrete Fourier transform (DFT) and discrete-time Fourier transform (DTFT) are covered and compared to the frequency transformations for continuous-time signals, namely FS and CTFT, respectively. Dual tone multi frequency (DTMF) signaling for touch-tone telephones and dithering to decrease signal distortion due to digitization are covered as applications.

VII. **Analysis of analog and digital systems:** This final lab brings together the techniques and mathematical transform tools learned in the previous labs towards performing analog and digital filtering. Both an analog and a digital filtering system are built and analyzed in an interactive way.

4. EXAMPLE SYSTEMS BUILT IN HYBRID MODE

This section provides an overview of two example systems that are redesigned using the above interactive, hybrid approach.

Figure 1 shows the FP and the BD of an echo cancellation system which provides an application of the third lab. The echo is produced when the signal (speech, in our case) is reflected off a non-absorbing surface like a wall. After implementing this system, what students would hear is the original signal superimposed onto the signal reflected off the wall (echo). Since the speech is partially absorbed by the wall, it will be decreased in amplitude and also get delayed. In other words, a speech signal $x(t)$ gets added to its delayed (Δt) and reflected (a) version which can be written as

$$y(t) = x(t) + ax(t - \Delta t) \qquad (1)$$

Using convolution, one can write Equation (1) as

$$y(t) = x(t) * \{\delta(t) + a\delta(t - \Delta t)\} \qquad (2)$$

Hence, students get to see that the echo added signal is the convolution of the original signal with two delta functions. Next, they see how to recover the original signal from the echo added signal via performing deconvolution. The autocorrelation of the output signal (echo free signal) is used to estimate the echo delay and the amplitude of the echo. For a perfectly echo free signal, they get to see in an interactive way that the side lobes of the autocorrelation should be kept small.

Figure 2 shows the FP and the BD of a linear RC circuit analyzed by using trigonometric Fourier series as an application of the fourth lab. The ability to decompose any periodic signal into a number of sine waves makes FS a powerful tool in electrical circuit analysis. The response of a circuit component when a sinusoidal input is applied to its terminals is well known in circuit analysis. Thus, in order to obtain the response to any periodic signal, one can decompose the signal into sine waves and then perform a linear superposition of the sine waves. The Fourier series coefficients of the input voltage signal are computed and phasor analysis is deployed to obtain the output voltage v_{c_n} of the circuit, where n represent the number of terms in the Fourier series. By using the voltage divider rule, the output voltage v_{c_n} can be written as

$$v_{c\_n} = \frac{1/jn\omega}{1 + 1/jn\omega} v_{in\_n} \qquad (3)$$

where ω represents the angular frequency and v_{in_n} the input voltage for a particular frequency. Then, since the sine and cosine components of the input voltage are known, the output is easily determined by adding the individual output components noting that the circuit is linear. Each output voltage component is determined by using Equation (3). The magnitude and phase of the sine and cosine components are displayed in the FP separately. Furthermore, a tab control is used to show the Fourier series and circuit output separately.

CONCLUSION

In this paper, an alternative approach to teaching signals and systems lab courses has been presented. By combining the graphical-user-interface and visualization features of LabVIEW with textual MATLAB coding, the codes are easily made interactive allowing students to experiment with their codes by varying parameters and seeing their effects. The interactive, hybrid approach presented in this paper is general purpose in the sense that the same approach can be applied to other similar lab courses in the electrical engineering curriculum.

REFERENCES

[1] J. Vicente, B. Garcia, A. Méndez, I. Ruiz and O. Lage, "Easysp: the easiest form to learn signal processing interactively," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

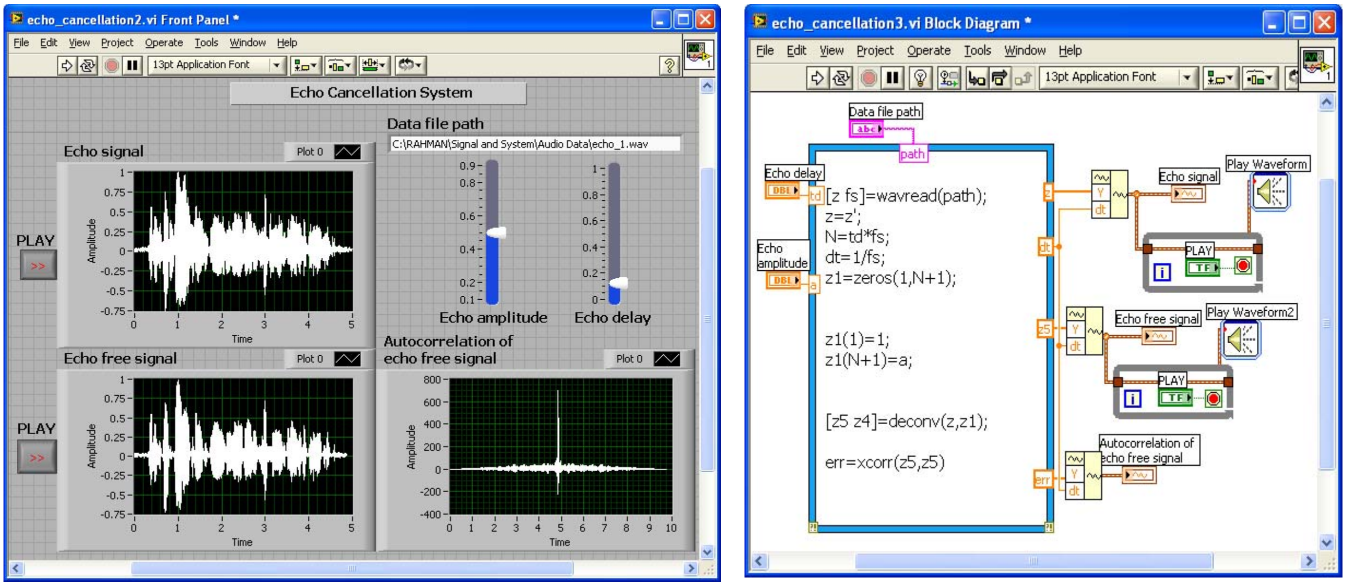
[2] N. Kehtarnavaz and N. Kim, *Digital Signal Processing System-Level Design Using LabVIEW*, Elsevier, 2006.

[3] National Instruments, *Developing Algorithms Using LabVIEW Mathscript*, http://zone.ni.com/devzone/cda/tut/p/id/3256, 2007.

[4] J. Dabney and T. Harman, *Mastering Simulink*, Prentice-Hall, 2003.

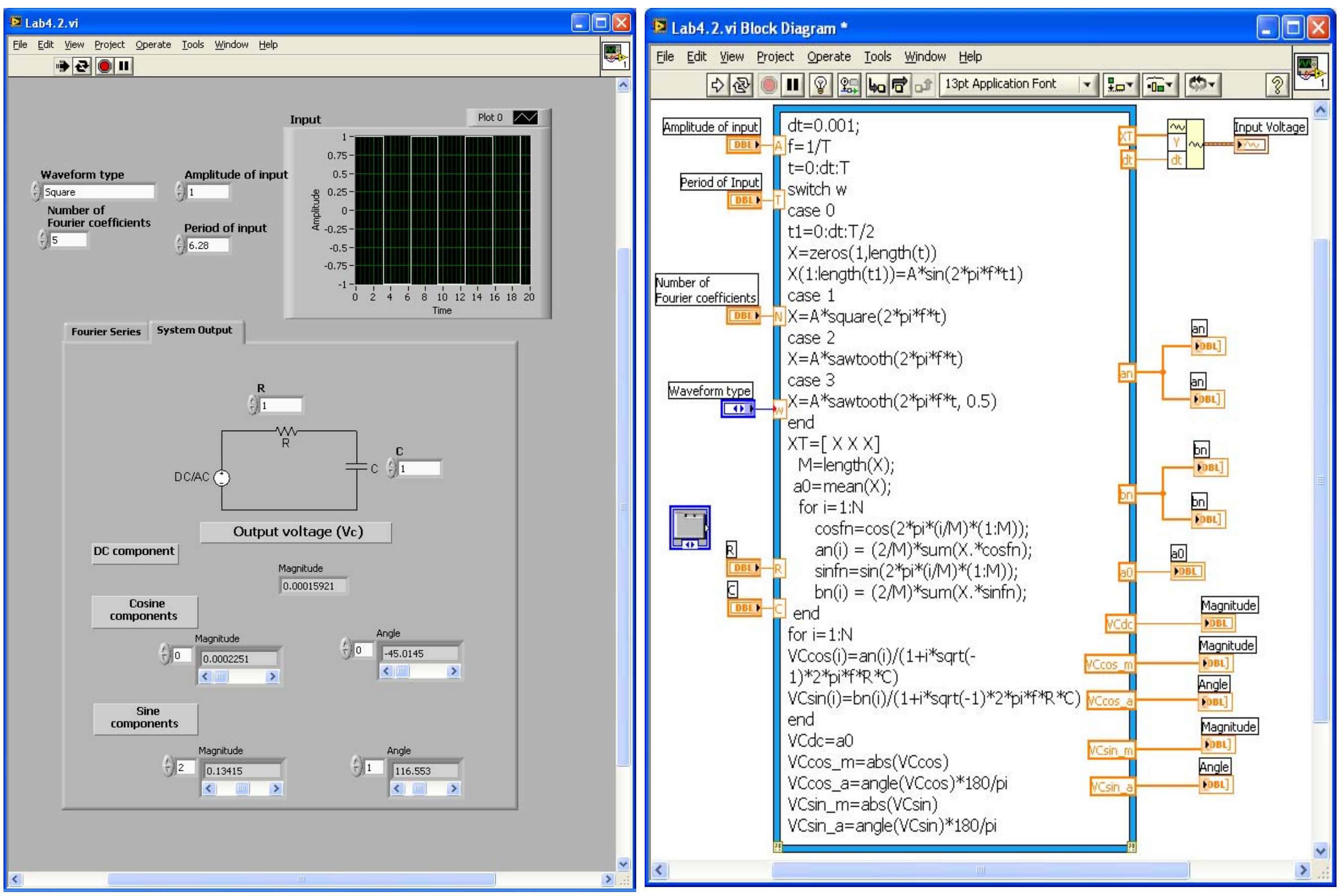
[5] N. Kehtarnavaz and C. Gope, "DSP system design using LabVIEW and Simulink: a comparative evaluation," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[6] N. Kehtarnavaz, P. Loizou, and M. Rahman, *An Interactive Approach to Signals and Systems Laboratory*, National Instruments, to appear in 2008.



(a) (b)

Fig.1 - Echo cancellation system: (a) Front Panel, (b) Block Diagram.



(a) (b)

Fig.2 - Linear circuit analysis using trigonometric Fourier series: (a) Front Panel, (b) Block Diagram.