Toward the Development of an Interactive Modeling, Simulation, Animation, and Real-Time Control (MoSART) Hardware/Software Testbed for a Tilt-Wing Rotorcraft

by

John Koenig
IEEE Student Member #40351269
Arizona State University

March 15, 1999

Contents

1 Abstract

This paper describes the development of an interactive Modeling, Simulation, Animation, and Real-Time Control (MoSART) hardware/software testbed for a small-scale tilt-wing rotorcraft which is being developed at Arizona State University. The hardware setup includes a test stand which has been designed and built to model the hover-flight mode of the tilt-wing vehicle. The software environment combines the use of real-time 3-D animation, graphing, and control with a flexible, user-friendly interface to facilitate the modeling and control system design process. Nonlinear and linear models of the pitch-axis motion of the test stand are developed and it is shown how the MoSART software environment, in conjunction with MATLAB, is employed in the design of the controller. The potential to link the software and hardware to form an integrated and powerful testbed for control system development is discussed.

2 Introduction

Tilt-Wing Rotorcraft Project. The High-Speed Autonomous Rotorcraft Vehicle (HARVee) project at Arizona State University is an undergraduate research project, the purpose of which is to design, build, and fly a small-scale tilt-wing aircraft. This vehicle, when flown successfully, will demonstrate the feasibility of the tilt-wing concept for small aircraft applications. A tilt-wing aircraft has the ability to perform hover flight like a helicopter while retaining the high-speed cruise capabilities of a conventional airplane. This class of aircraft accomplishes this feat through the rotation of its wing from a position perpendicular to its fuselage (for hover flight) to a position parallel to its fuselage (for cruise flight). The current configuration of the student designed HARVee aircraft is as shown in Figure 1 below, from which a cruise-only version of the tilt-wing vehicle has been built and flown successfully.

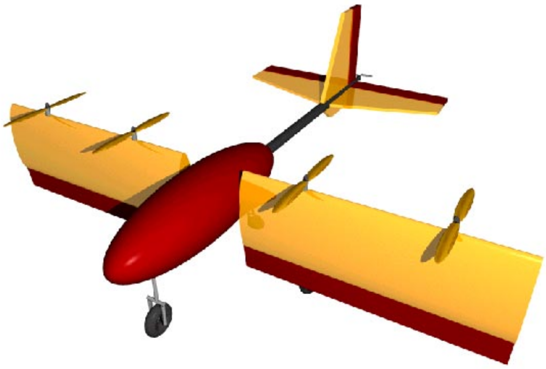


Figure 1: Current HARVee Configuration

Hover Test Stand. A hover-only vehicle and testing rig are being developed for a hover testing program, the purpose of which is to understand the dynamics of the hover flight mode. It is realized that it will be difficult to control the vehicle during hover because of a lack of aerodynamic damping. In addition, the conversion process will involve precise control of several inputs at once in order to prevent the airfoil from losing lift during the wing rotation process. For these reasons, a computer controlled stability augmentation system is being developed to assist the pilot with control of the vehicle. To assist in the development of the stability augmentation system, a test stand has been designed and constructed to model the dynamics of hover flight. The test stand is a T-shaped platform which allows up to 4 degrees-of-freedom (pitch, roll, yaw, and z-axis translation) and which utilizes two main engines, a pitchfan, and ailerons to model all of the control inputs of the full vehicle in hover flight. Rate gyros (which measure angular velocity) are employed to feed back data on the motion of the test stand. A top view of the layout of the test stand is provided in Figure 2.

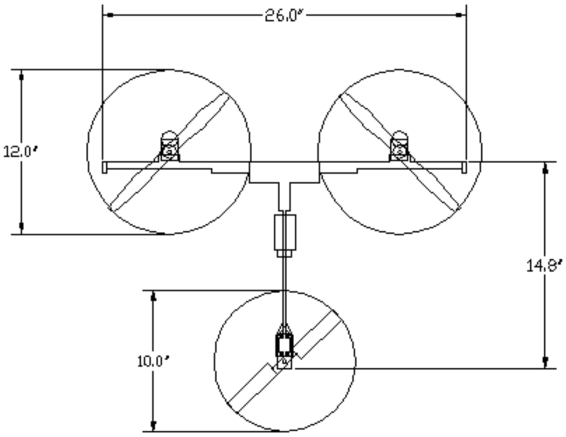


Figure 2: Top View of HARVee Test Stand

MoSART Software Environment. A software environment has been developed for the tilt-wing test stand to provide Modeling, Simulation, Animation and Real-Time control (MoSART) of the tilt-wing test stand. This environment is a useful tool for simulation of mathematical models, visualization using 3-D graphics, and controller development.

Interactive Hardware/Software Testbed. The MoSART software environment can be connected to the hardware test stand through the use of a data acquisition board in order to provide real-time data collection and control. This interactive hardware/software testbed will be used for the characterization of the dynamics of the test stand (which models the tilt-wing vehicle) and as a proving ground for tilt-wing rotorcraft control system development.

3 Mathematical Models: Pitch-Axis Motion

This section describes the development of non-linear and linear mathematical models of the pitch-axis motion of the hardware test stand. A system model is a set of equations which adequately predicts the behavior of a system to a set of known inputs [1]. The behavior under consideration is that of pitching motion and the input is the thrust generated by the pitchfan.

Physical Model Description. The test stand will be represented as a three point-mass system which rotates about a fixed point for the development of the mathematical model. Limiting the focus of the initial modeling to that of the pitching motion reduces the problem to a 2-dimensional one, thus simplifying the analysis and allowing the focus to be on the use of the MoSART environment in the design process (see Section 6). A 4-DOF model will be developed later and implemented into the software environment in the same way this pitch-only model has been.

The three point-masses represent the gross distribution of mass of the test stand, as shown in Figure 3. The definitions of the variables in the figure are given in Table 1. The trusts due to the pitchfan and main engines are represented by point forces at the center of each propeller. This representation is simple, but it is sufficient to understand the dynamics of the pitching motion.

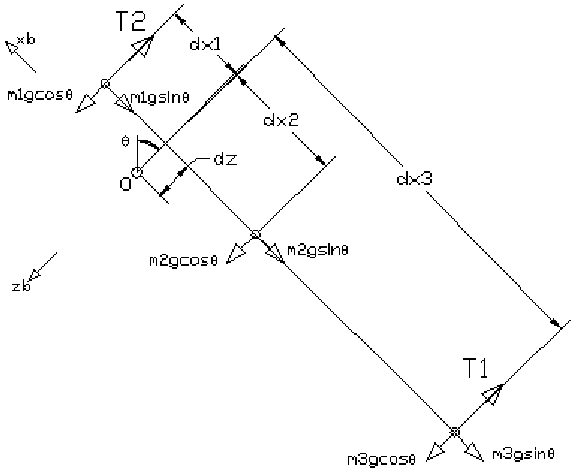


Figure 3: Three Point-Mass Model

Table 1: Model Variables

$T_1 = Pitchfan\ Thrust$
$T_2 = Main\ Engine\ Thrust$
$g = 32.283\ ft\ sec^{-2}$
$\theta = Pitch\ Angle$
$m_1 = Mass\ of\ Main\ Engines$
$m_2 = Mass\ of\ Pitchfan\ Engine$
$m_3 = Mass\ of\ Pitchfan$
$d_z = z_b - Axis\ Distance\ of\ All\ Masses$
$d_{x_1} = x_b - Axis\ Distance\ of\ m_1$
$d_{x_2} = x_b - Axis\ Distance\ of\ m_2$
$d_{x_3} = x_b - Axis\ Distance\ of\ m_3$

Nonlinear Mathematical Model. The mathematical model is developed beginning with the rotational analog of Newton's 2nd law, as derived in any classical physics textbook [2]. Application to the given system yields the following equation of the non-linear model, where I = Moment of Inertia.

$$\ddot{\theta} - \left[\frac{g d_z}{I} \left(m_1 + m_2 + m_3 \right) \right] sin(\theta) \; + \; \left[\frac{g}{I} \left(m_1 d_{x_1} + m_2 d_{x_2} + m_3 d_{x_3} \right) \right] cos(\theta) \\ = \frac{1}{I} \left(T_2 d_{x_1} \; - \; T_1 d_{x_3} \right) \tag{1}$$

Values were determined for the constants in this equation based on the geometry and mass distribution of the test stand. Due to the fact that the center of gravity of the stand lies on the z_b-axis (which passes through the point of rotation) the $cos(\theta)$ term becomes zero. In addition, T_2 (main engine thrust) is set to zero because the pitchfan (T_1) is the primary controlling input for the pitch motion of the system. These substitutions and simplifications result in a nonlinear model of the following form:

$$\ddot{\theta} \; - \; 25 \, sin(\theta) \; = \; -14.4 \, T_1 \tag{2}$$

Linear Mathematical Model. The model is now linearized by using the small angle approximation, $sin(\theta) \cong \theta$. The linear model is given by the following:

$$\ddot{\theta} \; - \; 25 \, \theta \; = \; -14.4 \, T_1 \tag{3}$$

Open-Loop Transfer Function. Taking the Laplace transform of the linear model yields the open-loop transfer function, referred to as the plant. The plant is the representation of the linear model which will be used to develop the control system. It is known that the plant is unstable because it possesses a pole at s=+5. Simulating the open-loop step response (Section 6) verifies this conclusion.

$$P(s) \; = \; \frac{-14.4}{s^2 \; - \; 25} \tag{4}$$

4 Controller Development

At this point, an automatic controller is developed which will artificially stabilize the system. An automatic controller compares the actual plant output with the desired value and produces a control signal which will reduce the difference between the actual and desired (referred to as the error) to zero or a small amount [3]. The goal of this mathematical design process is to determine a controller transfer function (or combination of TFs) which will produce the desired transient and steady-state response.

Series Compensation. The design of the controller begins with the design of a series compensation controller (in series with the plant). The fundamental form of the controller (K_1) is chosen to be that of a basic PID controller with the addition of a (s+5) term in the numerator to cancel the pole at s=-5, as shown in Equation 5.

$$K_1(s) = \frac{-k(s+a)(s+5)}{s} \tag{5}$$

Use of this controller yields a closed-loop transfer function of the following form:

$$TF_1 = \frac{14.4k(s+a)}{s^2+(14.4k-5)s+14.4ka} \tag{6}$$

The purpose of employing a control system is to modify the system such that the desired transient and steady-state response is obtained. Unit step responses for various damping ratios (ζ) are given by Ogata [3]. He states that a damping ratio between 0.4 and 0.8 should be chosen to achieve a desirable transient response. For this system, $\zeta = 0.7$ is chosen for its quick rise time and minimal overshoot. A damping ratio of 0.7 can be achieved by placing the closed-loop poles at -1±j. Values of (k) and (a) are solved for such that this condition is satisfied. The characteristic equation has the form of Equation 7 for pole positions of -1±j.

$$s^2+2s+2 = 0 \tag{7}$$

Comparison with the characteristic equation of the system results in (k) and (a) values as shown below.

$$k \;=\; 0.486 \qquad a \;=\; 0.286 \tag{8}$$

Equation 5 represents the fundamental form of the controller, but is in itself improper. In order to make the system strictly proper, the following term is included in the controller. This term will reduce the effects of high-frequency noise and limit the magnitude of the input to the plant.

$$K_{sp}(s) \;=\; \frac{50^2}{(s+50)^2} \tag{9}$$

The complete series compensation controller is shown in the following figure of the closed-loop control block diagram.

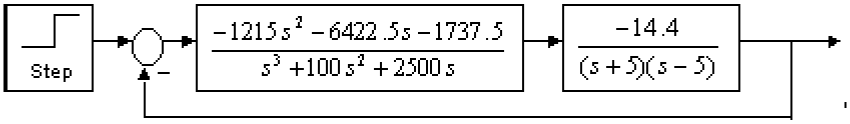


Figure 4: Series Compensation Control Block Diagram

Feed-Forward Compensation. This controller will stabilize to the reference command (i.e., no steady-state error), but a very large overshoot is also present (see details of the simulation in Section 6). This overshoot is due to the derivative control action (see numerator of Equation 6). A feed-forward controller is employed to reduce the overshoot (Equation 10). This controller is chosen based on the output of K_1 at large frequencies.

$$K_2(s) \;=\; \frac{k(s+5)(50)^2}{(s+50)^2} \tag{10}$$

The modified control block diagram (both series and feed-forward compensation and negative feedback) is shown in Figure 5 using the nonlinear model. The linear closed-loop system is obtained by replacing the nonlinear model blocks with the block representing the plant.

The closed-loop transfer function for the new block diagram is given below. The derivative term has been eliminated from the numerator and the overshoot is reduced to a small amount. Section 6 explores this in more detail.

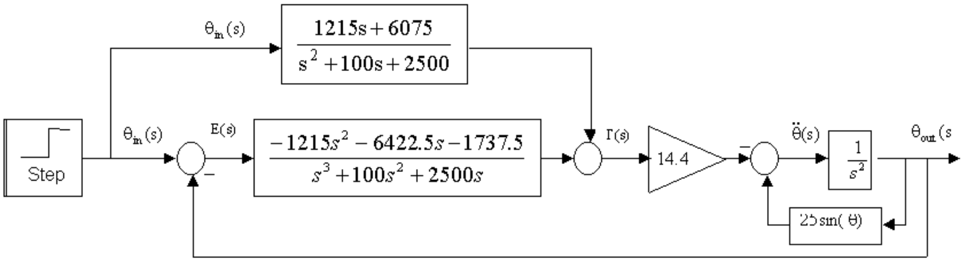


Figure 5: Complete Control Block Diagram

$$TF_2 = \frac{5004}{s^4 + 95s^3 + 2000s^2 + 4996s + 5004} \tag{11}$$

5 MoSART Software Environment

This section describes the MoSART software environment for the tilt-wing test stand. The general structure of the environment is discussed, as well as the functionality and content of the individual environment modules.

A MoSART environment is an interactive application for simulating and visualizing a variety of complex systems. The MoSART team at ASU has developed several system-specific environments [4] [5] [6] [7] [8] [9], using software which operates on any PC-compatible computer running Microsoft Windows '95 or NT. For optimum performance, a fast Pentium processor and 3D-accelerated video card is recommended.

The Tilt-Wing Test Stand MoSART Environment is 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 is discussed below.

5.1 Program Interface Module (PIM)

The Program Interface Module (PIM) provides an interface between a user and the Tilt-Wing Test Stand Environment program. Written in the Microsoft Foundation Classes (MFC) framework of Microsoft Visual C++ and standard Windows '95, the environment provides pull-down menus which permit the user to select models, algorithms, and parameters. The user, for example, can: select/edit a simulation model (e.g. linear/nonlinear), select/edit the input signals to the simulation (e.g. steps, sinusoids, etc.), view/change the simulation parameters, save/load the simulated data, or post-process the data in MATLAB. The active child window contains a block-diagram representation of the system (see Figure 6). The user may edit parameters associated with any of the available components simply by clicking the mouse on the box, or through the menus. Common functions are accessible through the floating/docking tool bar, which has a VCR-style control panel for controlling the simulation.

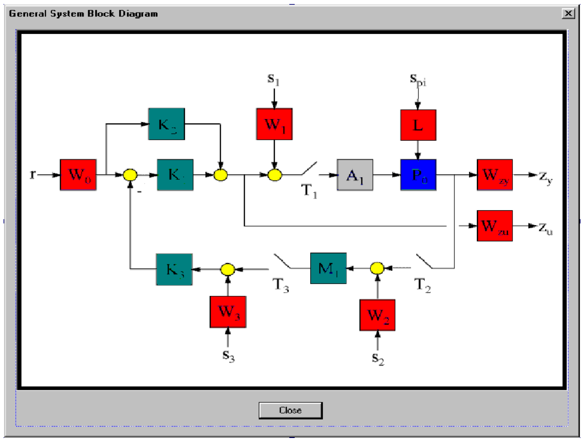


Figure 6: Block-Diagram System Representation

5.2 Simulation Module (SIM)

The simulation engine uses a matrix-algebra C++ class toolset specifically developed for this application. It utilizes an iterative algorithm which numerically solves the ordinary differential equations describing the system and is capable of simulating a system which is based on a very general block diagram structure (Figure 6). The user can specify the use of different integration methods, 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 using the *MATLAB Engine Communication Link*.

5.3 Graphics/Animation Module (GAM)

The main purpose of the Graphics/Animation Module (GAM) is to create/manipulate graphics and animations using data generated by the simulation module. Data, plots, and animations are displayed within child windows. 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, and 3-dimensional animation windows.

Real-Time Graphing Windows. This window shows a continuously updated list of the essential state variables of the simulation. Additional information including the simulated-time/real-time ratio and the number of displayed frames-per-seconds are also displayed.

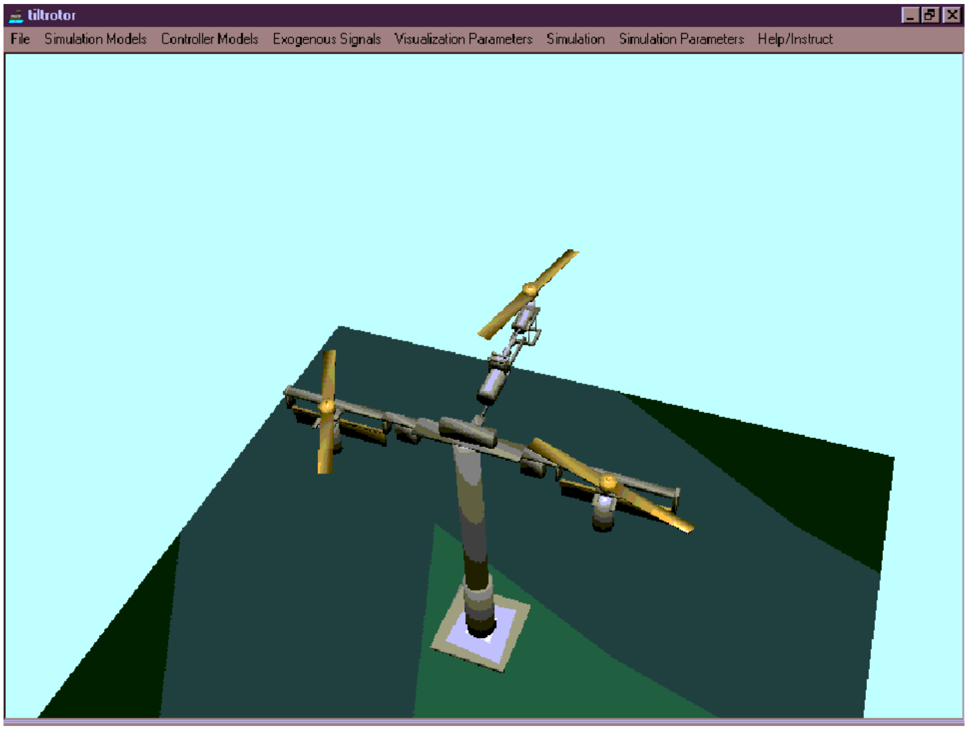


Figure 7: Tilt-Wing Test Stand 3-D Animation Window

3-Dimensional Animation Windows. This window shows an animated real-time 3-dimensional graphic. The object elements are represented by texture-mapped light-shaded polygons. The user may specify a view-point and viewing perspective in real-time. This view is rich in detail and allows the viewer to quickly discern the spatial relationships of the simulated objects. The animation is achieved using Direct-3D [10]. The environment utilizes Direct-3D version 3.0 which is supported on Windows '95 and Windows NT. Direct-3D offers many features, such as advanced rendering options (Gouraud shading and Phong shading [10]), texture mapping, and a standard object-definition file-format. A screen shot of the tilt-wing test stand 3-D animation window is given in Figure 7.

5.4 Help/Instruct Module (HIM)

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.

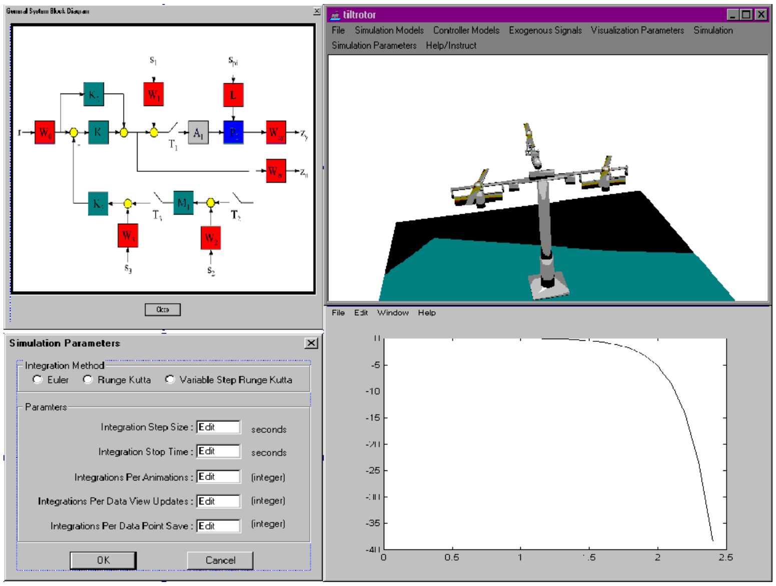
6 Educational Utility of Environment

In this section, the utility of the Tilt-Wing Test Stand MoSART Environment is discussed. Specifically, the usefulness of the environment in the controller design process is demonstrated.

6.1 Open-Loop Response

The step response of the nonlinear and linear models was simulated in order to explore the natural stability of the system. Figure 8 illustrates several aspects of the environment which facilitate the simulation process. The general control block diagram (Figure 6) provides easy modification of any system parameters (in the open-loop case only the plant is used and the feedback paths are not employed). The 3-D animation window provides visualization of the response of the system in real-time. Details of the simulation, including integration method and step size, can be adjusted through dialog boxes such as the one in Figure 8. Finally, the pitch-angle is plotted vs. time to provide a graphical representation of the instability of the open-loop system.

The example in Figure 8 is for the linear model, and Figure 9 is a plot of the open-loop step response of both the linear and nonlinear models together for comparison purposes.



Simulation Parameters

Integration Method

○ Euler ○ Runge Kutta ○ Variable Step Runge Kutta

Paramters

Integration Step Size : Edit seconds
Integration Stop Time : Edit seconds
Integrations Per Animations : Edit (integer)
Integrations Per Data View Updates : Edit (integer)
Integrations Per Data Point Save : Edit (integer)

Figure 8: Open-Loop Linear Step Response

This plot illustrates very well the differences between the linear and nonlinear models. The responses are very similar for small angles, but quickly diverge once the pitch angle becomes large. Also, the oscillatory nature of the nonlinear response is not represented in the linear model. Both models predict an unstable system, which is verified by observation of the actual hardware. The linear model is a good approximation for small angles, but the nonlinear model is certainly better over a large range of pitch angle. The software environment is a useful tool for visualizing the differences between the linear and nonlinear models because one can switch between them and observe the

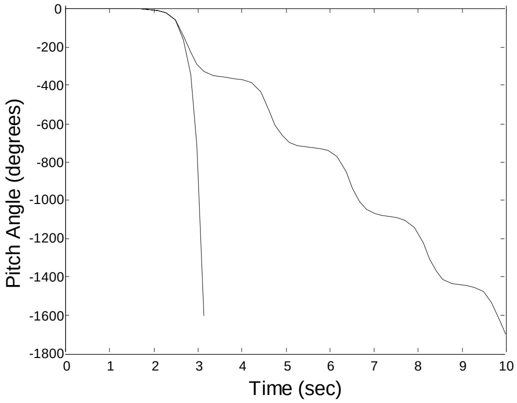


Figure 9: Linear/Nonlinear Open-Loop Step Response

corresponding responses rapidly.

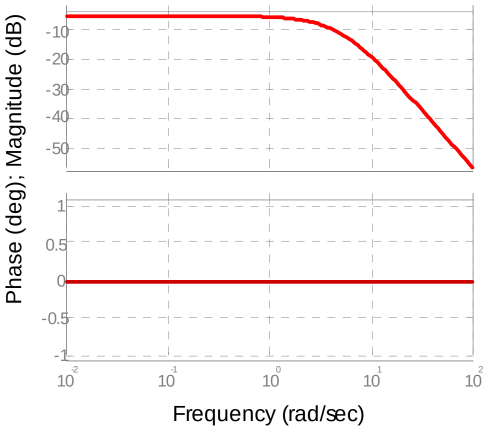


Figure 10: Open-Loop Bode Plots

The frequency response can also be explored through the use of Bode plots. The Bode plots for the plant can be obtained using the software environment through the use of MATLAB's capabilities. Given in Figure 10, the open-loop Bode plots show that the magnitude is $\sim -2\ dB$ for low frequencies and then drops off above $\omega = 5$.

6.2 Controller Design

Since the open-loop analysis has shown the system to be unstable, it is necessary to design a controller to stabilize the system and provide the desired transient and steady-state response. The control system was laid out in detail in Section 4, and in this section the utility of the MoSART software environment in the design process is discussed.

Series Controller. After a series compensator was developed with appropriate (k) and (a) values (see Section 4), the new closed-loop system was simulated for both the linear and nonlinear models. The step response curves were plotted and are shown in Figure 11. This graph shows that the series compensation does indeed stabilize the system to the input (10 degrees in this case), but there is a large overshoot associated with the response. This is a very undesirable effect which can be seen quite clearly through 3-D animation of the test stand.

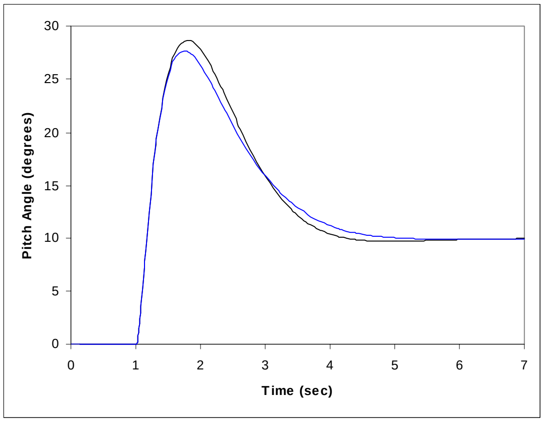


Figure 11: Linear/Nonlinear Series Compensation Closed-Loop Step Response

Series+Feed-Forward Control. As already mentioned in Section 4, the overshoot is caused by the derivative action term of the series compensator. The overshoot is eliminated through the use of a feed-forward compensator, which can easily be incorporated in the general block diagram of the MoSART environment (see Figure 8). A simulation of the system incorporating series and feed-forward control was performed and the overshoot was observed to be reduced to a very favorable amount (see Figure 12).

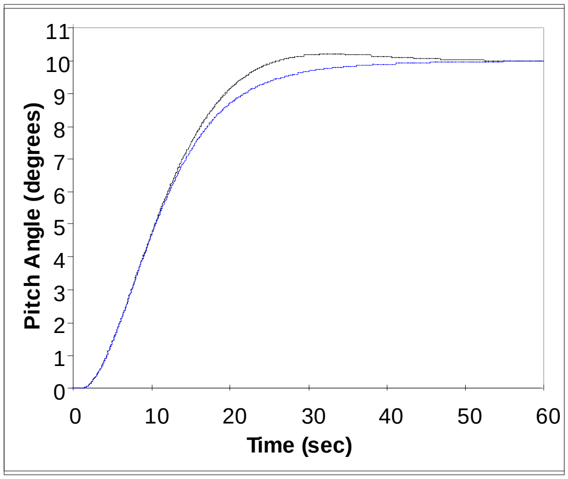


Figure 12: Linear/Nonlinear Series+Feed-Forward Closed-Loop Step Response

The animation provides a good visualization of the improvement in the performance when the feed-forward controller is added to the system. Changes such as this can be implemented in the environment quickly so that the positive and negative qualities of the change can be ascertained. The ability to make such changes and see the results (through both animation and graphing) within seconds makes the MoSART environment a powerful control system design tool.

High vs. Low Gain Controllers. From the previous two examples, the linear and nonlinear step responses were plotted and found to correspond very closely. A graph of the error between the two curves in given in Figure 13.

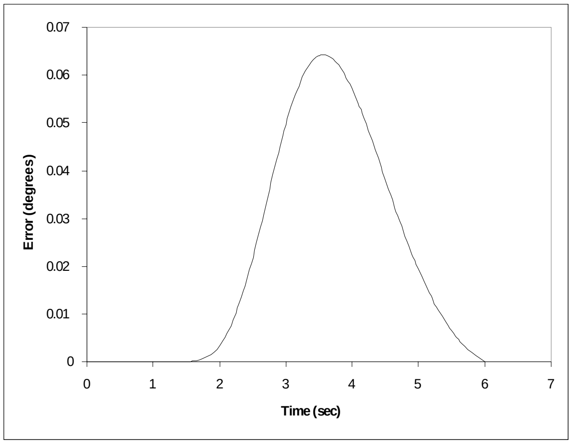


Figure 13: Error Between Linear and Nonlinear Closed-Loop Step Response for Original Design

The maximum error for a 10 degree input is 0.065 degrees, which corresponds to 0.65 percent error in the linear model. This error seems to be too small based on the fact that the approximation $sin(\theta) \approx \theta$ has 1.75 percent error at $\theta = 10\,degrees$. This "masking" of the non-linearity is due to the relatively high gain of the controller. By choosing to place the closed loop poles at a location closer to the origin, say $-0.1 \pm 0.1j$, the response will be slower and the non-linearity will be more apparent. In order to simulate this new lower-bandwidth controller in the MoSART environment, new values of (k) and (a) are calculated to be:

$$k = 0.361 \qquad a = 0.00385 \tag{12}$$

The new control system has the same form as the higher bandwidth version, but with different

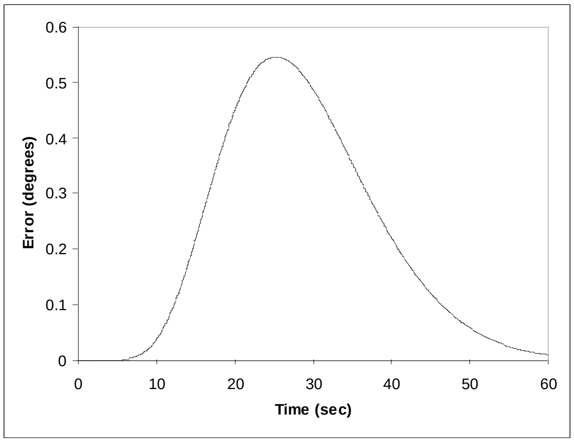


Figure 14: Error Between Linear and Nonlinear Closed-Loop Step Response for New Design

constants. This new system is easily incorporated in the software environment by adjusting the values of the general control block diagram. The error in the resulting linear and nonlinear response curves is plotted in Figure 14.

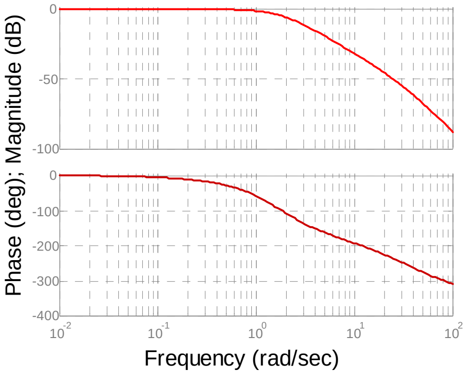


Figure 15: Closed-Loop Bode Plots

The maximum error is 0.55 degrees, which corresponds to 5.5 percent error. A 5 percent error at 10 degrees input is a much more reasonable value based on the $sin(\theta) \cong \theta$ approximation. An examination of the Bode plots of the closed-loop system will enlighten the situation. The software environment can take advantage of MATLAB's capabilities to produce the desired Bode plot (Figure 15). The magnitude vs. frequency plot shows that the magnitude drops off quickly at higher frequencies, but is unity for low frequencies.

This exercise has shown that the simulations appear to be running correctly. The initial controller which was designed has a high relative gain and thus makes the response of the non-linear model appear more linear. The second controller is of a lower gain which reveals the non-linearity which is present in the non-linear model. The first controller is more desirable because it yields a much faster settling time, however practical limitations may limit the gain on the controller which can be used in the actual system.

7 Conclusion and Future Directions

This paper has discussed the development of an interactive Modeling, Simulation, Animation, and Real-Time control (MoSART) software/hardware testbed for a tilt-wing rotorcraft. The development of nonlinear and linear models was presented and a control system was shown which employs series and feed-forward control in a closed-loop system to achieve the desired transient and steady-state response. A tilt-wing teststand MoSART software environment was discussed which allows rapid simulation and 3-D visualization of the system using various models and control system variations. The utility of this software environment in the control system design process was presented with several examples illustrating various aspects of the environment.

The next step in the development of the hardware/software testbed is to link the physical test stand to the software environment so that real-world data can be obtained and analyzed. This can be achieved through the use of a data acquisition board to transmit data from the rate gyros on board the test stand to the computer and to transmit control data from the computer to the engines and ailerons of the test stand.

Of course, for the full control of the tilt-wing test stand, a control system which is based on the 4-DOF system must be designed. The first step is to develop a 4-DOF model in a manner similar to that used to develop the pitch-axis model presented in this paper. The tilt-wing test stand MoSART environment can then be utilized in the control system design process as shown for the pitching motion case.

The hardware/software testbed which is being developed will provide an interactive tool for the development of a control system for the High-Speed Autonomous Rotorcraft Vehicle (HARVee).

This testbed will be invaluable due to its ability to collect real-world data for analysis as well as to provide real-time control of the physical system.

8 Acknowledgements

Dr. Rodriguez's support of me and this research has been simply amazing. His assistance during the controller design process was crucial to the success of this project thus far, and I have learned much from his expertise in this subject. Richard Metzger's technical assistance with the software environment has been invaluable, I can't thank him enough for his time and efforts. Karen Linda and Duane Whitcraft have helped me through many of the trials and tribulations of this project and my gratitude goes out to them also.

References

[1] R. G. Jacquot and F. M. Long, *Introduction to Engineering Systems*, Allyn and Bacon, Inc., 1988.

[2] R. A. Serway, *Principles of Physics*, Saunders College Publishing, 1994.

[3] K. Ogata, *System Dynamics*, 2nd Ed., Prentice Hall, Englewood Cliffs, NJ, 1992.

[4] 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.

[5] 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 International Conference on Simulation in Engineering Education*, Phoenix, AZ, January 12-15, 1997, pp. 71-76.

[6] 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 International Conference on Simulation in Engineering Education*, Phoenix, AZ, January 12-15, 1997, pp. 57-62.

[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 International Conference on Simulation in Engineering Education*, 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 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp. 175-180.

[9] 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.

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