Modeling, Simulation, Animation, and Real-Time Control (MoSART) Environments: Tools for Education and Research

Chen-I Lim [*] Richard P. Metzger Jr. [†] Armando A. Rodriguez [‡]

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

March 21, 1998

Keywords: MoSART, helicopter, real-time interactive simulation and visualization.

1 Abstract

This paper describes a set of Microsoft Windows '95/NT, Visual C++, Direct-3D based software environments for simulating and visualizing several different dynamical systems. Different simulation and animation models may be selected by the user for each environment. Users are also able to alter model and controller parameters "on the fly" - thus allowing them to quickly examine different scenarios. The environments take advantage of Direct-3D to produce high-quality three-dimensional real-time animated graphical models of the systems. Real-time plotting and graphical indicators are also employed to help users abstract-out key phenomena. The environments also accommodate data exchange with MATLAB. Users may readily export simulation data to MATLAB and use the associated toolboxes for post-processing and further analysis. The MoSART environments are valuable tools for enhancing both education and research. Examples are presented to illustrate their utility.

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

[†] Currently completing a Masters 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), Western Alliance to Expand Educational Opportunities (WAESO), the ASU Center for Innonvation in Engineering Education (CIEE), the Boeing A.D. Welliver Faculty Fellowship Program, the Intel Corporation, and Honeywell. For additional information, please contact aar@asu.edu.

2 Introduction: The Need for Interactive Visualization Tools

The modern personal computer has evolved into a serious platform for engineering design. The rapid increase in performance-per-price of personal computers is fast outpacing that for older generation systems. Indeed, many institutions that once employed expensive and unwieldy centralized mainframe and workstation hardware are now replacing them with inexpensive and versatile desktop computer systems. The ongoing "PC Revolution" is placing enormous computing power at the hands of more people than ever before. This current level of PC-technology now permits the creation of highly advanced PC-based engineering design tools.

While general-purpose simulation and analysis packages already exist (e.g. MATLAB, SIMULINK, MATRIX-X), there is still a need for advanced visualization tools. The recent advances in PC-technology have provided development platforms capable of creating such tools. This has permitted the Modeling, Simulation, Animation and Real-Time Control (MoSART) team at ASU to create several system-specific visualization environments [2], [3], [4], [5], [6], [7], [8], [11], [17]. In contrast to existing general-purpose animation/visualization packages (e.g. Working Model, DADS/Plant), our efforts have been focused upon developing optimized environments for specific systems. This has permitted us to exploit the characteristics of the particular system under study and to provide efficient simulation and analysis tools for that system. In this paper, we describe the central framework of these environments and present several examples of their utility.

Contribution of Work. This paper demonstrates how affordable state-of-the-art PC technologies may be combined to develop high quality system-specific MoSART environments that are useful for enhancing the education and research process. Specifically, in this paper, we will focus on an interactive MoSART cart base inverted pendulum environment, fixed base inverted pendulum environment, a robot-manipulator environment, and a helicopter environment. These environments permit students and control system engineers to more easily analyze, design for, and visualize the performance of specific systems via real-time and faster than real-time animation. Through a user-friendly graphical interface, users can alter model, controller, and signal parameters on-the-fly while commands are issued by either a program function generator or a user-controlled joystick.

The system-specific nature of our environments permit us to exploit the dynamical and animation model structures which are not exploited in general purpose simulation and animation packages. As such, the environments discussed within offers significant improvements over such general-purpose packages (e.g. Working Model, DADS/Plant) that have recently emerged. One major feature of our environments is the ability to import data from MATLAB for post-simulation animation and to access the MATLAB 5.0 engine - giving the environment direct access to the MATLAB toolbox suite of numerical tools (e.g. optimization, system identification, signal processing, etc.). These features make the environments an ideal centerpiece for highly extensible virtual design and test platforms.

The remainder of this paper is organized as follows. In Section 3 the various modules which make up the core of the environments are described. The utility of the environments as educational and research tools is demonstrated in Section 4. Finally, Section 5 summarizes the paper and presents directions for future work.

3 Description of Environment

This section describes our interactive MoSART environments. Although separate, each environment shares a common framework and modularity. We will discuss the development background of the software, and then describe the functionality and content of the individual modules of the environments.

The MoSART environments are interactive applications for simulating and visualizing a variety of complex systems. They run on any PC-compatible computer running Microsoft Windows '95 or NT. For optimum performance, fast Pentium processors (our development was done on 200 MHz Pentium-Pro machines) and 3D-accelerated video cards are recommended.

The environments were developed with Microsoft Visual C++ version 5.0 [10], and are built around the Microsoft Foundation Classes (MFC) framework. The use of the MFC framework provides a flexible and powerful function-base and assists in structuring the code. The basic visualization functions are enabled by the Windows General Drawing Interface (GDI) [10] while the three-dimensional rendering uses Direct-3D [19].

The MoSART Environments are organized as four core modules: the Program Interface Module (PIM), the Simulation Module (SIM), the Graphical Animation Module (GAM), and the Help/Instruct Module (HIM). Each of these modules are now discussed:

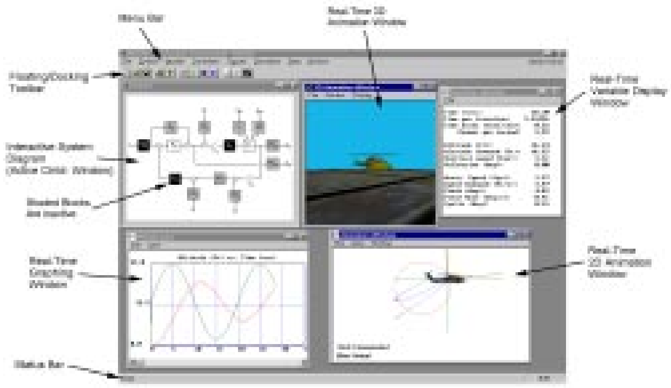

Figure 1: Program Interface

3.1 Program Interface Module

The PIM contains the general user-interface (UI) elements featured by the environment (Figure 1). By utilizing the MFC framework, standard Windows '95 user-interface elements are available to the user. The basic *Frame-Window* contains a *menu*, a *status bar*, as well as a shortcut *floating/docking-toolbar*. Through this interface the user can: Select/edit a simulation model, Select an animation model, Select/Edit the input signals of the simulation, View/Change the simulation parameters, Save/Load the simulated data, or Post-process the simulated data in MATLAB.

The active *child window* contains a block-diagram representation of the selected system. The user may edit parameters associated with any of the available components simply by clicking the mouse on the box, or through the menus.

3.2 Simulation Module

The simulation engine numerically solves the ordinary differential equations describing the particular system and is capable of simulating a system, which is based on a general block structure. The user can specify the use of different integration methods, control-laws, inputs, and other parameters of the system. The simulation can accept user changes in real-time, even as the simulation is progressing. Several integration methods are supported, including basic Euler and a 4th order Runge-Kutta. More complex simulations may be developed and can take advantage of direct access to MATLAB 5.0 scripts and toolboxes via the MATLAB-engine. Better than real-time simulations are possible on standard personal computers.

3.3 Graphical Animation Module

The ability to visualize the simulation is a key feature of this environment. Several visual representations of the simulation are available to the user, including: real-time variable display windows, real-time graphing windows, 2-dimensional animation windows, and 3-dimensional animation windows.

3.4 Help/Instruct Module

This module allows for the inclusion of on-line tutorials. It also contains basic help information for using the environment. With the inclusion of direct links to Hypertext-Markup Language (HTML) format documents, users can call up help and information directly from the environment. This allows the creation of detailed on-line tutorials and project guidelines.

4 Utility of Environments

In this section we will present examples from several interactive MoSART environments which demonstrate their utility for education and research. Specifically, we will focus on: an interactive MoSART cart base inverted pendulum environment, fixed base inverted pendulum environment, a robot-arm environment, and a helicopter environment.

4.1 Cart Base Inverted Pendulum Environment

The interactive cart base inverted pendulum environment is shown in Figure 2. This environment was created to simulate an inverted pendulum on a cart, which can move laterally. The environment's control objective is to keep the pendulum inverted while minimizing the error between the cart's current position and a reference position by applying a force to the cart. The environment implements a constant gain full state feedback controller [9] as well as a user-input controller allowing users to issue cart force commands with the computer's mouse. The controller generated force can be observed through the animated force vector indicator. Using the user interface users can change pendulum model parameters, cart model parameters, and controller parameters in real-time. Simulation data is observable through the simulation variable dialog box, data graphs, or 3D animation.

4.2 Fixed Base Inverted Pendulum Environment

Figure 3 shows the interactive fixed base inverted pendulum environment. This environment simulates an inverted pendulum experiment built at Arizona State University. The environment's control objective is to minimize the error between the pendulum current angle and a user selected pendulum angle reference command by applying a torque to the pendulum. The environment implements a default controller with feedback and feedthrough dynamics and a user input controller. Users can issue pendulum base torque commands with the computer's mouse. Using the environment's user interface users can change pendulum model parameters (pendulum length, mass, and base friction) and controller parameters in real-time. Simulation data can be observed through the simulation variable dialog box, data graphs, or 3D animation.

4.3 PUMA 560 Robotic Manipulator Environment

MoSART's interactive PUMA 560 robotic manipulator environment is shown in Figure 4. This environment simulates an Unimate PUMA 560 robot arm using a six degree-of-freedom nonlinear model [18]. The environment's control objective is to minimize the error between the end-effector's position and a desired position. To facilitate single input single output (SISO) systems all but a single joint can be locked resulting in a SISO system with the unlocked joint torque input and unlocked joint angle output. Through the user interface's menus and dialog boxes users can change robot arm model parameters and controller parameters in real-time. The current simulation data can be observed through the simulation variable dialog box, data plots or 3D animation.

4.4 Helicopter Environment

Figure 5 shows a screen capture from the interactive MoSART helicopter environment. This environment has been set up to simulate a Sikorsky UH-60 Blackhawk Helicopter near hovering trim. A proportional controller has been implemented with negative feedback, and the user is allowed to issue commands via joystick, which is prefiltered with a low-pass filter (unity dc gain, 1 second time constant). The figure shows the system's response to a user-introduced step command. The user is able to issue this command while observing the helicopter's pitch response on the pitch-indicator window, or in the 3D animation window (shown in Figure 1).

5 Summary and Directions

This paper has described Microsoft Windows '95/NT, Visual C++, Direct-3D based software environments for simulating and visualizing specific dynamical systems. Advanced animation and visualization aids are used to 'abstract-out' critical behavior. With the ability to interact with the simulation in real-time and see the results quickly, the user can rapidly grasp the underlying relationships in analyzing and designing controllers for the particular systems.

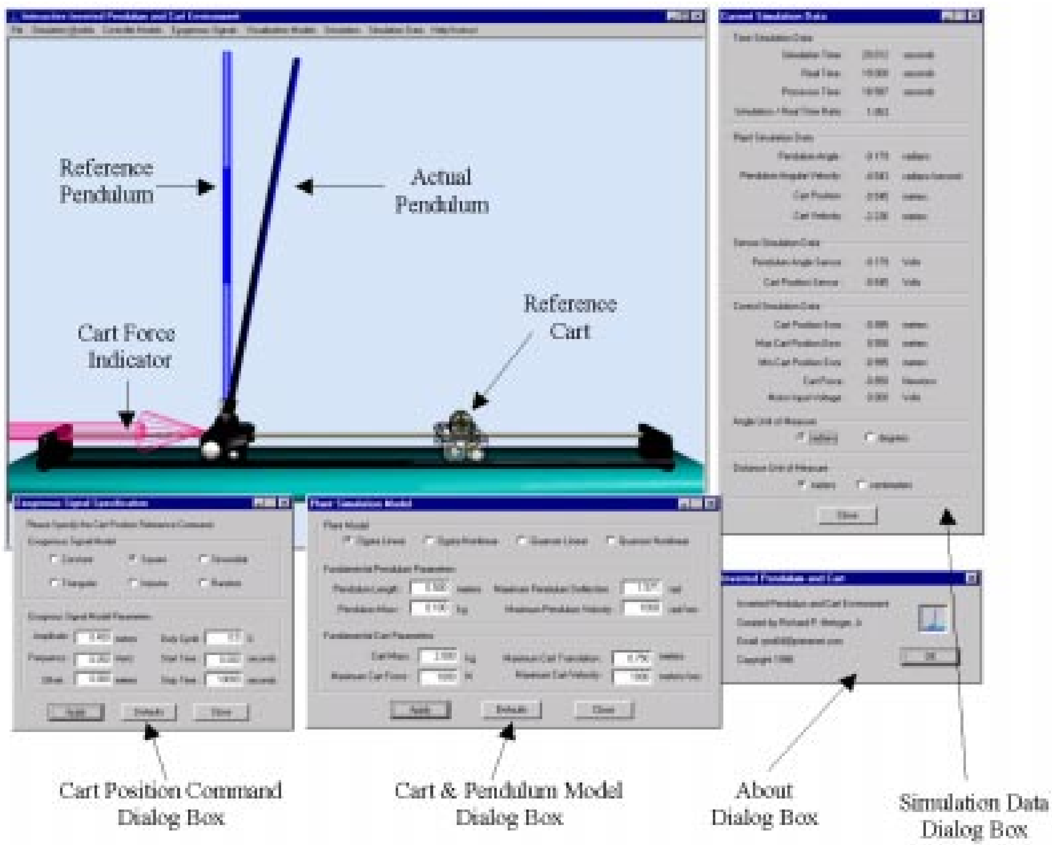

Figure 2: Cart Base Inverted Pendulum Environment

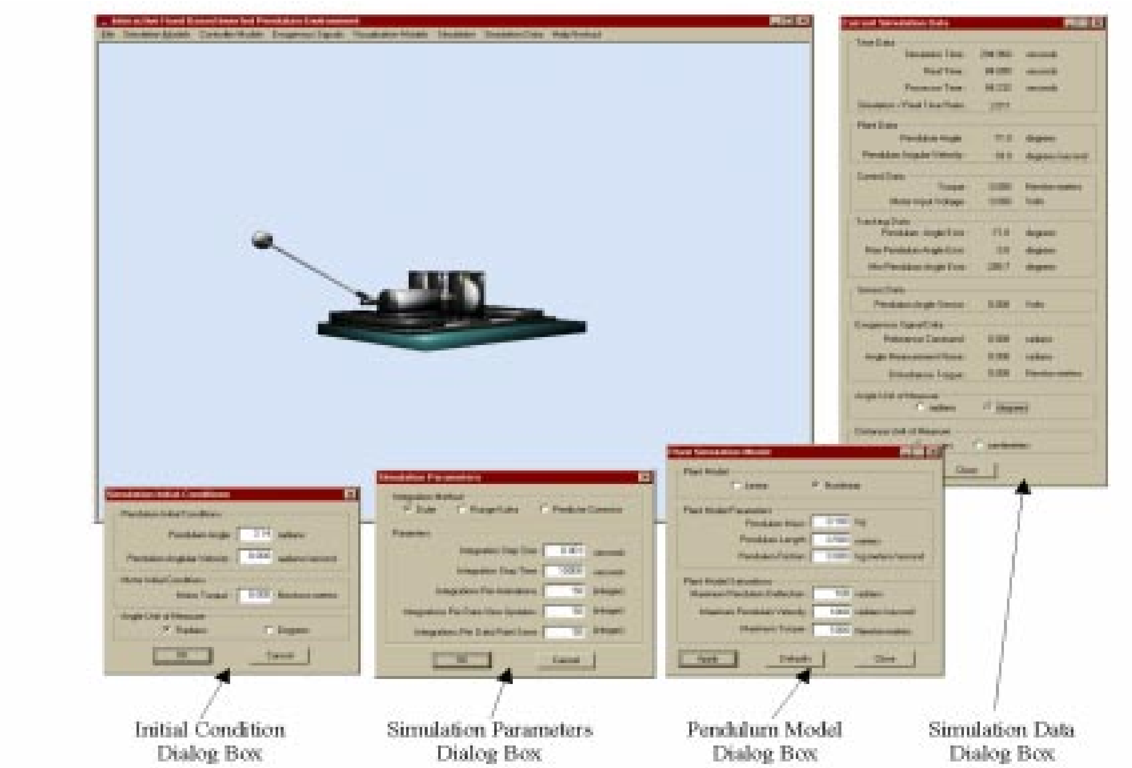

Figure 3: Fixed Base Inverted Pendulum Environment

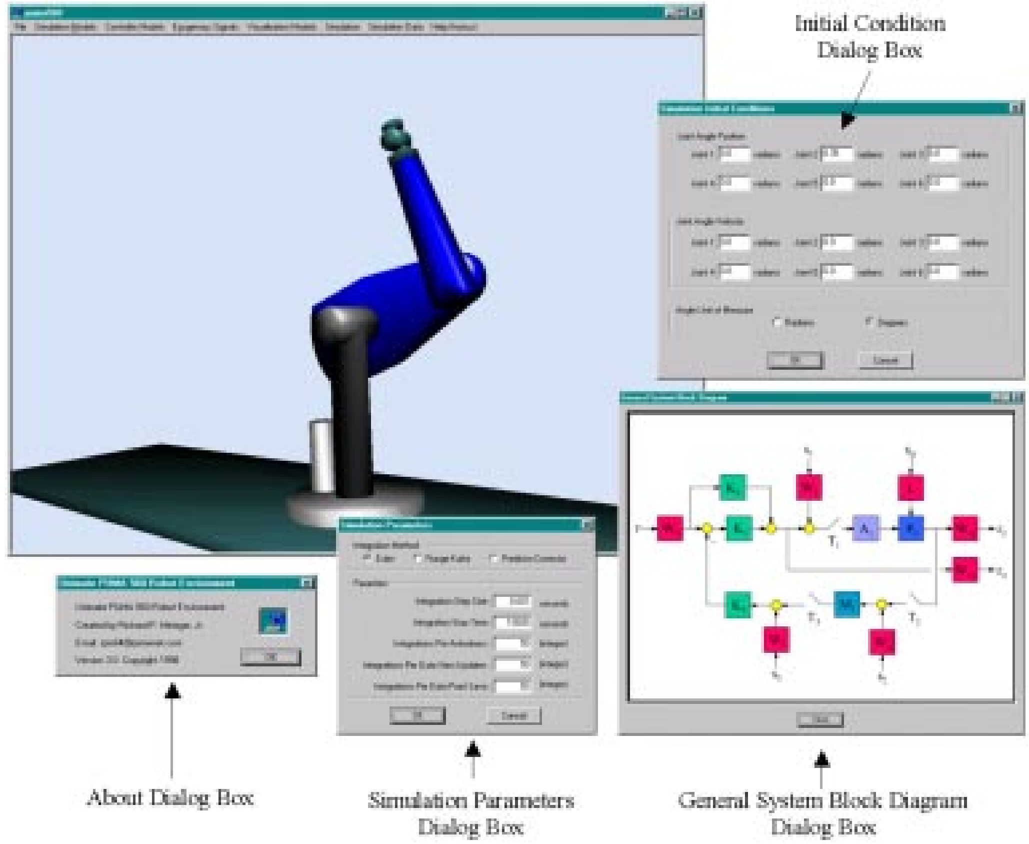

Figure 4: PUMA 560 Robotic Manipulator Environment

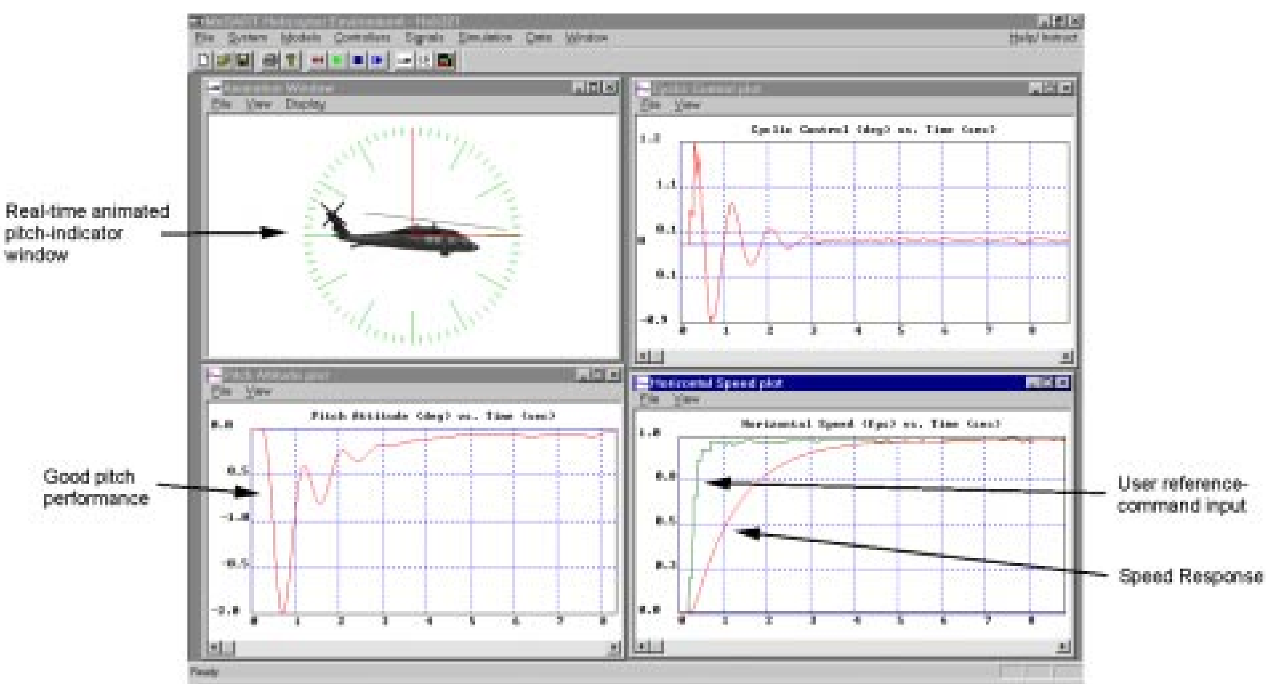

Figure 5: Helicopter Environment

These environments share a common core that is both flexible and modular. Thus, this framework may be easily extended to accommodate other systems. The use of Direct-3D also allows new animation models to be rapidly developed and implemented. Additionally, integration with the MATLAB engine provides access to advanced analysis tools. The extension of these environments as well as the creation of new environments will be the focus of future work.

References

[1] A.R.S. Bramwell, *Helicopter Dynamics*, John Wiley & Sons Inc., New York, NY, 1976.

[2] 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 (ICSEE)*, San Diego, CA, January 14-17, 1996, pp. 31-36.

[3] M.F. DeHerrera, A.A. Rodriguez, and R.P. Metzger Jr, "Teaching Systems and Controls Using a MATLAB-Based Interactive Environment," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 71-76.

[4] M.F. DeHerrera, A.A. Rodriguez, R.P. Metzger Jr, and D. Cartagena, "Modeling, Simulation, and Graphical Visualization of a Liquid Level control System," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 57-62.

[5] C.I. Lim and A.A. Rodriguez, "Modeling, Simulation, Animation, and Real-Time (MoSART) Control of Helicopter Systems," *Proceedings of the 1998 ACC*, Philadelphia, PA, June 22-24, 1998.

[6] R.P. Metzger Jr., A.A. Rodriguez, "Modeling, Simulation, Animation, and Control for a Single Robotic Manipulator," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 77-82.

[7] R.P. Metzger Jr., A.A. Rodriguez, R. Aguilar, C.I. Lim, "Teaching Control System Concepts Using a Virtual Inverted Pendulum Environment," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 134-139.

[8] 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 ICSEE*, San Diego, CA, January 14-17, 1996, pp. 175-180.

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

[10] C. Petzold, "Programming Windows 95," *Microsoft Press,* 1996.

[11] M. Roberts, M.F. DeHerrera, and A.A. Rodriguez, "The Evasive Monkey: An Environment for Evaluation Adaptive Learning Algorithms," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 51-56.

[12] A.A. Rodriguez, *Multivariable Control of a Twin Lift Helicopter System using the LQG/LTR Design Methodology*, Master's Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 1987.

[13] 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.

[14] A.A. Rodriguez and M. Athans, "Multivariable Control of a Twin Lift Helicopter System using the LQG/LTR Design Methodology," Proceedings of the American Control Conference, Seattle, WA, June 18-20 1986, pp. 1325-1332, Invited Paper.

[15] A.A. Rodriguez, M.F. DeHerrera, and R.P Metzger, "An Interactive MatLab-Based Tool for Teaching Classical Systems and Controls," Proceedings of the 1996 Conference on *Frontiers In Education,* Salt Lake City, Utah, Nov 6-9, 1996.

[16] A.A. Rodriguez and M.F. DeHerrera, "Modeling, Simulation, and Graphical Visualization of a Twin Lift Helicopter System Under AutomaticControl: An Educational Tool," Proceedings of the 1996 Conference on *Frontiers In Education,* Salt Lake City, Utah, Nov 6-9, 1996.

[17] A.A. Rodriguez and C.I. Lim, "Interactive Environments for Teaching Systems and Controls: The Need for Educationsl Tools," *Proceedings of the 1997 ICSEE*, Phoenix, AZ, January 12-15, 1997, pp. 9-14.

[18] T.J. Tarn, A.K. Bejczy, X. Yun, and X. Ding, "Dynamic Equations for the Six-Link Puma 560 Robot Arm," Laboratory Report SSM-RL-86-05, Department of Systems Science and Mathematics, Washington University, St. Louis, MO, Oct 1986.

[19] S. Trujillo, "Cutting Edge Direct 3D Programming," *Coriolis Group books,* 2nd Edition, 1996.