EXERCISES WITH A VIRTUAL TOOLKIT FOR COMMUNICATION SYSTEMS

Murat Tanyel
Dordt College

ABSTRACT

A typical communication systems course is rich with processes that are best described by block diagrams. While a typical textbook on the subject may provide examples on the applications of these processes, students are motivated when these block diagrams come alive as they implement these processes and are able to test signals at each block. Such an endeavor requires hardware, space and time allocations that not every institution is prepared to commit. The next best teaching tools are computer simulations in which students can observe signals at each stage of the process. Preparation of such simulations is simplified by software development tools tailored for digital signal processing, such as MATLAB which has become the standard package most recent communication systems books have adopted. Recent development of data-driven graphical programming languages has provided an improvement over textual languages such as MATLAB by enhancing the conceptual link from the block diagrams of these processes to their computer simulations. This paper will draw on a toolkit developed for communication systems in a graphical programming environment, namely LabVIEW, that was described in a recent paper [1]. The emphasis of this paper will be on the use of the toolkit in the classroom to demonstrate the effectiveness of this toolkit in illuminating different topics covered in a typical communication systems course, such as the generation and detection of different types of AM (conventional, DSB-SC, SSB), PM and FM, as well as digital signal encoding techniques such as PAM and PCM.

I. INTRODUCTION

This paper is a follow-up on recent papers that describe a simulation toolkit for communication systems [1] and its development with a freshman as the programmer [2]. In those papers we stated that in the absence of hardware that would reinforce the theoretical presentation, computer simulations of the systems described in class are the next available tools to bring these concepts alive. Those papers also describe the particular class environment and the process in which the software development tool, namely LabVIEW, was chosen. Although MATLAB is the standard software tool employed in the areas of signals and systems, as evidenced by the proliferation of books [3-5] devoted to MATLAB based exercises in those subjects, the choice of the software tool is justified in [1, 2, 6]. Another paper written for this conference [7] discusses this choice from an engineering design aesthetics point of view.

This paper will report on the first-time use of the toolkit in EGR 363, *Communication Systems* course offered at Dordt College in Spring 2002. Section II will provide an overview of the in-class presentations that made use of the toolkit with some particular examples. Section III will discuss the student projects and will offer a sample. I will then conclude with a discussion on the present state of the toolkit and on possible future developments.

II. PRESENTATIONS WITH THE TOOLKIT

For small classrooms (where small classes like the Communication Systems are held), the engineering department has a COW. COW stands for *computer on wheels*, (a very appropriate acronym for a college in a rural setting) and is basically a personal computer with a projection system on a cart that may be wheeled around. Every classroom at Dordt has network connection and any document saved on a network drive may be accessed in any classroom. The LabVIEW software development system is installed on the engineering COW. So any *virtual instrument* (or *VI*, a term used for LabVIEW programs) that I develop and save on the network drive may be easily run in any classroom. I use the COW in most *Communication Systems* classes to demonstrate the concepts covered on that day's topic.

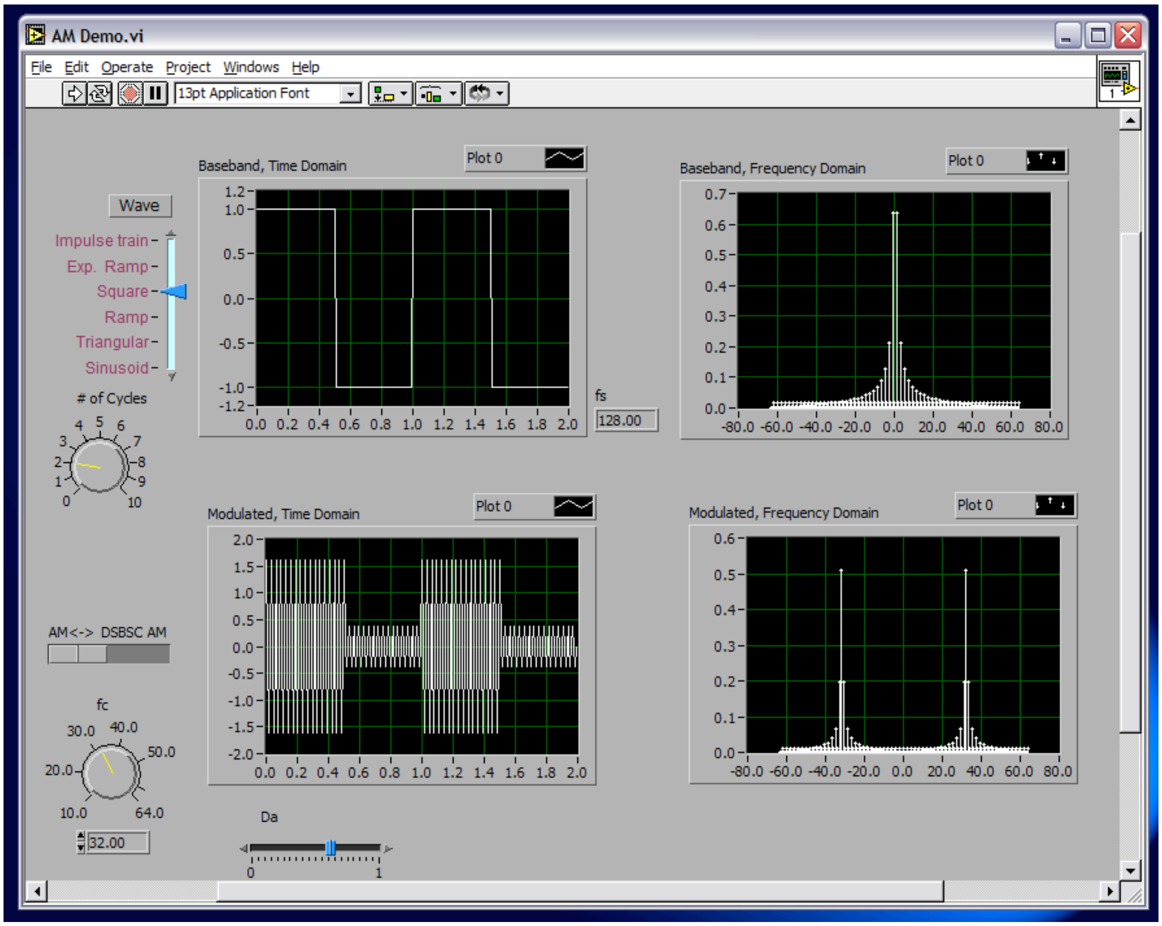

Figure 1: Demonstration of Amplitude Modulation (AM) using the toolkit.

Fig. 1 depicts the front panel of a VI that I use to demonstrate Amplitude Modulation (AM). The effectiveness of this VI is amplified when it is run in continuous mode, in which the VI runs, monitoring it inputs and adjusting its outputs, repeatedly until the stop button is pressed. In this mode, it becomes very easy to change parameters and see the results immediately. For this particular demonstration, one can easily switch through the different kinds of periodic signals as the baseband signal (upper left sliding switch in Fig. 1) and observe how the modulated signal looks both in time and frequency domain. The frequency of the carrier signal (lower left knob) may be varied and the students watch as the replica of the baseband spectrum moves up and down the frequency band in the lower right graph.

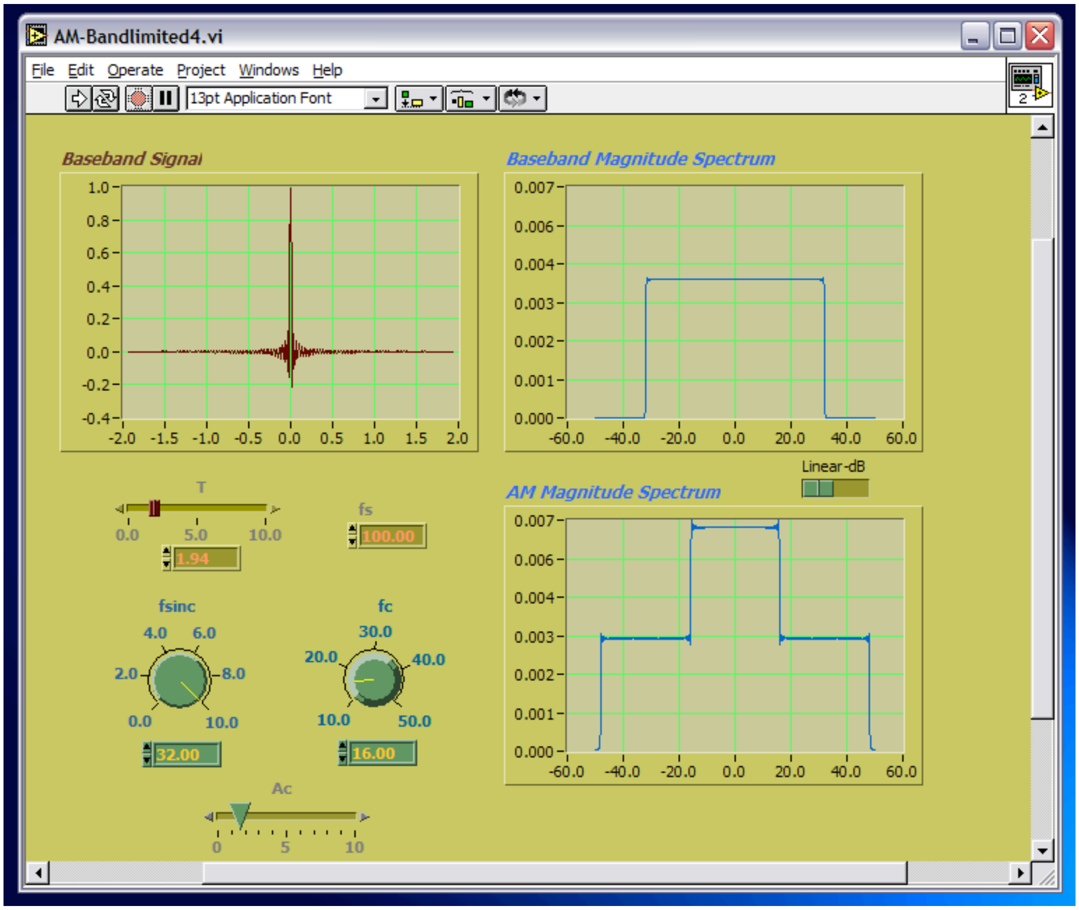

Figure 2: What if the carrier frequency is not high enough? The result is an aliased spectrum.

The continuous run mode renders 'what if' type explorations very effortless. For example, what if the carrier frequency is lower than the highest frequency content of the baseband signal? This is not a question we would encounter in typical communication systems textbooks because practical communication systems would employ carrier frequencies that are orders of magnitude higher than the modulating signal bandwidth, but it is a question worth considering, the answer of which reinforces the concept of aliasing when we consider sampling. An answer is depicted in Fig. 2. This example uses the double sideband suppressed carrier (DSSC) AM technique where the baseband signal is a truncated *sinc*:

$$w(t) = \begin{cases} \mathrm{sinc}(2\pi f t), & |t| < T \\ 0, & \text{otherwise} \end{cases} \quad (1).$$

As T increases, the spectrum approaches the ideal lowpass spectrum and the parameter f determines its bandwidth. In this example, f has been increased beyond the suggested upper limit of 10 Hz to 32 Hz (the left knob labeled `fsinc` in Fig. 2). The effect can be observed in the upper right graph in Fig. 2 where the lowpass spectrum extends to 32 Hz. At the same time, frequency of the carrier has been deliberately kept at 16 Hz (the right knob labeled `fc` in Fig.2). The result is (for lack of a better term) an aliased spectrum as seen in the lower graph of Fig. 2.

Fig. 3 displays the front panel of the VI that I use to demonstrate the detection of AM signals. This VI will employ either the envelope detection method or the product detection method, based on the switch (below the graphs in Fig. 3). This VI would not be all that interesting except that since we have a simulation, we can now observe the signal at intermediate stages. The behavior of the product detector is not all that interesting, since any textbook or a lecture covering the subject will elaborate on it. So I will concentrate on the method of envelope detection, which usually gets a very short treatment.

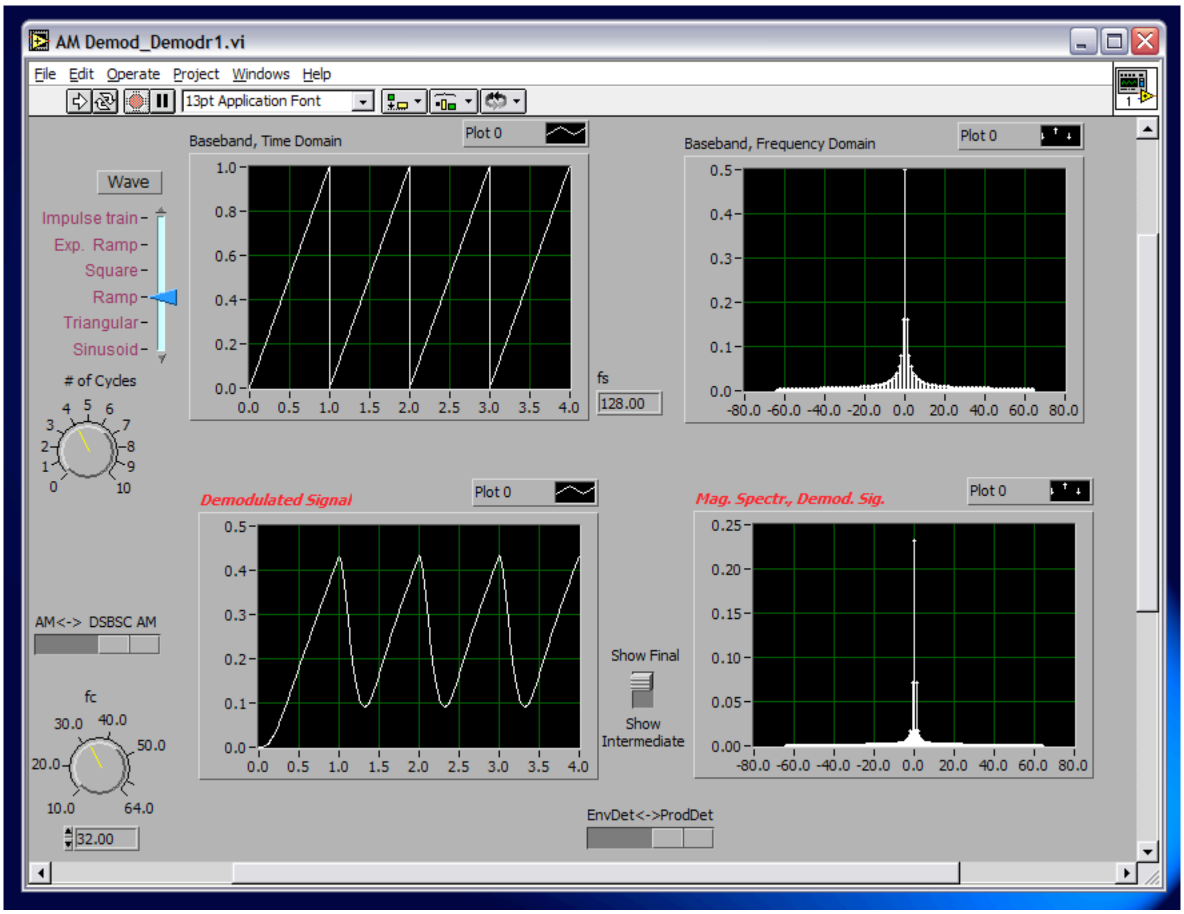

Figure 3: The VI to demonstrate AM demodulation in action.

To refresh the reader's memory, I note that an envelope detector is a diode followed by a lowpass filter. In the simulation, the diode is modeled by a comparator that passes only positive values. So, what happens to the spectrum of a signal after it passes through an ideal diode? Most textbooks on the subject are silent about this and neglect to mention why the lowpass filter is needed from a spectral point of view. When we observe the intermediate signal coming out of the diode (by flipping the switch between the two lower graphs down), and its spectrum (Fig. 4), we can now see that the lowpass filter is needed because we are interested in obtaining the replica of the original spectrum around DC (lower right graph in Fig. 4). There is a way to predict the spectrum of the modulated signal after it has passed through the ideal diode (and the ambitious reader may want to figure it out) and the next stage is to ask the class how they could obtain it analytically, generating further discussion. The accessibility of signals, albeit simulated, coupled with the continuous run mode of the software system lends itself to revealing findings and interesting discussion in class.

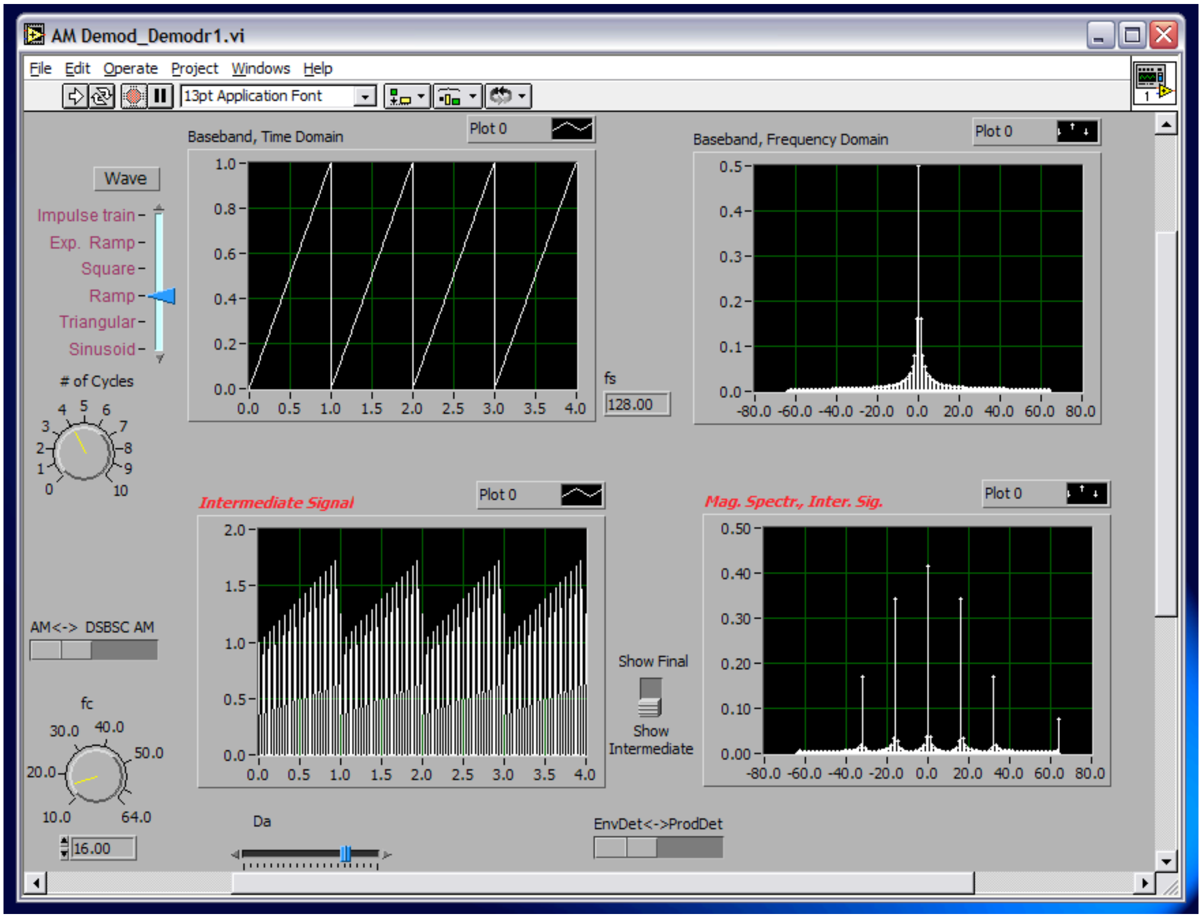

Figure 4: The intermediate signal from an envelope detector and its spectrum may also be viewed.

III STUDENT PROJECTS

This year, students in the communications class were asked to develop some new simulations in addition to the ones they had worked with in class. There were six students in class and they formed teams of two, each working on a topic they were interested in. One group worked on companding, another worked on pulse coded modulation (PCM) while the third worked on *M-ary Phase-Shift Keying* (*MPSK*). MPSK is a method of transmitting digital information over analog lines. The binary information is first converted into a multilevel signal (of *M* levels) and this multilevel signal determines the discrete phase angles of the complex envelope of the MPSK signal. A plot of the permitted values of the complex envelope is called a *signal constellation*. Fig. 5 is a snapshot from the front panel of the MPSK Demonstration VI [8]. In this example $M = 16$, so the signal constellation displays the 16 permitted values of the complex envelope (upper left graph in Fig. 5). Fig. 6 displays the constellation in the presence of added noise, a feature which the students have incorporated into the VI. In this simulation, $M = 8$.

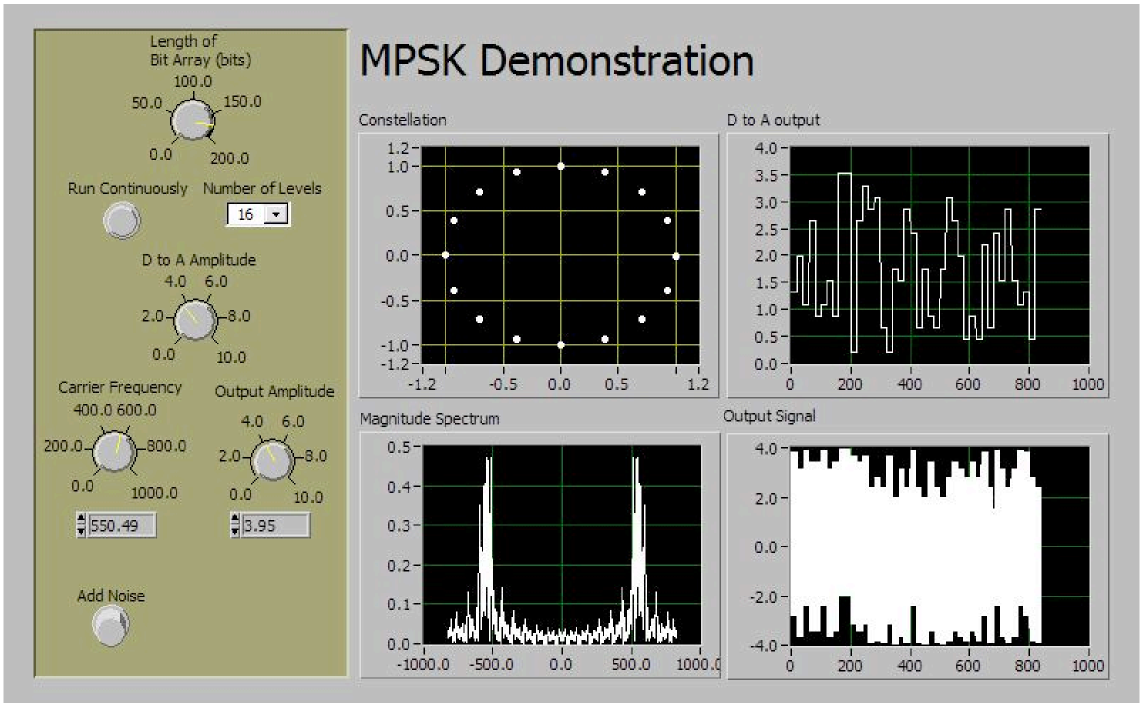

Figure 5: The front panel of the MPSK *Demonstration VI [8].*

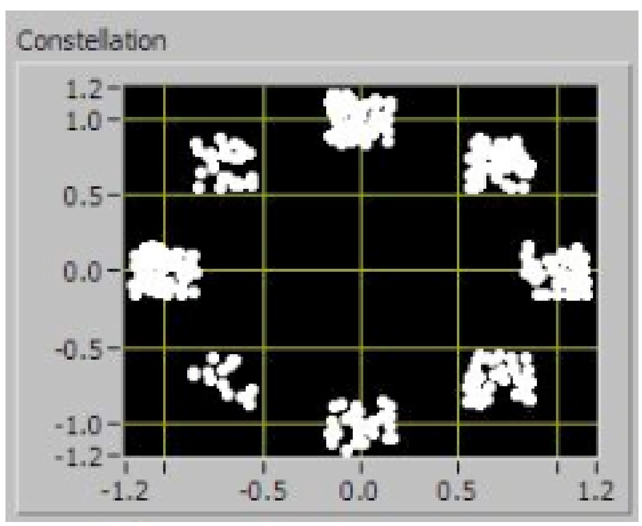

Figure 6: The signal constellation with additive noise. In this simulation M = *8.*

IV. Discussion

The utilization of the tools in this toolkit is understood better with a rudimentary understanding of discrete signals. Since these are computer simulations, hence, simulations of analog processes are valid when the results of the processes all fall within the Nyquist rate determined by the original chosen sampling frequency. The concept of sampling frequency gets interesting when we simulate the sampling of analog signals where we can talk about two sampling frequencies: the sampling frequency of the entire simulation and the sampling frequency at which we retain samples from the simulated signal. I refer to the former as 'simulation clock frequency' and the latter as simply sampling frequency. Since my students take communication systems after DSP, I have chosen to make the simulation clock frequency as one of the inputs and expect the students to pay attention to the Nyquist rate as they implement these simulations and watch for aliasing in their simulations.

Clearly, the list of VIs in the toolkit may be expanded. Spring 2002 offering of EGR 363 has added a few and the work will be continued. There is also the task of documentation to be finished, which always seems to lag in amateur programming projects.

Students also appreciate the exposure to LabVIEW. I have noted that most senior design projects have incorporated LabVIEW in one way or another this year. As a matter of fact, one of the senior design project teams has combined LabVIEW and MATLAB, using the LabVIEW interface to call MATLAB scripts because the LabVIEW interface is aesthetically more pleasing. As I have noted elsewhere [7]: "Part of the enthusiasm is due to the aesthetics of LabVIEW virtual instruments. I have seen many a student revisiting and perfecting the way the front panel looks long after his/her VI has achieved its computational goals and long after the class period has ended."

ACKNOWLEGEMENTS: The LabVIEW software used in this course was purchased through a grant from Johnson Controls in Holland, MI. I would like to thank Johnson Controls for the upgrade of the electronics lab which also benefited the courses mentioned in this paper. I would also like to thank Kathrine Nguru, who programmed some of the subVIs used in the VIs depicted in this paper [2] as well as Howard Gorter and Brian Matherly, whose project is mentioned in this paper.

Bibliography

1. Tanyel, M., "A Virtual Toolkit for Communication Systems", in *Proceedings of 63rd Annual ASEE North Midwest Section Meeting*, Grand Forks, N. D., Sept. 2001.
2. Tanyel, M., Nguru, K., "Preparation of a Virtual Toolkit for Communication Systems," in *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*, Montréal, QC, June 16-19 2002.
3. Proakis, J. G., Salehi, M., *Contemporary Communication Systems using MATLAB*, Pacific Grove, CA: Brooks/Cole (2000).
4. Frederick, D., Chow, J., *Feedback Control Systems using MATLAB and the Control Systems Toolbox*, Pacific Grove, CA: Brooks/Cole (2000).

5. Ingle, V. K., Proakis, J. G. *Digital Signal Processing using MATLAB*, Pacific Grove, CA: Brooks/Cole (2000).
6. Viss, M. and Tanyel, M. "From Block Diagrams to Graphical Programs in DSP," in *Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition*, Albuquerque, NM, June 24-27 2001.
7. Tanyel, M., Adams, C., "On the Aesthetics of Computer Aided Tools for Signal Processing", in preparation for *Proceedings of 64th Annual ASEE North Midwest Section Meeting*, Madison, WI, Oct. 2002.
8. Gorter, H., Matherly, B., *MPSK Demo Project Report*, EGR 363 Project Report, Dordt College, Sioux Center, IA, May 2002.

MURAT TANYEL
Murat Tanyel is a professor of engineering at Dordt College. He teaches upper level electrical engineering courses. Prior to teaching at Dordt College, Dr. Tanyel taught at Drexel University where he worked for the *Enhanced Educational Experience for Engineering Students* (E^4) project, setting up and teaching laboratory and hands-on computer experiments for engineering freshmen and sophomores. For one semester, he was also a visiting professor at the United Arab Emirates University in Al-Ain, UAE where he helped set up an innovative introductory engineering curriculum. Dr. Tanyel received his B. S. degree in electrical engineering from Bo_aziçi University, Istanbul, Turkey in 1981, his M. S. degree in electrical engineering from Bucknell University, Lewisburg, PA in 1985 and his Ph. D. in biomedical engineering from Drexel University, Philadelphia, PA in 1990.