Teaching Control System Design Concepts Using A Virtual Inverted Pendulum Environment

Richard P. Metzger, Jr. [*] Armando A. Rodriguez [†] Ricardo Aguilar [‡] Chen I. Lim [§]

Department of Electrical Engineering
Center for System Science and Engineering
Arizona State University
Tempe, AZ 85287-7606

October 1, 1996

Keywords: Inverted pendulum, interface, simulation module, graphics module, education module.

Abstract

This paper describes a Windows/C_{++}-based pc environment for simulating and animating an inverted pendulum under automatic control. The program consists of four modules: (i) a program user-interface module, (ii) a simulation module, (iii) a graphics/animation module, and (iv) an education module. The program user interface module allows the user to interact with the program. More specifically, this module permits the user to select the desired pendulum model structure, model parameters, control law structure, control law parameters, reference commands, disturbances, initial conditions, integration routine, and integration routine parameters. The simulation module is responsible for generating a numerical solution for the closed loop dynamics. The graphics/animation module updates plots and animation on the screen using data generated by the simulation module. The education module provides the user with interactive lessons designed to teach fundamental systems and controls concepts. Designed to communicate with MATLAB via DDE files, it is shown how this environment may be used to analyze many "what if" scenarios - observing system dynamics both graphically and through animation, making the environment a very useful tool for teaching control system design concepts.

1 Introduction

The recent revolution in personal computing has seen computing power soar while prices have dropped. As a result, powerful pc's are now commonly available. Given this, it is now possible to exploit the new technology to significantly enhance the way in which systems and controls education is delivered. Because of today's computing power [6], for example, complex simulations - not long ago considered formidable - can now be easily performed. Recent trends in computer speed have also made fast animation of dynamical systems a possibility [3]. In short, today's pc technology now permits the development of new interactive graphical visualization environments which could revolutionize the teaching of systems and controls. This opportunity, has permitted the authors to develop such environments [4], [5], [13], [17], [18], [19], [20]. This paper describes the development of an interactive *inverted pendulum* environment. It is shown how many fundamental system and control concepts may be readily observed and understood with such an environment. The remainder of this paper is or-

---

[*] Currently completing a Master of Science in EE.

[†] This research has been supported, in part, by the National Science Foundation (NSF) through the Coalition to Increase Minority Degrees (CIMD), the ASU Center for Innovation in Engineering Education (CIEE), the Boeing A.D. Welliver Faculty Fellowship Program, and the Intel Corporation. For additional information please contact aar@asu.edu.

[‡] Bachelor of Science in Mathematics.

[§] Currently completing a Bachelor of Science in EE.

ganized as follows. Section 2 describes the different inverted pendulums which are available to the user. Mathematical models are discussed in this section. In Section 3, features within the inverted pendulum environment are described. Section 4 then demonstrates the utility of the environment as an educational tool. Finally, Section 5 summarizes the paper and presents directions for future research.

2 System Description: Models

The inverted pendulum environment currently permits the study of two inverted pendulum systems:

i. an inverted pendulum with a fixed base, and
ii. an inverted pendulum on a moving cart.

Each system is now described.

Inverted Pendulum with a Fixed Base. The model for theinverted pendulum with a fixed base was derived from first principles and results in a second order nonlinear dynamical system. This model does not capture motor dynamics or pendulum flexing. The only system input is a torque applied at the base. A linear model is also available to the user. Of interest here is maintaining the pendulum upright in the presence of a disturbance or the pointing of the pendulum given a desired attitude. The critical variable here is thus the angle which the pendulum makes with the vertical. The nominal nonlinear fixed-base pendulum dynamics are given by:

$$\ddot{\theta} = \frac{g}{l} sin\theta + \frac{1}{ml^2}\tau \tag{1}$$

where θ denotes the angle which the pendulum makes with the vertical and is measured in radians, τ denotes the torque applied at the base and is measured in Newton-meters, m denotes the mass of the pendulum measured in kilograms, l denotes the length of the pendulum measured and g denotes the acceleration due to gravity measured in meters per second^2.

Inverted Pendulum on a Moving Cart. The model for the inverted pendulum on a moving cart was derived from first principles and results in a fourth order nonlinear dynamical system. This model does not capture motor dynamics or pendulum flexing. The only system input is the force applied to the cart. A linear model is also available to the user. Of interest here is maintaining the cart at a desired position with the pendulum upright. The nominal cart-pendulum nonlinear dynamics are give by [14, pp. 104-107]:

$$(M+m)\ddot{x} + mlcos\theta\,\ddot{\theta} - mlsin\theta\,\dot{\theta}^2 = F \tag{2}$$

$$cos\theta\,\ddot{x} + l\ddot{\theta} = gsin\theta \tag{3}$$

where x denotes the position of the cart, θ denotes the angle which the pendulum makes with the vertical and is measured in radians, F denotes the force applied to the cart measured in Newtons, m denotes the mass of the pendulum measured in kilograms, l denotes the length of the pendulum measured in meters, and g denotes the acceleration due to gravity measured in meters per second^2.

Other, more complex, pendulum systems will be considered in the future (e.g. double inverted pendulum).

3 Description of Environment

The inverted pendulum environment is written in Windows/C_{++} [1], [2], [8], [10], [16], [21]. It consists of four modules:

- a program user interface (PUI) module,
- a simulation module,
- a graphics/animation module, and
- an education module.

The environment was divided into modules to facilitate updates and enhancements created by different programmers.

Program User Interface Module. The PUI provides an interface between a user and the inverted pendulum environment program. Written in Windows/C_{++}, the environment provides pull-down menus which permit the user to select which pendulum system model to use. Menus also permit the user to modify critical system parameters in real-time. These parameters include, for example, mass properties, moments of inertias, cart properties, etc. Timers are used to implement this feature [1], [2], [7], [8]. The user may also select amongst different control laws (e.g. proportional, PD, PI, PID, and modern). Initial conditions, reference commands, disturbances, and integration routines (e.g. Euler, Runge-Kutta, etc.) may be selected by the user from a menu. Other menu options for data storage and plotting exist. Data storage routines automatically format saved simulation data for use within other environment modules as well as external programs (e.g. MathWork's MATLAB and Microsoft's Excel). User-selected variables saved from an earlier simulation may be plotted against current simulation data. Multimedia lessons, which use live pendulum video as well as examples from robotics are also accessible through the PUI.

Simulation Module. The main purpose of the simulation module is to accurately solve the appropriate set of ordinary differential equations. The simulation module contains routines required by the different pendulum systems, control laws, integration methods, and data storage routines. All of the environment's pendulum models and control law routines are included in this module.

Graphics/Animation Module. The main purpose of the graphics module is to use data provided by the simulation module and update graphics (i.e. plots) and animations on the screen. Data and plots are displayed within *child windows* [9], [15]. Animation is created using high quality bitmaps generated with Corel's CorelDRAW and CorelPHOTO-PAINT [1].

Education Module. The education module contains routines which implement interactive multimedia lessons. The lessons use text, audio, video, and animation to convey ideas. Users answer questions via interactive menus. Correct answers are supported with a multimedia explantion - including "supportive audio." Incorrect answers are followed up by hints, partial explanations, and additional chances.

4 Educational Utility

The utility of the environment as an educational tool is now demonstrated. Two demonstrations will be presented: one for the fixed-base pendulum system and one for the cart-pendulum system. Additional demonstrations will be presented at the 1997 ICSEE.

Fixed Base Pendulum Demonstration. The model parameters used for the fixed-base pendulum demonstration were: $m = 0.1$ kg, $l = 0.5$ m, and $g = 9.8 \frac{m}{sec^2}$. The controller used was as follows:

$$\tau = K_1 e \tag{4}$$

$$e = r - K_2 \theta \tag{5}$$

$$K_1(s) = k_1 \left[\frac{s+a}{s} \right] \tag{6}$$

$$K_2(s) = k_2(s+b) \left[\frac{100}{s+100} \right] \tag{7}$$

where $k_1 = 0.1250$, $k_2 = 1.6854$, $a = \sqrt{\frac{g}{l}} = 4.4272$, $b = 0.5933$ were used as design parameters. The pole at $s = -100$ in K_2 was introduced only to make K_2 proper. With a second order plant and "essentially" a first order compensation scheme, three dominant closed loop poles result. The design parameters were selected such that the dominant closed loop poles were at

$$s = -2 \pm j1, -4.4272. \tag{8}$$

Here, the last root is a pole of the plant linearized about the vertical (i.e. $\theta \approx 0$). Selecting the zero of K_1 to be $s = -a = -\sqrt{\frac{g}{l}} = -4.4272$ fixes one of the closed loop poles at $s = -4.4272$. Given this, one then obtains the following design equation for the other compensator parameters:

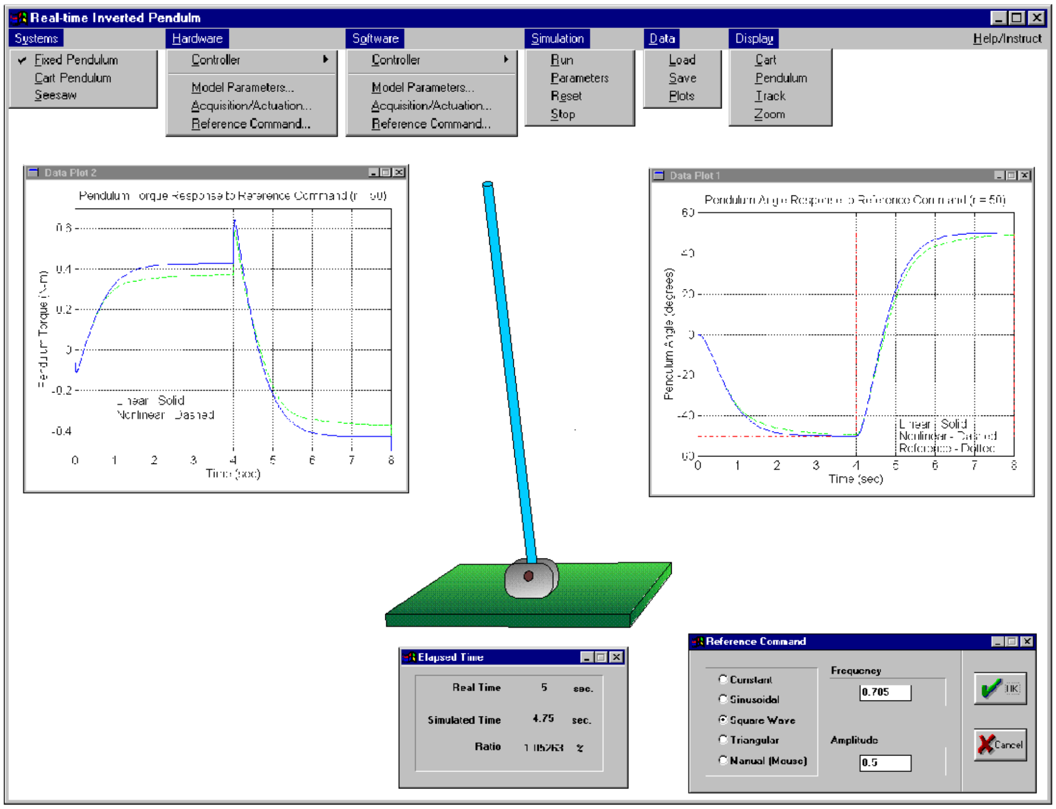
$$s^2 + \left[\frac{k_1 k_2}{ml^2} - \frac{g}{l}\right]s + \frac{k_1 k_2 b}{ml^2} = s^2 + 4s + 5 \tag{9}$$

Figure 1 shows a screen dump of the environment for a reference command of $r = 50$ degrees. Both linear and nonlinear plots are given for θ and τ. Figure 2 shows a screen dump of the environment for different reference commands: $r = 20, 40, 60, 80$ degrees. Once again, both linear and nonlinear plots are given for θ and τ. The plots show that the linear and nonlinear models agree quite well for the smaller reference commands. As the reference commands get larger, the maximum error in the angle plots approach 10 degrees while that in the torque plots approaches a steady state maximum of about 0.2 Newton-meters.

Cart-Pendulum Demonstration. The model parameters used for the cart-pendulum demonstration were: $M = 2$ kg, $m = 0.1$ kg, $l = 0.5$ m, and $g = 9.8 \frac{m}{sec^2}$.

---

[1] CorelDraw and CorelPHOTO-PAINT are trademarks of Corel Corp. CorelDraw is a general purpose drawing/graphics manipulation utility. CorelPHOTO-PAINT is a general purpose image processing utility.



Real Time 5 sec.

Simulated Time 4.75 sec.

Ratio 1.05263 %

○ Constant

○ Sinusoidal

● Square Wave

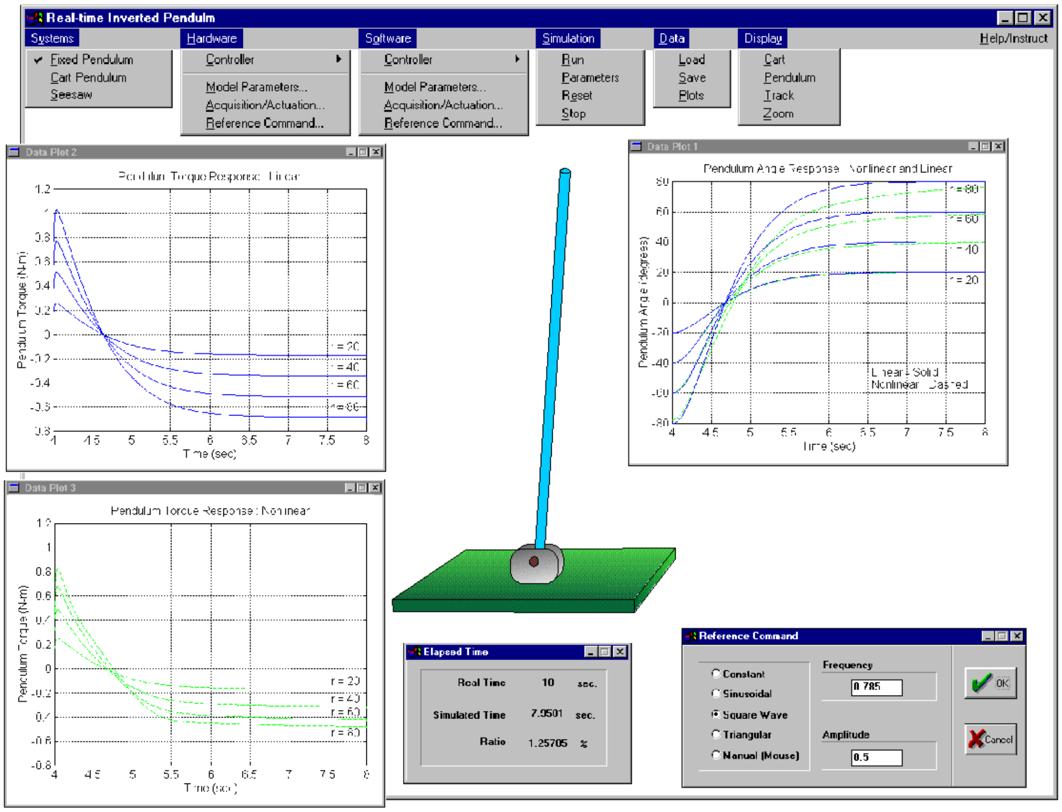
○ Triangular

○ Manual (Mouse)

Frequency 0.705

Amplitude 0.5

Figure 1: Fixed-Base Pendulum Command Following ($r = 50$ degrees)



Real Time 10 sec.

Simulated Time 7.9501 sec.

Ratio 1.25705 %

○ Constant

○ Sinusoidal

● Square Wave

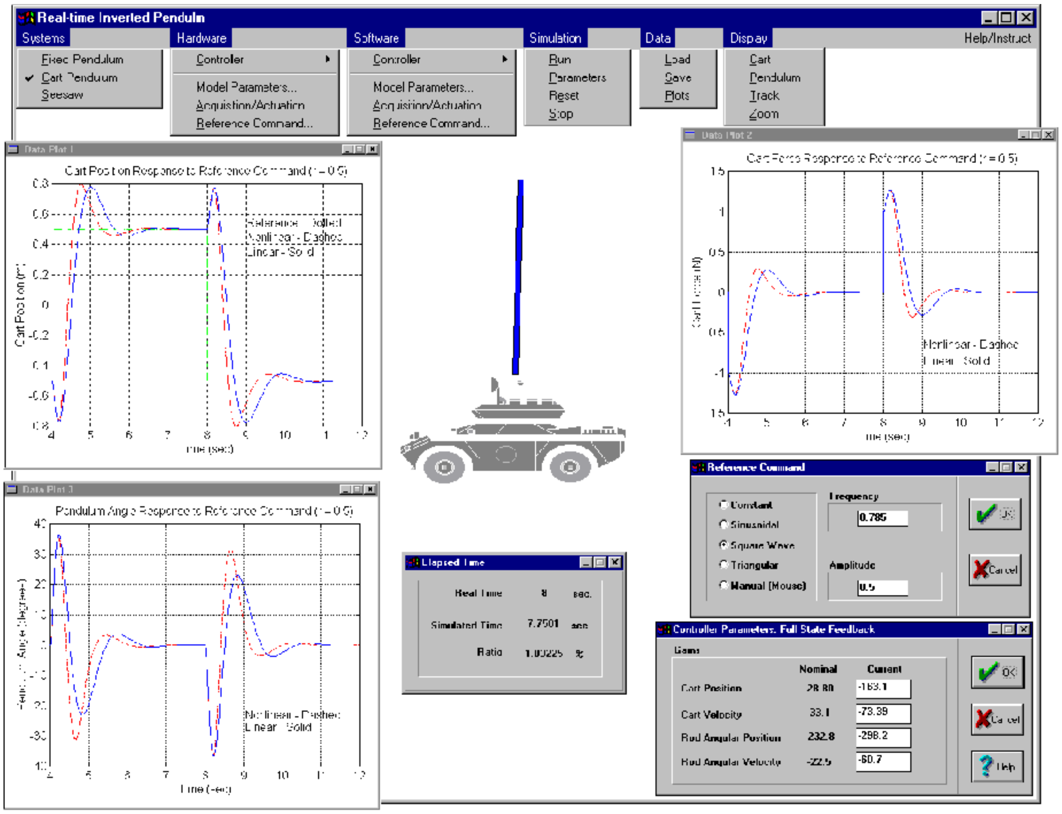
○ Triangular

○ Manual (Mouse)

Frequency 0.785

Amplitude 0.5

Figure 2: Fixed-Base Pendulum Command Following ($r = 20, 40, 60, 80$ degrees)



Elapsed Time

Real Time	8	sec.
Simulated Time	7.7501	sec
Ratio	1.03225	%

Reference Command

○ Constant
○ Sinusoidal
● Square Wave
○ Triangular
○ Manual (Mouse)

Frequency: 0.785

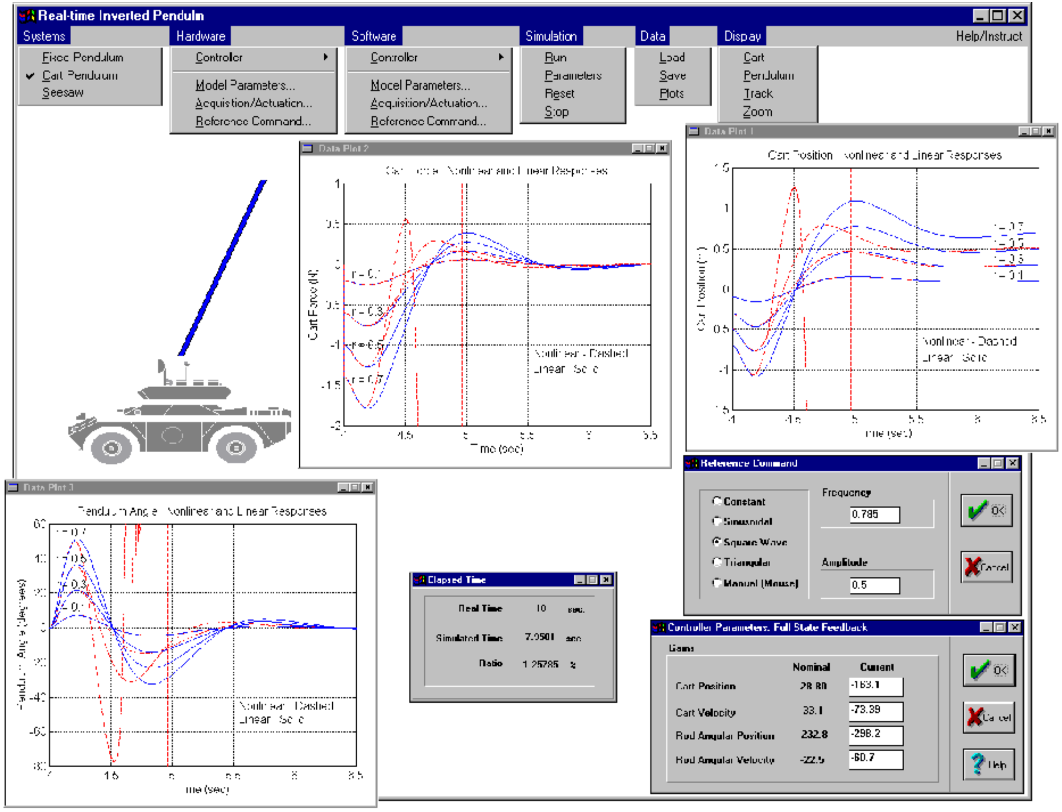
Amplitude: 0.5

Controller Parameters, Full State Feedback

Gains

	Nominal	Current
Cart Position	28.80	-163.1
Cart Velocity	33.1	-73.39
Rod Angular Position	232.8	-298.2
Rod Angular Velocity	-22.5	-60.7

Figure 3: Cart-Pendulum Command Following ($r = 0.5$ meters)



Elapsed Time

Real Time	10	sec.
Simulated Time	7.9501	sec
Ratio	1.25785	%

Reference Command

○ Constant
○ Sinusoidal
● Square Wave
○ Triangular
○ Manual (Mouse)

Frequency: 0.785

Amplitude: 0.5

Controller Parameters, Full State Feedback

Gains

	Nominal	Current
Cart Position	28.80	-163.1
Cart Velocity	33.1	-73.39
Rod Angular Position	232.8	-298.2
Rod Angular Velocity	-22.5	-60.7

Figure 4: Cart-Pendulum Command Following ($r = 0.1, 0.3, 0.5, 0.7$ meters)

The control law used was as follows [14, pp. 787-795]:

$$F = -[g_1 \;\; g_2 \;\; g_3 \;\; g_4]\,[r - x \;\; \dot{x} \;\; \theta \;\; \theta]^T \qquad (10)$$

with design parameters $g_1 = 163.1$, $g_2 = -73.4$, $g_3 = -298.2$, and $g_4 = -60.7$ selected such that the closed loop poles were at

$$s = -2 \pm j3.464, -10, -10. \qquad (11)$$

Figure 3 shows a screen dump of the environment for a reference command of $r = 0.5$ meters. Both linear and nonlinear plots are given for x, θ, and F. Figure 4 shows a screen dump of the environment for different reference commands: $r = 0.1, 0.3, 0.5, 0.7$ meters. For small reference commands the linear and nonlinear responses are in close agreement. Figure 4 shows that for large reference commands, the responses differ greatly - the "actual" nonlinear response going unstable. This occurs, because of the destabilizing $\dot{\theta}^2$ nonlinearity which is not captured in the linear state feedback design (Equation 10). Once again, both linear and nonlinear plots are given for x, θ, and F.

5 Summary and Directions

The inverted pendulum environment provides a flexible educational tool for teaching control system design concepts. Currently, more complex models and control laws (e.g. motors, saturations, anti-windup, etc.) are being implemented. JAVA and VRML implementations are also being explored [11], [12]. Future work will focus on more complex pendulum systems (e.g. double inverted pendulum, etc.) and the continued development of lessons. The goal is to obtain an interactive multimedia environment which can be used to build open-ended problem solving skills while encouraging exploration and self discovery.

References

[1] L. Adams, *Supercharged C++ Graphics*, McGraw Hill, 1992, pp. 207-223.

[2] R. E. Bradford, *Real-Time Animation Toolkit in C++*, J. Wiley & Sons Inc., New York, NY, 1995.

[3] R. Braham, "Math and visualization: New Tools, New Frontiers," *IEEE Spectrum,* November 1992, pp. 17-36.

[4] K.J. Elliott and A.A. Rodriguez, "A Graphical Dual Robot Simulation for the PC," *Proceedings of the 1995 International Conference on Simulation in Engineering Education*, Las Vegas, NV, January 15-18, 1995, pp. 77-81.

[5] M.F. DeHerrera and A.A. Rodriguez, "Trying to 'Shoot' an Evasive Monkey: A Tool for Designing and Evaluating Adaptive Learning Algorithms," *Proceedings of the 1996 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp. 31-36.

[6] L. Geppert, "The New Contenders," *IEEE SPECTRUM: Fast and Powerful*, December 1993.

[7] D. Hearn and M.P. Baker, *Computer Graphics*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[8] L. Heing, *Advanced Graphics Programming Using C/C++*, John Wiley & Sons Inc., 1993, Chapter 1.

[9] L. Heiny, *Windows Graphics Programming with Borland C++*, John Wiley & Sons Inc., New York, NY, 1992.

[10] S. Holzner, *Borland C++ Windows Programming*, Brady Publishing, Indianapolis, IN, 3rd Edition, 1994.

[11] L. Lemay and C.L. Perkins, *Teach Yourself JAVA in 21 Days,* Sams Net Publishing, IN, 1996.

[12] S.N. Matsuba and B. Roehl, *Special Edition Using VRML,* Que Corp, IN, 1996.

[13] R.P. Metzger, K.J. Elliott, and A.A. Rodriguez, "Modelling, Analysis, and Graphical Visualization of a Dual Robot Arm System: A PC Based Environment," *Proceedings of the 1996 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp.175-180.

[14] K. Ogata, *Modern Control Engineering*, Prentice Hall, Englewod Cliffs, NJ, 1990.

[15] S. Oualline, *Windows Programming with Borland C++*, M&T Books, New York, NY, 1993.

[16] S. Potts and T. S. Monk, *Borland C++ 4 By Example*, Que Corporation, Indianapolis, IN, 1994.

[17] A.A. Rodriguez and R. Aguilar, "Graphical Visualization of Missile-Target Air-to-Air Engagements: An Educational Tool for Designing and Evaluating Missile Guidance and Control Systems," *Journal of Computer Applications in Engineering Education*, Vol. 3, No. 1, 1995, pp. 5-20.

[18] A.A. Rodriguez and M.F. DeHerrera, "Modeling, Simulation, and Graphical Visualization of a Twin Lift Helicopter System Under Automatic appear in the Proceedings of the 1996 Conference on *Frontiers In Education,* Salt Lake City, Utah, Nov 6-9, 1996.

[19] M. Sonne, A.A. Rodriguez, "A PC-based Graphics System for the Evaluation of Missile Guidance and Control Laws", Proceedings of the American Control Conference. Baltimore, MD. June 29-July 1, 1994, pp. 2726-2730.

[20] M. Sonne, A.A. Rodriguez, "PC's in the Design and Evaluation of Guidance and Control Systems for Missiles," Proceedings of the ICSEE. Tempe, AZ.

[21] P. Yao, *Borland C++ 4.0 Programming for Windows*, Random House Inc., New York, NY, 1994.