# DEVELOPMENT OF AN INTERDISCIPLINARY DATA ACQUISITION AND CONTROL LABORATORY

**Ray Bachnak and Carl Steidley**

Department of Computing and Mathematical Sciences
Texas A&M University–Corpus Christi

## Abstract

This paper describes the development of a new Data Acquisition and Control Laboratory that offers several capabilities, including internet-based experimentation, process variable measurements and analysis, control system modeling and design, signal and image processing, and sensor technology applications. This paper presents the major components of the laboratory, discusses the interdisciplinary activities, and describes the experiments used in a recently taught course.

## Introduction

The Department of Computing and Mathematical Sciences (CAMS) at Texas A&M University-Corpus Christi (A&M-CC) is in the process of developing a new Data Acquisition and Control Laboratory that provides interdisciplinary educational and research capabilities across several science and engineering areas. The laboratory gives Computer Science (CS) and Engineering Technology (ET) students unique training opportunities and benefits by having both groups work on interdisciplinary projects that address real-world problems. The laboratory is also used to teach a three-course sequence, consisting of Principles of Measurements, Control Systems I, and Control Systems II.  The purpose of these courses is to provide students with a solid knowledge of control concepts, techniques, and applications.  The following topics are covered in the three-course sequence (not necessarily in this order):  linearization; types of sensors and measuring devices; sensor and measuring instrument characteristics; calibration; selection of sensors; sensor time response (first order, second order); analog and signal conditioning; display of data; electronic devices; transfer functions; open-loop and closed-loop control; controller design; process characteristics; analog and digital controllers; modes of control; Laplace transform applications; control-loop characteristics; motor speed control; controller mode selection; controller tuning procedures; control-loop characteristics; distributed control systems; supervisory control; networks; data transmission; analysis of control systems: stability, gain and phase margins; response time considerations; understanding of a complete control system and its integration in a real situation; analyzers; and data acquisition systems.  This paper presents the major components of the laboratory, discusses the interdisciplinary educational and research capabilities that the laboratory supports, and describes the experiments developed for teaching Control systems I in fall 2001.

# The Laboratory

The Data Acquisition and Control Laboratory[1] is housed in a new Science and technology building that opened in fall 2001. It consists of the following major equipment: desktop computer stations, data acquisition systems, software (MATLAB and LabVIEW), level and flow measurement system, process control system, wireless data loggers, calibrators, and a variety of test equipment. The following paragraphs briefly describe the features and capabilities of some of the major components.

## Data Acquisition Systems

Each system consists of several components, including a data acquisition board (NI PCI-MIO-16E-4), a 32-channel analog input module (NI SCXI-1102C), a 6-channel analog output module (NI SCXI-1124), mounting terminal block (NI SCXI-1325), isothermal terminal block (NI SCXI-1303), and a 4-slot chassis (NI SCXI-1000). These components are integrated into real-time measurement and control units that interface to desktop computers and process control equipment as shown in Fig. 1. The National Instruments (NI) Software Solutions Department license, which consists of LabVIEW 6.1 (Laboratory Virtual Instrument Engineering Workbench), control and simulation toolkit, Internet, Gmath, and signal processing suites, is available for programming these units. LabVIEW 6.1 offers Internet ready capabilities and supports data acquisition tasks, virtual instrumentation functions, and process monitoring. Students will be able to access virtual instruments from their Web browser and run experiments with built-in Internet tools. They will be able to receive and download live data to their own workstations where they can perform further analysis and processing.

## MATLAB and Associated Toolboxes

MATLAB is one of the most popular programming languages used worldwide for a range of computing tasks in engineering and science. The package available in the laboratory consists of MATLAB 6, Simulink 4, Data Acquisition Toolbox, Signal Processing Toolbox, Control Systems Toolbox, Instrument Control Toolbox, Image Processing Toolbox, MATLAB C/C++ Graphics Library, MATLAB C/C++ Math Library, MATLAB Compiler, and Symbolic Math. These tools offer capabilities in data acquisition, data analysis and exploration, control system design, digital signal processing, image processing, data visualization, algorithm prototyping, modeling and simulation, and application development. Built-in interfaces allow quick access to instruments, files, and external databases. A serial port communication interface offers communication with external instruments and the GUI design tools allow developing interfaces and displays. Simulink is an interactive tool for modeling, simulating, and analyzing dynamic systems. It enables building graphical block diagrams, simulating dynamic systems, evaluating system performance, and refining control system designs. A MATLAB Student version that

has most of the full version capabilities at a reasonable price (around $100.00) is available for students to buy online or from the University bookstore.
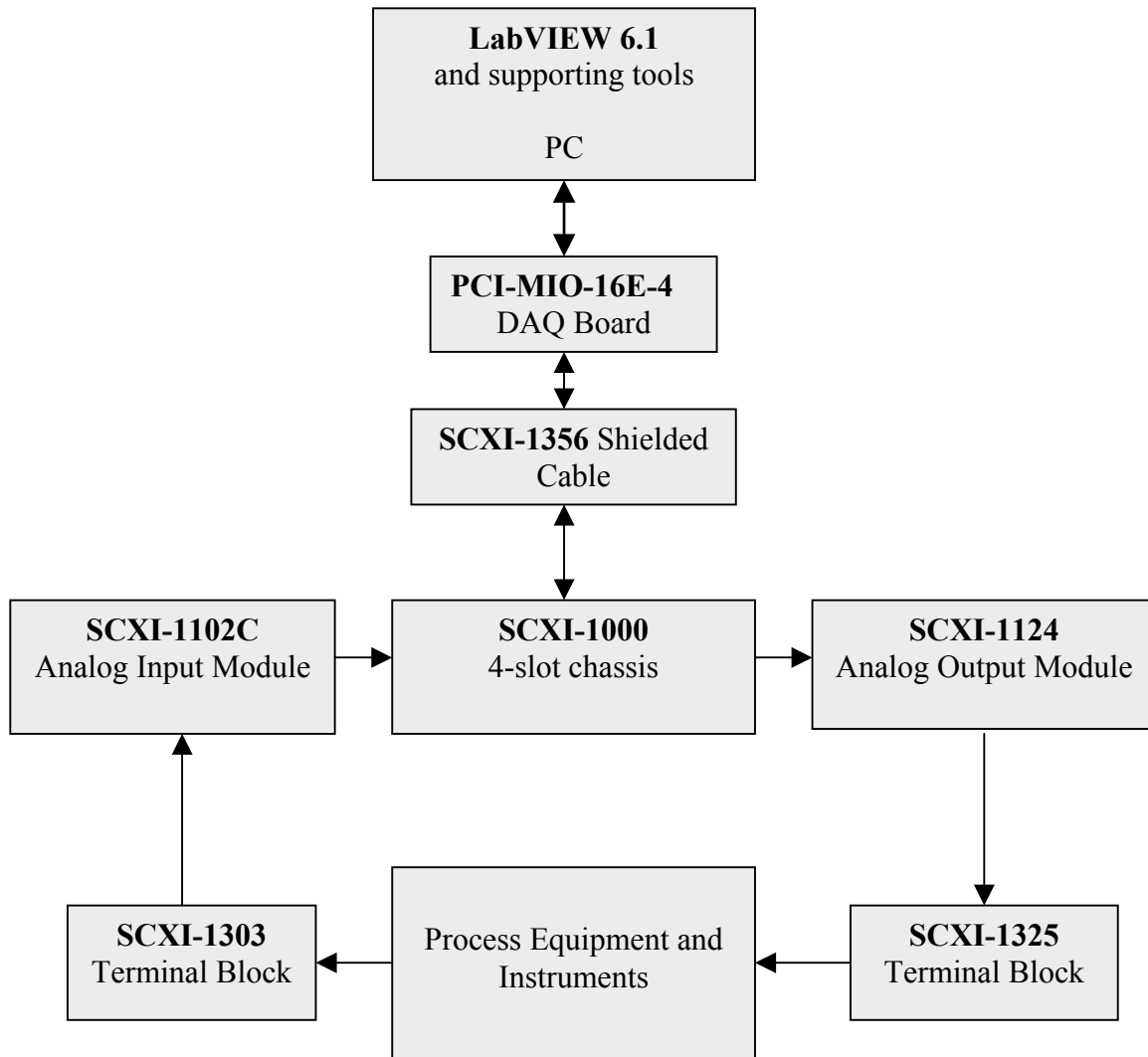
```
          ┌──────────────────────┐
          │   LabVIEW 6.1        │
          │  and supporting tools│
          │                      │
          │         PC           │
          └──────────────────────┘
                    ↕
          ┌──────────────────────┐
          │   PCI-MIO-16E-4      │
          │    DAQ Board         │
          └──────────────────────┘
                    ↕
          ┌──────────────────────┐
          │  SCXI-1356 Shielded  │
          │       Cable          │
          └──────────────────────┘
                    ↓
┌──────────────┐  ┌──────────────┐  ┌──────────────┐
│ SCXI-1102C   │→ │  SCXI-1000   │→ │  SCXI-1124   │
│Analog Input  │  │ 4-slot chassis│  │Analog Output │
│  Module      │  │              │  │   Module     │
└──────────────┘  └──────────────┘  └──────────────┘
      ↑                                    ↓
┌──────────────┐  ┌──────────────┐  ┌──────────────┐
│ SCXI-1303    │← │   Process    │← │  SCXI-1325   │
│Terminal Block│  │ Equipment and│  │Terminal Block│
│              │  │ Instruments  │  │              │
└──────────────┘  └──────────────┘  └──────────────┘
```

Fig. 1  Block Diagram of the Data Acquisition and Control System

**Level/Flow Process Control System**

This system, by Feedback (38-001 Level and Flow Process Control Trainer, including 38-100 Level/flow basic plant, 38-200 Process Interface, 38-300 Process Controller, 38-400 Level Sensor Pack, 38-420 Flow Sensor Pack, 38-490 Digital Display Module, 38-900 Discovery Data Acquisition Curriculum Software), offers level and flow measurements and display.  Temperature, displacement, pressure, and pH sensors and transducers will be added to provide input data to both LabVIEW and MATLAB through the data acquisition systems.

**Process Control Workshop**

This system, by Feedback (37-001 Process Control Workshop using MATLAB), offers real-time control via MATLAB and SIMULINK. This allows using the system for advanced control concepts and research-oriented projects. The process is represented by a heating element controlled by a thyristor circuit that feeds heat into the airstream circulated by an axial fan along a polypropylene tube. A thyristor detector may be placed at one of three points along the tube, sensing the temperature at that point. The volume of air flow is controlled by varying the speed of the fan via a potentiometer. The system allows experimentation with distance and velocity lag, calibration, proportional control, disturbance and system response, and frequency response. MATLAB and its associated toolboxes can be used for real-time digital control, mathematical modeling, transfer function and state-space representation, PID control, and linear quadratic control.

**Wireless Logging System**

The wireless logging system, by Fluke, consists of two wireless data loggers (Hydra 900 MHz wireless system 2625A/WL) communicating to a base station consisting of a wireless modem and windows application software (2625A/WL-700). The software allows a variety of functions, including instrument setup, data collection, creating real time trend graphs, and other data acquisition tasks. The wireless data loggers use spread spectrum radio transmission to reliably send data with a high degree of immunity to electrical noise and interference. Range of transmission may be up to 120 m inside buildings and to over 300 m line of sight. The base station can support up to 20 remote data loggers. The software allows exporting data in real time to Lotus 1, 2, 3, and Excel by using a Dynamic Data Exchange (DDE) link.

**PC Workstations**

The PC workstations are interfaced to the process technology equipment through the data acquisition systems. A direct connection exists between one of the PCs and the process control workshop. Each workstation has a Pentium IV processor, 1.7 GHz, 256 MB RDRAM, 21" Display, two 40 GB hard drives, 1.44 MB 3.5" floppy drive, Ethernet 10/100, 8/4/32X CD-RW & 8X DVD-ROM, zip250 drive. A networked laser printer is also available in the lab.

**Test Equipment**

Test equipment that supports the laboratory includes thermocouple calibrators (Fluke 714), digital thermometers (Fluke 51-2), temperature calibrators (Fluke 724), scopemeters (Fluke 123), hand-held meters (Fluke 73), multi-meters (Agilent 34401A), function generators (Agilent 33120A), power supplies (Agilent E3631A), and frequency counters (Agilent 53131A).

# Interdisciplinary Activities

Course design and laboratory development are major tasks that faculty at institutions of higher education are constantly addressing [1-7]. In recent years, however, the focus has shifted toward creating environments that foster interdisciplinary educational activities and experiences [8, 9]. The Department of Computing and Mathematical Sciences (CAMS) at A&M-CC consists of programs in Computer Science, Engineering Technology, Geographic Information Science, and Mathematics. The focus of Engineering Technology is on the applied aspects of science and engineering and the courses are designed to allow intensive instruction in real world applications. The Computer Science program is applied in nature and the curriculum is oriented toward the technical competencies required of a modern computer professional with emphasis on the development, evaluation, and integration of software systems. The Data Acquisition and Control laboratory affords CS and ET students excellent training opportunities leading toward productive jobs and graduate work in engineering and science. Students using the laboratory will be exposed to the workings of real devices, including designing, building, programming, and testing fully operational electronic and computer-based systems. Topic areas that the laboratory supports include sensor technology, signal conditioning, calibration, process characteristics, data acquisition, system modeling and analysis, system design, control mode selection, feedback control applications, controller design, system response, loop gain and stability, controller tuning, data analysis and representation, data visualization, image and signal processing, algorithm development, automation, system programming, and internet-based experimentation. The laboratory will help CAMS accomplish the following goals:

- Provide an educational environment that fosters learning through effective use of technology and the integration of theory and practice.

- Provide opportunities for undergraduate students to become competent in the use of state-of-the-art instruments and procedures they are likely to encounter in graduate studies or in the work environment.

- Enhance students' abilities to use technology to conduct applied research and scientific investigations.

- Provide an environment that allows ET students and CS students to work on interdisciplinary projects.

Having students work on interdisciplinary design projects that address real-world problems is of high priority. Such projects typically involve integrating software and hardware into fully operational units that meet predetermined specifications. The goal is to prepare students to apply effective problem solving techniques and help them achieve a deeper understanding of scientific and engineering concepts. This includes the ability to define problems, identify alternative solutions, design circuits, program systems, and troubleshoot problems. ET students serve as system designers while CS students serve as systems programmers. The two groups benefit from interacting with each other and the exposure to

both the theoretical and practical aspects involved in the hardware-software integration phase. This phase encompasses many tasks that improve their skills, including interpreting design specifications, software development techniques, and problem troubleshooting. The interdisciplinary approach gives CS students hardware systems understanding that they otherwise do not gain during their undergraduate education.

The laboratory is currently used by a ET student working on a scholarship project to develop software processes and procedures that implement Internet-based experimentation using LabVIEW. Another CS graduate student is working on a project that implements internet-based instrument control through the GPIB port. A second ET student is working on a project that involves exploring the use of the Fluke data loggers for data measurement and control applications (see Fig. 2).



Fig. 2  An Engineering Technology Student Configuring a Wireless Data Logger

A number of problems have been identified as senior design projects that CS and ET students can undertake.  One of the projects involves the use and programming of the wireless logging system in environments where data transmission is crucial but not possible due to hazardous conditions in emergency situations.  The project will involve acquisition and analysis of data of at least five variables.  Another project entails the design and programming of a feedback control system that monitors the level, temperature, pressure, and pH in a tank.  The hardware includes a pump, sensors, valves, and the software involves programming combinations of the proportional, integral, and derivative control modes to select the one that meets a fast response within an acceptable overshoot.  This project will also involve generating trend plots and acquiring and storing data for further analysis.

## Teaching Activities

Although the laboratory is still under development, it was used in fall 2001 to teach a new course, Control Systems I. This course was developed and offered for the first time in fall semester 2001, using two textbooks [10, 11].  It is a four-credit hour course with a weekly laboratory of three hours. Topics covered include control system concepts; open and feedback control; Laplace transform; frequency response; control valves; electric motors; P, PI, and PID modes of control; analog and digital controllers; process characteristics; analysis of control systems; gain and phase margin; and stability.  Laboratory activities included six experiments and four laboratory assignments.  One of the experiments illustrated the use of the Fluke wireless loggers for data acquisition applications.  The other experiments used MATLAB to demonstrate the analysis and design of control systems. This included plotting time domain responses, deriving transfer functions, graphing unit-step responses, computing rise-times, analyzing steady-state errors, plotting Bode diagrams, calculating gain and phase margins, generating Nyquist plots, determining system stability, and utilizing the root locus method for system design and parameter calculations.

Experiment 6 consisted of executing and examining four MATLAB programs, three of which are discussed below.   Students used MATLAB help to learn about the MATLAB commands and functions used in the programs and examined and understood the effects of each instruction. The program of Fig. 3 generates the open-loop Bode plots as shown in Fig. 4.  The open-loop Bode diagrams express the frequency response of a control system when the feedback path is disconnected from the error detector.  A closed-loop control system is considered stable if it has a gain margin of at least 6 dB and a phase margin of at least 40$^o$.   The system is considered marginally stable if it satisfies the stability conditions, but its gain margin is less than 6 dB or its phase margin is less than 40$^o$.  Stability refers to the ability of a control system to dampen out any oscillations that result from an upset. From the Bode plots, students are able to determine the gain and phase margins, the stability of the system, and the value of the magnitude or phase angle at any value of frequency.  The gain margin of this system is about 21 dB and the phase margin is about 37$^o$.  This system is, therefore, considered marginally stable.

```
%Given: G(s) = 38/[(s+0.01)(s+1)(s+20)] and H(s) = 1
numg=[38]                    %numerator of G
deng=conv([1 0.01], [1 1])   %partial denominator of G
dengt=conv(deng, [1 20])     %all denominator of G
G=tf(numg, dengt)            %Create transfer function G(s)
margin(G)                    %plot and compute gain and phase margins
%TF=feedback(G, 1.0)         %transfer function TF=G(s)/[1+G(s)H(s)]
%figure;                     %create a new figure
%bode(TF)                    %plot closed-loop frequency response
```
Fig. 3  A MATLAB program that generates Bode plots

Students can read the gain and phase margins directly from the open-loop Bode diagram. They can also click on any point of the curve to generate a textbox that displays frequency, magnitude, and phase information.  By examining this info and running the program with different parameters, students learn how to design stable systems that meet predetermined specifications.
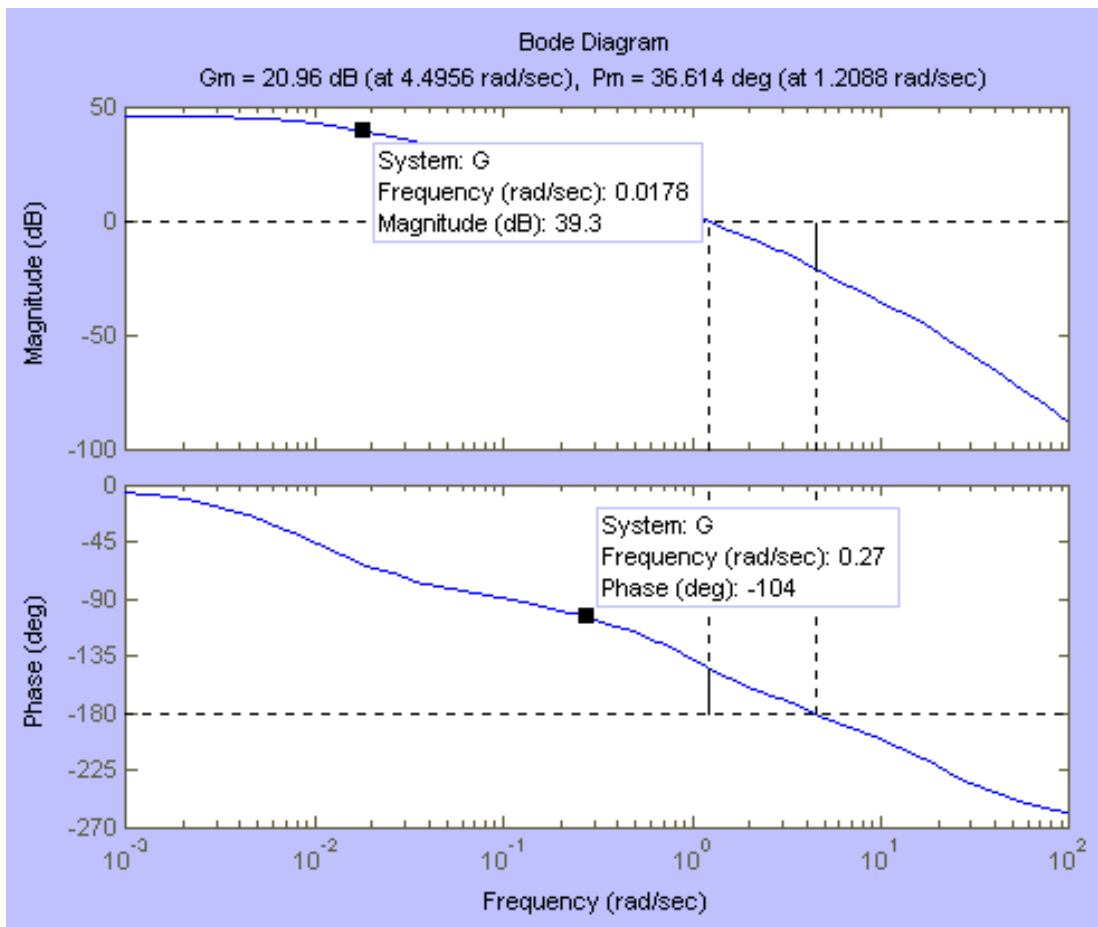


Fig. 4  Bode Diagram Plots Generated by the Program of Fig. X

The program of Fig. 5 determines the overall transfer function of a control system and generates the Nyquist plot shown in Fig. 6.  The Nyquist criterion is another graphical method for determining stability of control systems.  A control system is considered stable if the Nyquist plot does not encircle the gain margin point (-1+j0) and does not encircle the phase margin point $1\angle-140^\circ$.

```
%Given: G(s) = 1.85/(s^2+2s+3) and H(s) = 2s^2+5s+1
numg=[1.85]            %numerator of G
deng=[1 2 3]           %denominator of G
G=tf(numg, deng)       %Create transfer function G(s)
numh=[2 5 1]           %numerator of G
denh=[1]               %denominator of H
H=tf(numh, denh)       %Create transfer function G(s)
TF=tf(G*H)             %overall transfer function
figure;                %create a new figure
nyquist(TF)            %generate Nyquist plot
```

Fig. 5  A MATLAB Program that Generates a Nyquist Plot

Clicking at any point on the curve results in displaying a textbox showing the coordinates of the point and its frequency.  This information, as shown in Fig. 6, represents the open-loop gain and phase values for a given frequency value.
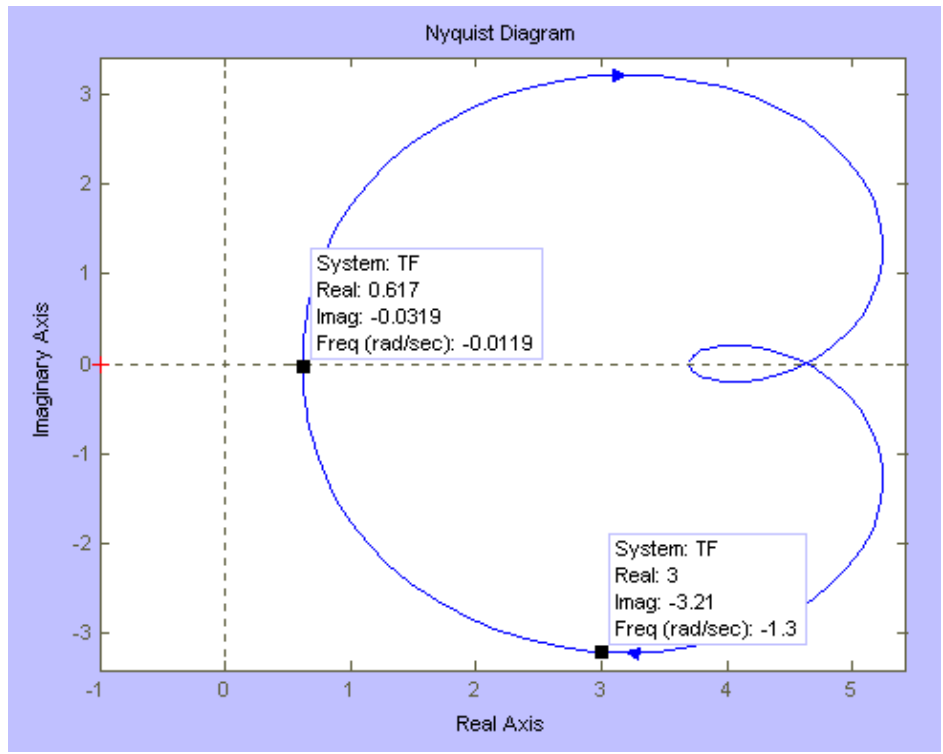


Fig. 6  Nyquist Plot

The program of Fig. 7 determines the transfer function of a control system and draws its root locus, as shown in Fig. 8. The root locus is a graphical method in which all possible roots of the closed-loop characteristic equation are plotted as the controller gain, K, is varied from 0 to infinity. The root locus offers the designer very useful information regarding the relationship between the gain and stability of the system. The value of the gain when the root locus curve crosses the imaginary axis is important. Clicking at that point results in identifying the maximum gain value (kk) for a stable system (refer to Fig. 7). Knowing this value allows the user to enter a value for Kmax (less or equal to kk) to plot the unit-step response for a number of gain values (for K=0.5:0.5:Kmax/6). The step, 0.5, and final value, Kmax/6, can be changed to better suit the problem at hand.

```
%Given: G(s)= 4.8 K/[(2s+1)(0.5s+1)] and H(s) =  1/(0.05s+1)
numgh=[4.8]                 %numerator of G
deng=conv([2 1], [0.5 1])    %denominator of G
dengh=conv(deng, [0.05 1])   %denominator of GH
G=tf(4.8, deng)              %create transfer function of G
H=tf(1, [0.05 1])           % create transfer function of H
GH=tf(numgh, dengh)          %Create transfer function G(s)H(s)
rlocus(GH)                   %compute and plot the root locus
axis([-25 10 -15 15])        %adjust plotting area
%calculate gain value and closed-loop poles for a selected point by user
[kk, clroots]=rlocfind(GH)      %click on a point on the curve and examine info
Kmax = input('Enter max value of K: '); %ask user to input Kmax
for K=0.5:0.5:Kmax/6;        %for K=0 to Kmax/6 (Kmax=12 in this ex),
step=0.5, do
TF=feedback(K*G, H);        %create closed-loop TF
stepresponse=step(TF);       %compute unit-step response
figure                       %new figure
plot(stepresponse)           %plot unit-step response
%hold on     %if "figure is outside loop" and you include this command,
end          %all plots will be on one graph
```

Fig. 7  A Program that Graphs the Root Locus and Plots the Unit-Step Response

Clicking at any point on the curve results in a display of several parameters, including the gain, damping ratio, overshoot, and frequency.   This allows the user to easily determine the gain and expected overshoot for a specified damping ratio. Fig. 8, for example, shows that for a damping ration of 0.46, the overshoot is 19.8%. The program generates a unit-step response each time it executes the loop. The unit-step response may be used to determine the value of K that meets certain design specifications. Specifications may include rise time, overshoot value, time to reach peak, settling time, and residual error.  An exercise that students completed included determining the value of K that results in a unit-step response that has no more than 10% overshoot and a time to reach peak (max value) of no more than 1 s.  The program of Fig. 7 produces four unit-step graphs (not shown here). The graph resulting from k=2 shows a 23% overshoot and a time to reach peak of 19 sec.
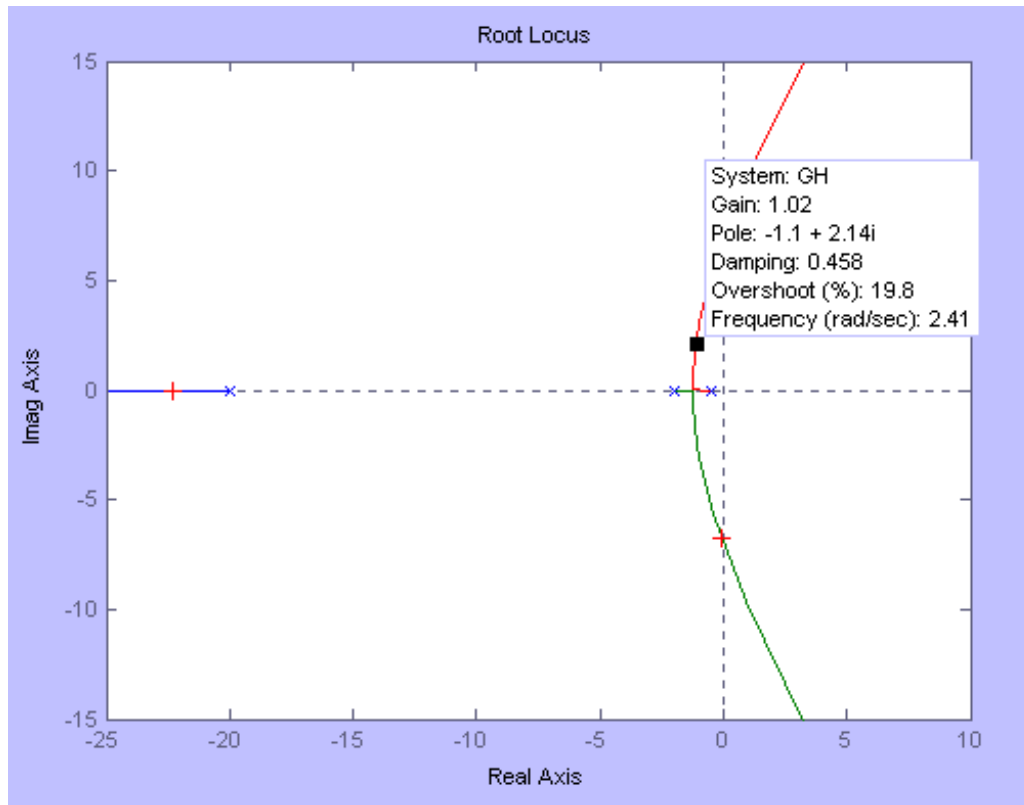
Fig. 8  The Root Locus

## Conclusion

The Data Acquisition and Control Laboratory provides state-of-the-art instruments and software to support teaching several engineering technology courses.  It also furnishes an environment that allows CS and ET students to work on interdisciplinary design projects. The implementation of Internet-based measurement and experimentation will facilitate offering technical courses online.

## References

1.  K. Agarwal, A. Critcher, D. Foley, R. Sanati, and J. Sigle, "Setting up a classroom lab," The Journal of Computing in Small Colleges, Vol. 16, No. 3, March 2001, pp. 281-286.

2.  A. Avila and J. Hinojosa, "Integrating active and cooperative learning strategies to a redesigned microprocessor based system design course," 1999 ASEE Annual Conference Proceedings, Session 2532, 10 pages.

3.  M. Beheshti, "Teaching programming paradigms using a laboratory approach," The Journal of Computing in Small Colleges, Vol. 16, No. 3, March 2001, pp. 144-148.

4.  G. Francia, III, and R. Smith, "The design and implementation of a heterogeneous computer networking laboratory," The Journal of Computing in Small Colleges, Vol. 16, No. 2, January 2001, pp. 231-238.

5.  H. Jack, "Teaching control with PLCs," 2000 ASEE Annual Conference Proceedings, Session 1566, 8 pages.

6.  N. Karlson, "Closed laboratory courses as an introduction to the information science curriculum," The Journal of Computing in Small Colleges, Vol. 16, No. 3, March 2001, pp. 325-329.

7.  D. Smith and R. Douglas, "A revised assembly language programming course for a computer engineering technology program," 1999 ASEE Annual Conference Proceedings, Session 2548, 7 pages.

8.  C. Steidley and T. Coppinger, "Bringing theory and reality together through CIM," The Journal of Computing in Small Colleges, Vol. 16, No. 3, March 2001, pp. 127-131.

9.  R. Stephen Dannelly and C. Steidley, "A student laboratory environment for real-time software systems development," The Journal of Computing in Small Colleges, Vol. 16, No. 3, March 2001, pp. 132-137.

10. Introduction to Control System Technology, 7th Edition, R. Bateson, Prentice Hall, NY, 2002.

11. MATLAB Student Version Release 12, including the Control Systems Toolbox, The MathWorks, http://www.mathworks.com/products/studentversion/index.shtml.

RAFIC BACHNAK
Rafic (Ray) Bachnak is an Associate Professor of Engineering Technology at Texas A&M University-Corpus Christi.  He received his B.S., M.S., and Ph.D. degrees in Electrical and Computer Engineering from Ohio University in 1983, 1984, and 1989, respectively. Dr. Bachnak was previously on the faculty of Franklin University and Northwestern State University.

CARL W. STEIDLEY
Carl W. Steidley is a Professor of Computer Science.  He earned his Ph.D. at the University of Oregon.  His research interests are in robotics and the applications of artificial intelligence.  Steidley has had research appointments at NASA Ames Research Center, Oak Ridge National Laboratories, and with Electro Scientific Industries, Portland, OR.