Hands-On Projects for a Freshman Course in Computer Applications in Engineering

Kourosh Rahnamai, Ronald E. Musiak and Thomas Keyser

Western New England College
Springfield, MA 01119

Abstract

All first-year engineering students at Western New England College are required to take a two-credit hour course that deals with computer-aided engineering. This course has two components: computer-aided problem solving and engineering practice. To satisfy the engineering practice portion of the course, all students are expected to present and demonstrate a successful solution to an engineering design problem. The other portion of the course concentrates on developing students' problem-solving skills using engineering computational tools such as MATLAB and MathCAD. In this paper we concentrate on presenting hands-on computer experiments. These are designed to motivate students, enhance problem-solving skills, and introduce students to the latest technology in data acquisition tools as well as data manipulation and processing using MATLAB.

Introduction

Two thirds of the semester is used for teaching the use of MathCAD and MATLAB in solving engineering problems and one third of the semester is used for hands-on experiments. Considering that these are first-year students who have limited or no background in using statistics, some basic statistical concepts such as mean, standard deviation, and uniform and normal distributions are introduced while teaching the use of MATLAB and MathCAD. Two experiments are conducted which require generating a histogram from experimental data and interpreting the histogram to determine the statistical distribution of the data. For the first experiment students throw a die 200 times, record the data and generate the histogram, which should have a uniform distribution. For the second experiment, students build a paper helicopter, release it fifty times from a fixed height, and record the flight time. This is a classical time-of-arrival problem that produces a normal distribution.

The most novel part of this course involves teaching students to perform signal processing without any theoretical background. Students write a MATLAB program to implement a moving-window averaging algorithm on a signal vector of arbitrary length. To validate the integrity of their program, students are required to create a noisy signal with known information content and use their program to extract the information from the noisy data and determine the accuracy of the results.

Moving-window averaging project

Two lecture hours are spent on describing the algorithm using simple vectors with less than ten elements. Then, students are required to perform the following tasks. Write a MATLAB program to perform the moving-window averaging algorithm on a vector of arbitrary length. Test the program with several vectors with a limited number of elements and verify the output of the program with manual calculations. Fully test the functionality of their program on a vector with around six hundred noisy data points. This test vector is generated by adding normally distributed noise with zero mean and standard deviation of 0.2 to a sinusoidal function of magnitude one. Figure 1 shows the noisy and the filtered signal after using the moving-window averaging program. At this point students have created and tested a low-pass filtering subroutine without any knowledge of signal processing. They must then write an engineering report clearly documenting the functionality of their program.

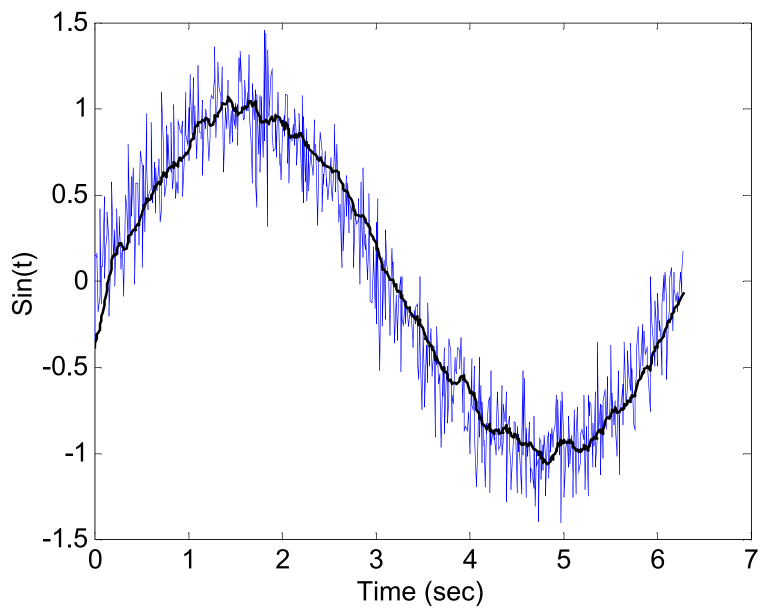

Figure 1. Noisy and filtered sinusoidal signal

Data acquisition and signal processing experiment

The final phase of this exercise is to have students in groups of two to three collect actual noisy data from a plastic extruder machine in a Controls laboratory. The extrusion system consists of a three-quarter-inch extruder, a water trough, and a combination puller/pelletizer. Figure 2 is a schematic diagram of the extruder. The plastics extruder and related control systems hardware and software were funded by an NSF-sponsored equipment grant. The temperatures of the three zones are controlled by three amplifier and linearizer circuits and three Uerotherm temperature controller modules with built in PID controllers.

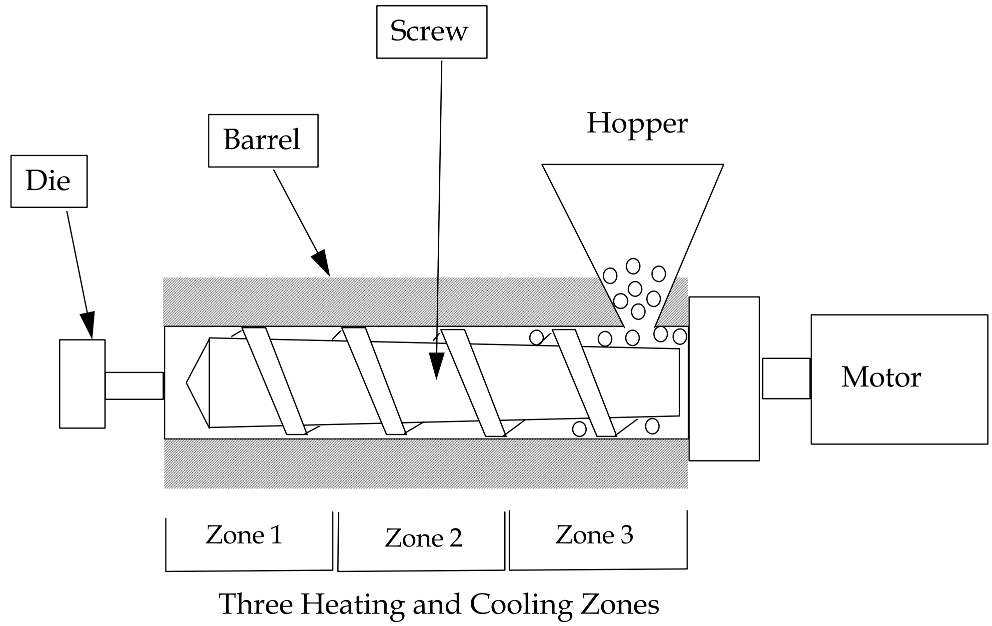

Figure 2. Extruder schematic diagram

For this project the PID parameters are set such that the step response has overshoot and oscillation. Simulink, Real-Time Workshop, and Real-Time Windows Target toolboxes of MATLAB are used to collect step responses of the system, which are directly imported into the MATLAB environment. A National Instruments PCI-MIO-16E-4 data acquisition card is used to collect and store the amplified and linearized thermocouple voltage. Real-Time Windows Target toolbox has special I/O blocks for communicating with this popular board. Figure 3 shows the Simulink program that is used for collecting real time data.

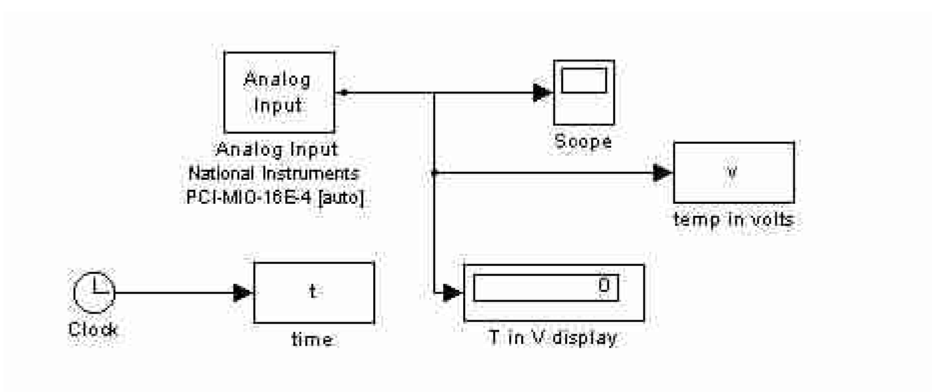

Figure 3. Simulink program for collecting real time data

Since we are interested in temperature versus time data, students manually record temperature versus voltage information for one of the zones of the extruder while the Simulink program is collecting voltage versus time data for the step response of the system. Figures 4 and Table 1 show the collected information.

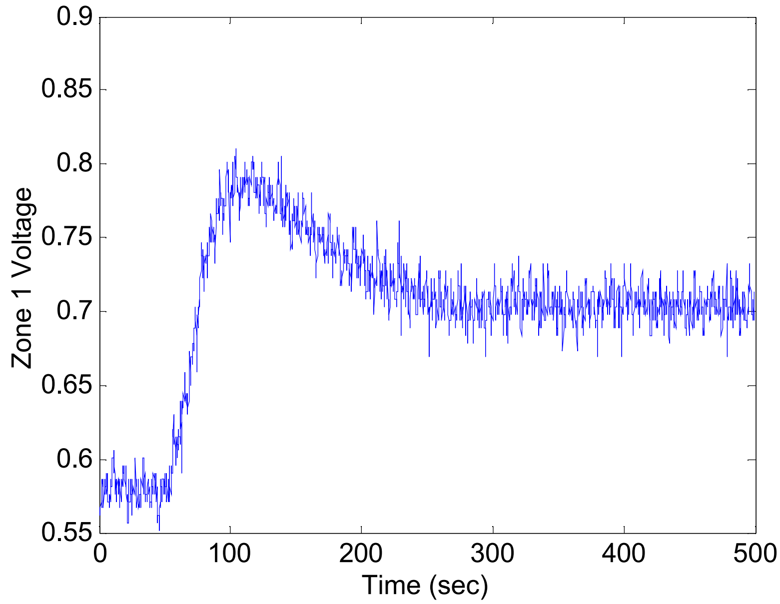

Figure 4. Step response of zone 1

Table 1: Voltage and Temperature Measurements

Temperature (degC)	Voltage (V)	Temperature (degC)	Voltage (V)
80	0.566	96	0.77
82	0.593	98	0.792
84	0.616	100	0.817
86	0.644	102	0.848
88	0.673	104	0.875
90	0.702	106	0.900
92	0.718	108	0.923
94	0.747	110	0.951

Using a regression method that is discussed in class and the built-in *polyfit* function of MATLAB, students find the equation of the line that converts the collected data from voltage versus time to temperature versus time. The best line fit results are shown in Figure 5 and the temperature versus time data in Figure 6.

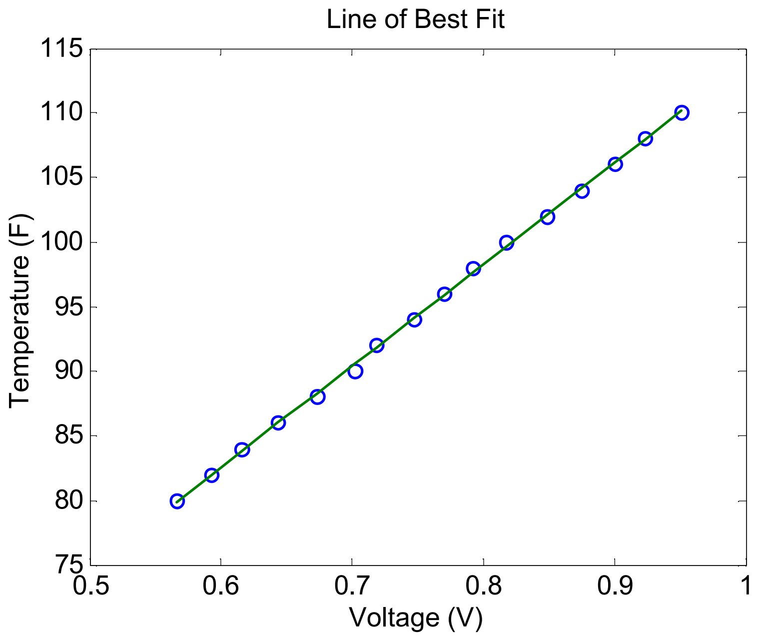

Figure 5. The best line fit for the data in Table 1

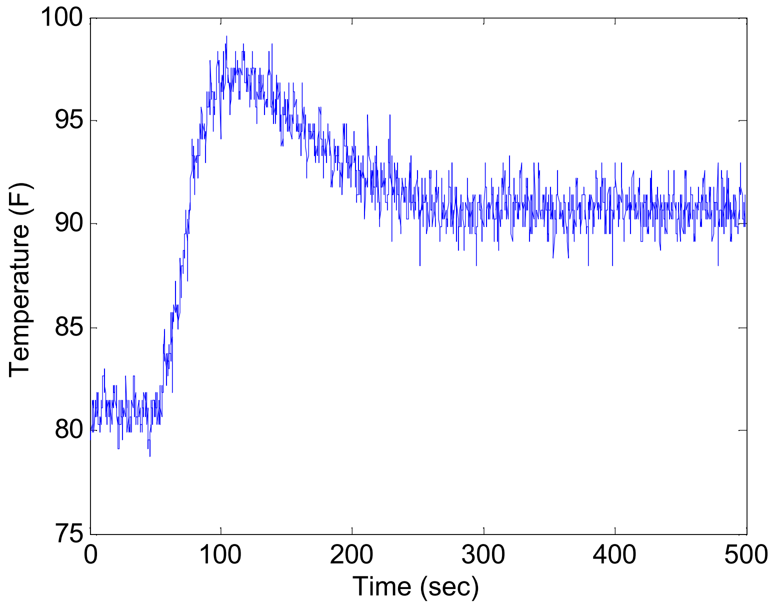

Figure 6. Temperature versus time step response for zone 1

Figure 7 shows the step response of a second-order system. Standard step-response performance criteria such as: rise time T_r, peak time T_p, peak value M_p, steady-state value F_v and percent overshoot are clearly explained to students.

$$P.O = \frac{M_p - F_v}{F_v} * 100 \%$$

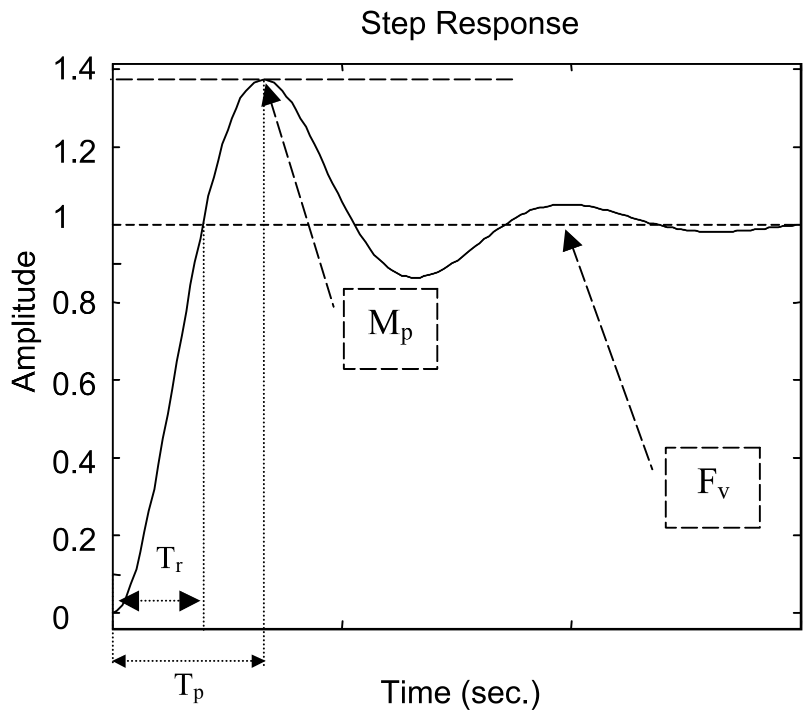

Figure 7. Step response of a second order system

Using noisy data, students estimate: peak time, steady state value, peak value, rise time, percent overshoot, mean and standard deviation for the steady state data, selecting an appropriate time range with justification for their selection. Notice that since data is noisy, students have to use engineering judgment in calculating these parameters. Students use their moving-window averaging program to filter noise and smooth the data so that the rise time, peak time, settling time and steady-state values of the system can be determined more accurately. Students are encouraged to experiment with different window lengths as shown in Figures 8-9

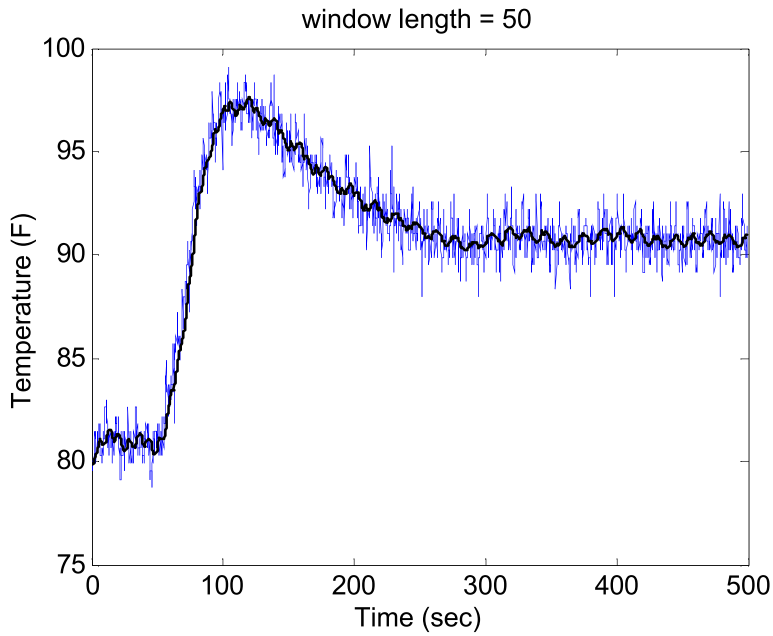

Figure 8. Noisy and filtered step responses

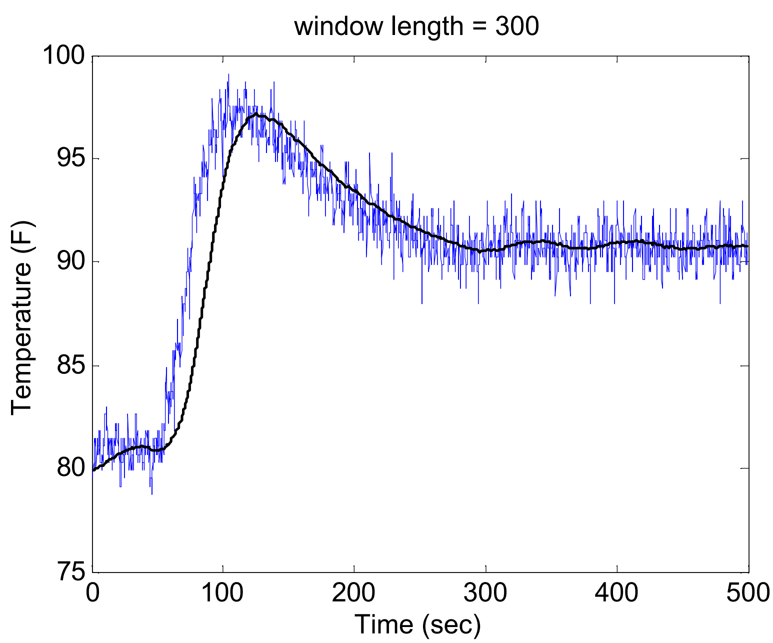

Figure 9. Noisy and filtered step responses

Conclusion

A course in the application of computers in engineering is developed and offered in the School of Engineering at Western New England College with a novel approach for introducing advanced signal processing and engineering concepts to first-year students. We have had very positive feedback from students about our hands-on projects and practical use of engineering concepts.

References

1. Kourosh Rahnamai, "Frequency Analysis Tools for a First Control Course Using MATLAB", ASME Computers In Engineering Conference, San Diego, CA, August 8-12, 1993.

2. Kourosh Rahnamai, "Mathematical Modeling Using Experimental Data", New England Section of the American Society for Engineering Education Conference, University of New Hampshire, New Hampshire, September 30-October 1, 1994.

3. Kourosh Rahnamai, "Electrical Engineering Freshman Encounter", New England Section of the American Society for Engineering Education Conference, University of Connecticut, CT, April 28-29, 1995.

4. Kourosh Rahnamai, and Kevin J. Gorman, "Real-Time Data Acquisition and Controls Using MATLAB" ASME Computers In Engineering Conference, Boston, MA, September 13-17, 1995.

5. Kourosh Rahnamai and John Maleyeff, "Implementation of an Integrated Manufacturing & Controls Laboratory", *Proceedings of the 1997 ICEE-97*, the International Conference on Engineering Education Conference, Chicago, IL, August 14-15 1997.

6. Kourosh Rahnamai and Brian Yanke, "Mathematical Modeling of a DC Motor With Encoder Feedback Using Real Time Simulation", *Proceedings of the ASME Design Engineering Technical Conferences and 21st Computers and Information in Engineering (CIE) Conference*, September 9-12, 2001, Pittsburgh, Pennsylvania.

7. Kourosh Rahnamai and Brian Luce, "Controller Design for a Multi-Fan Hovering System Using MATLAB/XPC Interface", *Proceedings of the Electrical Manufacturing & Coil Winding* Conference and Expo. October 16-18, 2001, Cincinnati, Ohio.

8. Attended Earth Science Technology Conference, June 10-13 2002, Pasadena CA.

9. Kourosh Rahnamai and Andrew M. Masiunas, "Fuzzy Logic Regulated Power Supply for a UHF Digital Television Transmission", *Proceedings of the NAFIPS (the North American Fuzzy Information Processing Society)*, New Orleans, LA, June 27-29, 2002.

10. Kourosh Rahnamai and Steve Northrup, "Fan and Plate Apparatus for Illustrating Stability Concepts Using xPC rapid-Prototyping Environment", *Proceedings of the ASME International Design Engineering Technical Conferences and The Computers and Information in Engineering (CIE) Conference*, September 29-October2, 2002, Montreal, Canada.