DOT/FAA/AR-03/51

Simulation and Flight Test Assessment of Safety Benefits and Certification Aspects of Advanced Flight Control Systems

August 2003

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

Technical Report Documentation Page

1. Report No. DOT/FAA/AR-03/51	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle SIMULATION AND FLIGHT TEST ASSESSMENT OF SAFETY BENEFITS AND CERTIFICATION ASPECTS OF ADVANCED FLIGHT CONTROL SYSTEMS		5. Report Date August 2003
		6. Performing Organization Code
7. Author(s) James E. Steck, Kamran Rokhsaz, U. J. Pesonen, Sam Bruner*, Noel Duerksen*		8. Performing Organization Report No.
9. Performing Organization Name and Address Wichita State University *Raytheon Aircraft Company 1845 North Fairmont Street Wichita, KS 67260 Wichita, KS 67260-0044		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. 00-C-WSU-00-14
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Office of Aviation Research Washington, DC 20591		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code AIR-120

15. Supplementary Notes

The FAA William J. Hughes Technical Center Technical Monitor was Charles Kilgore.

16. Abstract

An adaptive inverse controller for customized fight control systems for general aviation aircraft is presented. The purpose of the system is to render a general aviation aircraft easier to fly via decoupling its flight control system. Artificial neural networks are used to counteract the errors due to nonlinearities and noise and to adapt to partial control failure, thus allowing the pilot to continue to safely control the aircraft. The system was verified with MATLAB simulations for longitudinal flight. Simulations of changing configurations, payload, and partial control system failures have shown that the controller does rapidly adapt to these changes without a need for pilot response. It has also been demonstrated that the controller code can be generated from the Simulink model that is compatible with existing code in the flight computer on a Raytheon Beech Bonanza F33C fly-by-wire test bed. The longitudinal flight theory was extended to lateral-directional flight and an inverse controller was derived. The potential increased safety and certification issues are discussed.

17. Key Words Adaptive flight control, Software certification, MATLAB simulink, Aircraft safety, General aviation, Artificial neural networks, Inverse control	18. Distribution Statement This document is available to the public through the National Technical Information Service (NTIS) Springfield, Virginia 22161.		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 57	22. Price

Form DOT F 1700.7 (8-72)   Reproduction of completed page authorized

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF ACRONYMS AND SYMBOLS

AGATE	Advanced General Aviation Transport Experiments
ANNs	Artificial Neural Networks
ANSI-C	American National Standards Institute Standard for C code
DARPA	Defense Advanced Research Projects Agency
DGPS	Differential Global Positioning System
DOF	Degree of Freedom
EZ Fly System	Velocity Vector Command with Envelope Protection – Control System
FAR	Federal Aviation Regulations
IFR	Instrument Flight Rules
MIMO	Multi-Input-Multi-Output
NASA	National Aeronautics and Space Administration
PID	Proportional, Integral, Derivative
RTCA/DO-178B	Radio Technical Commission for Aeronautics Aviation Software Document: "Software Considerations in Airborne Systems and Equipment Certification"
SATS	Small Aircraft Transportation System
$C_{L_{\delta_e}}$	Change in lift coefficient with elevator deflection
$C_{D_{\delta_e}}$	Change in drag coefficient with elevator deflection
$C_{m_{\delta_e}}$	Change in pitching-moment coefficient with elevator deflection
C_{m_α}	Change in pitching-moment coefficient with angle of attack
$C_{T_{\delta_T}}$	Change in thrust coefficient with throttle
C_L, C_m, C_D	Lift, pitching moment and drag coefficients
$\bar{c}$, b	Aircraft reference chord length and wing span
$F_{A_x}, F_{A_y}, F_{A_z}$	Aerodynamic forces in forward, side, and vertical directions
I_{xx}, I_{yy}, I_{zz}, I_{xz}	Aircraft mass moment of inertias
L, M, N	Aircraft rolling, pitching, and yawing moments
M	Mass of the aircraft
P, Q, R	Roll, pitch, and yaw aircraft angular velocities
$\bar{q}$	dynamic pressure

S	Aircraft reference wing area
T	Thrust
U, V, W	Forward, side, and vertical aircraft velocities
V_p, or V_{true}	Total aircraft airspeed
V_c, θ_c, γ_c	Commanded airspeed, pitch angle and flight path angle
α, β	Aircraft angle of attack and sideslip
$\delta_a, \delta_e, \delta_r$	Aileron, elevator, and rudder deflections
γ	Aircraft flight path angle
ϕ_T, d_T	angle and moment arm of the thrust force
Ψ, Θ, Φ	Aircraft heading, pitch, and roll angles

A more detailed explanation of these symbols, which are used in the equations of motion in this report, can be found on pages vii-xxviii of "Airplane Flight Dynamics and Automatic Flight Controls," Jan Roskam, DARcorporation, Lawrence, KS, as well as in the government publication "USAF Stability and Control DATCOM," D.E. Hoak and R.D. Finck, Wright-Patterson Air Force Base, Engineering Documents, 1990 M Street N.W., Suite 400, Washington D.C.

EXECUTIVE SUMMARY

An advanced flight control system, which has been demonstrated to compensate for unanticipated failures in military aircraft, was investigated for use in general aviation. The method uses inverse control to decouple the flight controls and modify the handling qualities of the aircraft. The system can render a general aviation aircraft easier to fly by decoupling its flight control system, making the aircraft handling more natural to a nonpilot. Artificial neural networks are used to counteract the modeling errors in the inverse controller, but more importantly, to adapt to control system failures during flight, thus allowing the pilot to continue to safely control the aircraft. Since this system is software-based, the control system is fly-by-wire.

It is difficult for general aviation to incorporate the level of redundancy required in such flight control systems; therefore, the demonstration of the system's capability to handle control system failures is critical to future certification efforts. The system was verified with MATLAB simulations for longitudinal flight. In simulations, the control system was shown to be able to track pilot velocity, pitch angle, and flight path angle commands. Simulations of changing configurations, payload, and partial control system failures have shown that the controller does rapidly adapt to these changes without a need for pilot response. It has been demonstrated that the controller code can be generated from the Simulink model that is compatible with existing code in the flight computer on a Raytheon Beech Bonanza F33C fly-by-wire test bed.

Flight tests have not yet been conducted. The theory was extended to lateral-directional flight, and as a result, an inverse controller was derived. The theory has not yet been verified in simulation. Investigations related to if and how this type of control system could be certified point to two issues. First, MATLAB is used extensively in the development and testing of flight control systems; therefore, the ability to generate reliable code for these types of systems needs to be verified. Second, the algorithms used in the control system are straightforward and deterministic, except for the artificial neural network learning, which depends on the learning data that is presented.

Advances in modern fight control design provide a means to design an operationally simplified control system that allows a general aviation aircraft to be flown safely with a lower level of piloting skills. This can make personal air transport available to a larger group of people who are not aviation enthusiasts. The National Aeronautics and Space Administration (NASA) Small Aircraft Transportation System (SATS) Program aims to provide reliable personal air travel for a wider audience, with the development of a flight control system that provides a reduced pilot workload through decoupled control modes and stability augmentation. The envisioned SATS aircraft will also enable pilots with low experience to operate the aircraft in most weather conditions, will have an emergency autoland capability, and will adapt to changes in aircraft behavior due to unanticipated actuator and sensor failures or structural damage.

Most nonlinear control techniques are based on linearizing the equations of motion and using nonlinear feedback. Brinker, et al. [1] found that for their dynamic inverse controller for aircraft, the longitudinal stability and flying qualities were robust to parameter uncertainties, but the lateral-directional flying qualities were sensitive to uncertainty in stability derivatives. For longitudinal flight control of a missile, McFarland and Calise [2] proposed a method where neural networks and direct adaptive control are used to compensate for unknown nonlinearities, while dynamic nonlinear damping provides robustness to unmodeled dynamics. In a later study [3], the same authors used a similar methodology for a bank-in-turn control of a missile. In both studies, a neural network was used to adaptively cancel linearization errors through on-line training. Artificial neural networks (ANNs) are capable of approximating continuous nonlinear functions with very little memory and computational time required, but due to the empirical character of these methods, it has previously been difficult to guarantee sufficient reliability for such a high-risk application as flight control. Using neural networks for nonlinear inverse control of the XV-15 tilt-rotor, Rysdyk, et al. [4] showed theoretically as well as by simulation that the ANN weights remain bounded during on-line training. This is an important step towards certifying aircraft control systems that use ANNs. The guaranteed boundedness of signals and tracking error is discussed in detail in Rysdyk's PhD thesis [5]. Rysdyk, et al. [6] also obtained consistent response characteristics throughout the operating envelope of a tilt-rotor aircraft. Further, in a recent study, [7] Rysdyk, et al. applied ANNs to a Total Energy Control System for longitudinal flight control.

Soloway and Haley, at NASA, studied reconfigurable aircraft control [8]. Their model is capable of real-time control law reconfiguration, model adaptation, and identification of failures in model effectiveness. A full 6 degree-of-freedom (DOF) model of a conceptual commercial transport aircraft was used to simulate the elevator freezing in flight, and the algorithm reconfigured itself to use symmetric aileron deflections to control pitch rate, thereby stabilizing the aircraft. Again, an artificial neural network was used to learn the changed dynamics of the aircraft with frozen elevators. Kim and Calise [9] developed a direct adaptive tracking control using neural networks to represent the nonlinear inverse transformation needed for feedback linearization. It was shown that the adaptation algorithm ensured uniform boundedness of all signals in the loop and that the weights of the on-line neural network converged to constant values. In a study concerning serial-link robot arm control, Lewis, et al. [10] found that on-line training of the neural network together with a signal that adds robustness guarantee tracking as well as bounded ANN weights. Standard back propagation, however, yielded unbounded ANN

weights if the network was not able to reconstruct a required control function or there were unknown disturbances in the system dynamics. Alternate back-propagation schemes corrected this problem.

In flight tests, adaptive control designs have been demonstrated for a mid-size transport (NASA) and in the X-36 (Air Force) with unanticipated control failures. The NASA demonstration included a piloted simulation using only propulsion for backup flight control. Applicability of an emergency flight control system greatly increases if it can provide desirable responses over a wide range of unanticipated failures via adaptive control.

An operationally simplified flight control system already exists (modifications funded by NASA Advanced General Aviation Transport Experiments on a Raytheon Beech Bonanza F33C fly-by-wire test bed). This system was designed to follow pilot input as follows: Longitudinal stick position commands pitch angle or vertical flight path angle, where centered stick commands level flight. Lateral stick position commands bank angle and centered stick commands constant heading. A speed command lever commands airspeed, where stall speed plus 5 kts and never exceed flight speed minus 10 knots is allowed. Plus or minus 7 degrees was allowed in vertical flight path angle, and lateral control was limited to plus or minus 60 degrees. Eventually, the goal is to include an automatic turn coordination, a yaw damper, an angle-of-attack limiter, bank angle limiter, an automatic overspeed limiter, and a load factor limiter. The system could also automatically lock to a Differential Global Positioning System (DGPS) approach and landing with no pilot action. The aircraft should hold straight and level if the pilot lets go of the stick, and if the pilot maneuvers into an approach capture zone and lets go of the stick, the airplane will execute the approach and land with no pilot input. The pilot can re-establish direct control of the airplane by moving the stick from the centered position.

In the current project, an adaptive nonlinear inverse controller was designed for the Raytheon Beech Bonanza F33C single-engine general aviation aircraft. The new controller will be added to the existing system on the Raytheon Beech Bonanza F33C fly-by-wire test bed, which will then be used for test flights. The longitudinal flight controller was designed in the MATLAB Simulink^{TM} environment with the ANNs developed, using the MATLAB Neural Network Toolbox^{TM}. Nonlinear inverse control is used and ANNs are trained on-line to counteract modeling error. The ANNs are also used to adapt to changing flying characteristics due to unanticipated control failures, icing, or physical damage to the airplane. Operationally simplified flight controls will allow the pilot to focus his/her full attention to an emergency while the airplane maintains its current path, thus reducing the cognitive effort required to operate the airplane safely. This type of an adaptive control system could eliminate accidents that are related to loss of control, such as stall, spin, disorientation, overspeeding the airplane, or over-stressing the airplane. It is also a future goal to bring all nonthunderstorm and nonicing Instrument Flight Rules conditions within the capability of all pilots and to have autoland capabilities that are Instrument Landing System—Category IIIb equivalent with no pilot training requirement. The types of accidents that can be reduced are controlled flight into terrain and unsuccessful resolution of an emergency due to high pilot workload. Flight control system failures might lead to different types of accidents, which must be reduced to acceptable levels by proper analysis and certification of the new flight control system. Reliability and safety of such systems specifically require dependable integration of software and hardware. Other benefits of this flight control

system are reduced initial pilot training requirements, reduced proficiency requirements, easier access to personal air transportation, improved perception of aviation safety, and consistent handling qualities across most of the flight envelope and between different aircraft.

2. PLAN OF WORK.

The three planned tasks for this research effort are detailed below.

2.1 TASK 1: MODELING AND SIMULATION.

1.1 **Develop a nonlinear aircraft model of the Raytheon Beech Bonanza F33C**.

    1.1.1 **Gather data to build the model.**
Proprietary geometry, force, and moment data were obtained from Raytheon Aircraft for a v-tail Bonanza in the form of drawings and plots. The data were adjusted to account for the straight tail on the F33C flight test Bonanza used in the project. Linear lift, pitching moment, drag polar, linear rolling, yawing moment models, and a linear side force model were derived. The data used in the MATLAB Simulink model is discussed in section 4.

    1.1.2 **Develop the longitudinal nonlinear aircraft model.**
A longitudinal aircraft model was developed in Simulink using the 6-DOF aerospace equations of motion block, which inputs the forces and moments and calculates the state variables of the aircraft motion. Control surface deflections feed into the force and moment models, which then feed the equations of motion block. Details of the model are presented in section 4.

    1.1.3 **Extract the approximate model for use in the model inversion control method.**
The equations of motion with the force and moment models were inverted to solve for the control deflections to give a required commanded flight variable acceleration. This is a straightforward, yet ingenious, algebraic manipulation of these equations, which is detailed in section 7.

    1.1.4 **Extend the model to lateral-directional characteristics.**
The force, moment, and 6-DOF equations of motion Simulink model have been extended to include longitudinal and lateral directional motion and the inversion of the equations, which has been formulated in section 8.

1.2 **Incorporate the nonlinear aircraft control architecture into the existing Raytheon Bonanza "velocity vector command with envelope protection" control system (EZ-fly system)**

    1.2.1 **Verify the model for multi-input-multi-output control augmentation.**
A system is multi-input-multi-output (MIMO) if it has more than one independent input and output. The longitudinal aircraft model is MIMO, since thrust and elevator are control inputs, and the outputs are the forward velocity pitch angle and flight path angle. Both the longitudinal and full aircraft models were tested by trimming the aircraft in steady-state rectilinear flight and then applying step inputs to the multiple-control surface inputs to verify correct response of the aircraft. Standard output feedback control was implemented with inverse control but without neural networks. Tracking of commanded inputs was demonstrated

when there was no modeling error, that is, when the inverse controller was an exact inverse of the aircraft.

1.2.2 ***Implement nonlinear adaptive control by adding the neural network to compensate for modeling error due to nonlinearities and failure modes.***
Various neural network architectures and methods of incorporation into the control were developed and tested, with the final versions tested and reported in section 4. It is amply demonstrated in the results of this section that the neural networks are more than adequate at compensating for modeling errors and failure modes.

1.2.3 ***Demonstrate that the neural network in combination with approximate inversion control successfully inverts the aircraft model, allowing testing of various pilot control schemes.***
This item was actually performed prior to task 1.2.2 as a demonstration of tracking pilot inputs with no modeling error, which was done as part of task 1.2.1. More importantly, pilot input tracking was demonstrated in the presence of multiple modeling errors and failures, demonstrating the ability of the neural networks combined with inverse control to compensate for large changes and failures in the aircraft during flight. See section 4.

1.2.4 ***Address issues of windup of network weights, including Pseudocontrol Hedging***
The antiwindup method that was investigated involved turning off the training of a neural network error estimator if the control associated with that network has reached its limits and saturated. This is discussed in section 4. Pseudocontrol hedging has not been investigated in this study but will be incorporated in follow-on work.

1.2.5 ***Implement velocity vector command control scheme for longitudinal flight***
Velocity and flight path angle tracking are incorporated in the Simulink control system and tracking of both are shown in the results of section 4.

1.2.6 ***Investigate alternate neural network control architectures to compensate for actuator limits and failure modes.***
Various neural network architectures and methods of incorporation into the control were developed and tested, with the final versions tested and reported in section 4. Somewhat radical departures from the neural network schemes of Calise, et al. were reviewed and not implemented since they involved neural network in the main control path, rather than being active only in the event of a failure. It was felt that these methods were entirely outside the bounds of what could be possibly certifiable, but they may be investigated in follow on studies.

1.2.7 ***Extend control scheme to lateral-directional characteristics***
The aircraft Simulink model and the inverse control scheme have been extended to lateral-directional flight. Details are reported in section 8.

1.3 ***Demonstration phase using the combination of the model and the controller.***

1.3.1 ***Demonstrate velocity vector command with nonlinear adaptive control architecture.***
Velocity and flight path angle tracking are incorporated in the Simulink control system and tracking of both, for longitudinal flight, is shown in the results of

section 4. In addition, aircraft pitch angle can be tracked instead of flight path angle, as shown in the results of this section.

1.3.2 ***Demonstrate adaptive control effectiveness while changing configuration and changing payloads.***

The adaptive inverse controller with the ANN are able to compensate for (1) a 40% change of static margin (C_{m_α}), (2) a 25% loss of throttle-to-thrust gearing ($C_{T_{\delta_T}}$), and (3) a 50% loss of elevator effectiveness ($C_{m_{\delta_e}}$ and $C_{L_{\delta_e}}$) and still track commanded velocity and flight path angle.

1.3.3 ***Demonstrate adaptive response of control system to maintain acceptable handling qualities and safety during unanticipated failure modes (fault tolerance), for example:***

- ***Sensor failure with no backup sensor***
- ***Actuator failure with no backup actuator***
- ***Sensor/actuator noise***

The adaptive inverse controller with the ANN are able to compensate for actuator failures, as shown in the simulation results of section 4, where a 25% loss of throttle-to-thrust gearing ($C_{T_{\delta_T}}$) and a 50% loss of elevator effectiveness ($C_{m_{\delta_e}}$ and $C_{L_{\delta_e}}$) and still track commanded velocity and flight path angle. Both of these could be the result of partial actuator failure. Turbulence and sensor failures have not been investigated. Implementing the inverse controller and neural networks in Simulink took much longer than originally anticipated.

1.4 ***Assess flight test safety issues in conjunction with Task 3.***

This was not accomplished since flight testing the advanced flight controller remains to be done.

2.2 TASK 2: HANDLING QUALITIES AND CERTIFICATION.

2.1 ***Survey of existing military developments and standards on advanced flight control systems with emphasis on nonlinear adaptive flight control augmentation.***

A report of the work performed by Dr. Rolf Rysdyk in the first year of this project gives a survey of existing advanced flight control concepts as well as a thorough description of the theoretical background of the adaptive inverse controller used in this project. A summary of this report is given in section 3.

2.2 ***Investigate the impact of sensor/actuator failure levels on handling qualities and safety (simulation) when there is no redundant secondary sensor or actuator.***

Actuator failures were investigated in section 4. Sensor failure has not been looked at.

2.3 ***Investigate the system performance in the presence of external disturbances, including both wind-shear and atmospheric turbulence (simulation) from the viewpoint of safety.***

This has not been accomplished.

2.4 ***Assess software and hardware integration.***

2.4.1 ***Identify algorithmic accuracy, verifiability, and deterministic issues, in addition to any other issues that may arise, with respect to RTCA/DO-178B and possible certification***

The MATLAB Simulink system development environment was used to develop and test the adaptive inverse controller. The Real-Time Workshop toolbox has been used to dump embedded system C code directly from the Simulink control system model. Work to compile this code as a subroutine embedded in the existing Raytheon Beech Bonanza F33C flight computer EZ-fly software is not complete but is a major effort currently. The advantage of dumping code directly from the development and simulation environment is twofold. First, the code generated is as algorithmically correct and free of coding errors as with the Simulink model, which has been extensively developed and tested. Second, MATLAB was used in many certification projects, although it has not been qualified as a development tool. Typically its output was verified, or it was used with a qualifiable tool that accepts Simulink models. This is addressed in section 6.

2.4.2 ***Identify the critical issues for implementation of the software in Task 3.***

See task 2.4.1 above.

2.4.3 ***Investigate how nonlinear adaptive control affects the required level of redundancy in design and safety features of the flight control system.***

Section 4.5 demonstrates that the present scheme can compensate for partial system failure. This is the first step towards evaluating the necessary level of redundancy in the flight control system.

2.3 TASK 3: FLIGHT TEST DEMONSTRATION.

This task will be performed under supervision of Dr. Noel Duerksen of Raytheon Aircraft Company. Dr. Duerksen is the program lead and test pilot for the Raytheon AGATE/SATS program. The Raytheon Beech Bonanza F33C fly-by-wire test bed will be used for flight test demonstration of the example scenario. Furthermore, it will allow for implementation issues specific to advanced flight control augmentation to be highlighted.

3.1 ***Develop a flight test scenario in conjunction with Task 1.***

The longitudinal velocity and flight path angle tracking inverse controller was identified as the first system to be tested on the Raytheon Beech Bonanza F33C.

3.2 ***Port the controller architecture to the American National Standard Institute standard for C code (ANSI-C) format as used on the Raytheon Beech Bonanza F33C.***

See section 2.4.1 for the discussion on the MATLAB Simulink system development environment.

3.3 ***Implement the controller onto onboard computers in Raytheon Beech Bonanza F33C.***

Raytheon aircraft has brought a second flight computer test bench back on-line for testing the combined EZ-fly and adaptive control system software. ***Assist in design and execution of flight tests.***

Flight testing has not yet occurred as of the end of this project. Porting the code from MATLAB Simulink to C has been worked out, but reworking it to execute on the current

flight computer has become a much more time consuming task than originally envisioned. The Raytheon Beech Bonanza F33C is currently stripped down for inspection and modification work of the onboard computer systems, instrumentation, and flight control hardware. Current plans are to flight test the longitudinal portion of the EZ-fly system in the spring of 2003.

3.4   *Analyze and demonstrate selected flight tests.*
Not accomplished.

3. SURVEY OF EXISTING ADVANCED FLIGHT CONTROL CONCEPTS.

A report [11] of the work performed by Dr. Rysdyk in the first year of this project gives a survey of existing advanced flight control concepts as well as a thorough description of the theoretical background of the adaptive inverse controller used in this project. A summary of this report is given below.

Modern flight control technology, including fly-by-wire, has been used in commercial aviation for over a dozen years (Airbus A320, A330, A340, and Boeing 777). Reliability is expressed as the probability of not being able to continue safe flight and landing due to failures must be less than 10^{-9} per flight hour or not more than 1 in 1 billion and a decreasing trend of probability of occurrence of failures with increasingly severe effects. There can be failures with much higher probability as long as their effect is not catastrophic, and a decreasing trend exists of probability of occurrence of failures with increasingly severe effects. This reliability was established through high-integrity design including redundancy of various independent parts of the control system, and validated through a certification process requiring careful documentation of the design process and extensive demonstration including failure scenarios. General aviation can benefit from modern flight control technology, by providing a huge reduction of flight accidents commonly referred to as pilot error.

The nonlinear adaptive control methods investigated in this project provide an alternate method to achieving a required level of safety by adapting the flight control system in flight to accommodate failures of different subsystems. The other objective of this research was to work toward a low-cost flight control systems design for general aviation aircraft with enhanced safety features that reduces the impact of pilot errors and low piloting skills on accident probabilities, as well as their severity. In fact, failure tolerance goes beyond redundancy by providing a flexible means to deal with unforeseen failures, thereby improving the chance of a safe landing. Nonlinear adaptive control provides the following important benefits:

- Consistency in response without the need for expensive gain scheduled design.

- The ability to take advantage of inherent control redundancy without a priori knowledge of the failure.

- A required guarantee of boundedness with traditional analytical tools. This establishes finite bounds on the response of the aircraft with the control system in place. In other words, the control system does not drive the aircraft to a large response, even active over a long period of time.

The disadvantages are:

- The supporting theories are complex.

- Hardware, software, and its integration require investigation.

- The adaptive nature requires careful application.

- It may be difficult to prove determinism of the algorithms, to prove their stability and their performance robustness under all operating conditions.

The Defense Advanced Research Projects Agency (DARPA) has recognized the need for advanced flight control systems with an initiative called software-enabled control (SEC). This program emphasizes:

- A full exploitation of run-time information and active-model prediction for better disturbance rejection and precision control of high-performance maneuvers.

- On-line control adaptation for transient and permanent failures and damage.

- Coordinated control of integrated subsystems.

- Open, modifiable software control implementation.

Though the SEC initiative is an advanced research and development program, it was evident that there will be significant benefits for civilian applications of modern flight control technology. Therefore, the results of this DARPA effort should be of great interest to the Federal Aviation Administration.

4. ADAPTIVE INVERSE CONTROL DESIGN.

4.1 THEORY FOR THE 1-DOF MODEL.

Initial tests were conducted with 1-DOF system in MATLAB. An inverse controller was designed and tested for effects of actuator failure, system delay, and noise. Partial control failure was modeled with a gain that reduced control effectiveness, mechanical limits on control surface motion were modeled with a saturation switch, and the delay in signal processing was included by adding a delay of one time step to the measured states. The 1-DOF model was based on the following equations.

A nonlinear equation of motion

$$m\ddot{x} + c\dot{x} + k \cdot \sin(x) = \delta(t) \tag{1}$$

The linear version of the above equation (i.e., an estimate)

$$m\ddot{x} + c\dot{x} + kx = \hat{\delta}(t) \tag{2}$$

The following values for the system constants were used: $m = 1$, $c = 0.2$, and $k = 2.0$, the combination of which ensures an underdamped system. The 1-DOF model is shown in figure 1. The Linear Inverse block solves the equation

$$\delta_{corr} = m\ddot{x}_{err} + c\dot{x} + kx \tag{3}$$

The Analytical Inverse block solves the following equation.

$$\delta = m\ddot{x}_c + c\dot{x}_c + kx_c \tag{4}$$

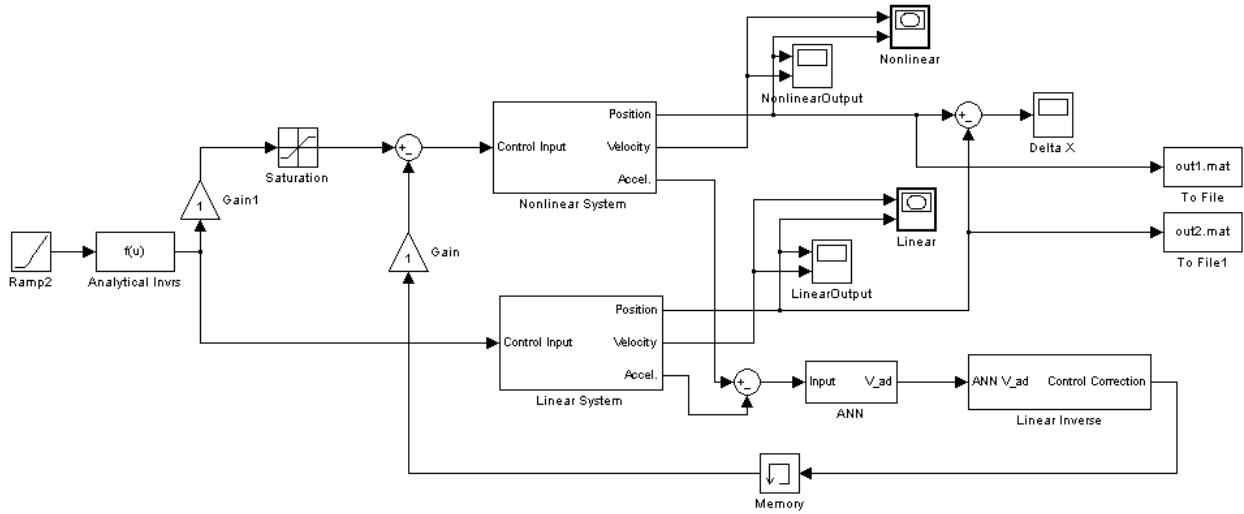
FIGURE 1. THE 1-DOF MODEL

4.2 RESULTS FOR THE 1-DOF MODEL.

The 1-DOF system was tested with a sinusoidal input. Using a feedback loop without an ANN corrects a 50% control failure enough to bring the response up to 75% of the desired amplitude. Eliminating the system delay allows for an almost perfect response, even with 50% control failure. Partial compensation for 50% actuator saturation can be achieved with feedback. Experiments showed that the system was also robust in the presence of signal noise. Figure 2 shows results for control failure and control clipping tests. Figures 2a and 2b show 50% control failure first without correction and then with feedback and eliminated delay. Figures 2c and 2d show 50% control clipping first without correction and then with feedback and eliminated delay. For these results, the system delay was simply removed from the system without the neural network being active, but the goal in designing this system was to eliminate the delay with the adaptive neural network.
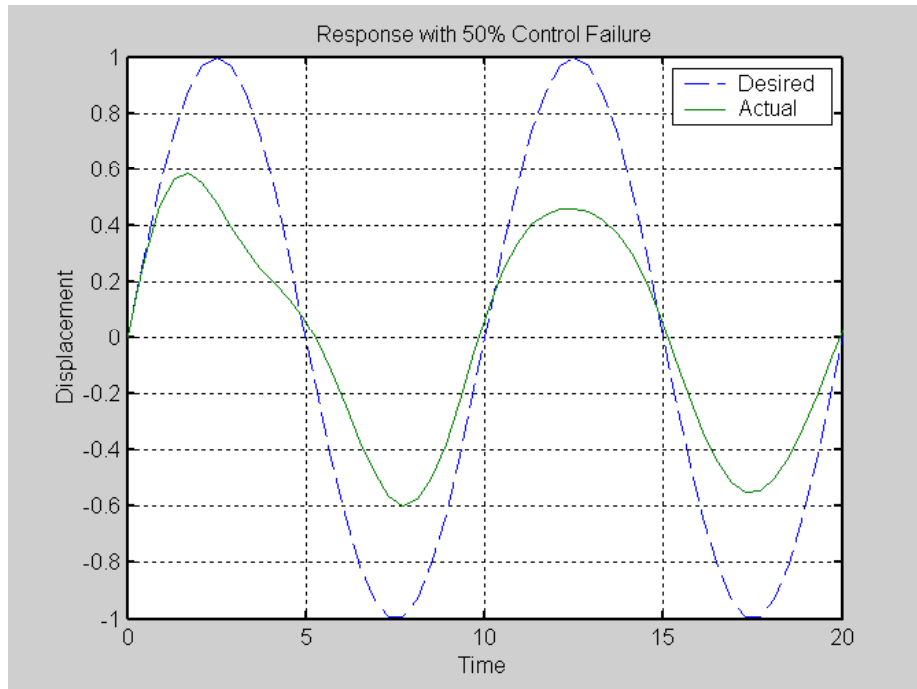
FIGURE 2a. CONTROL FAILURE—RESULTS FOR 1-DOF MODEL WITH
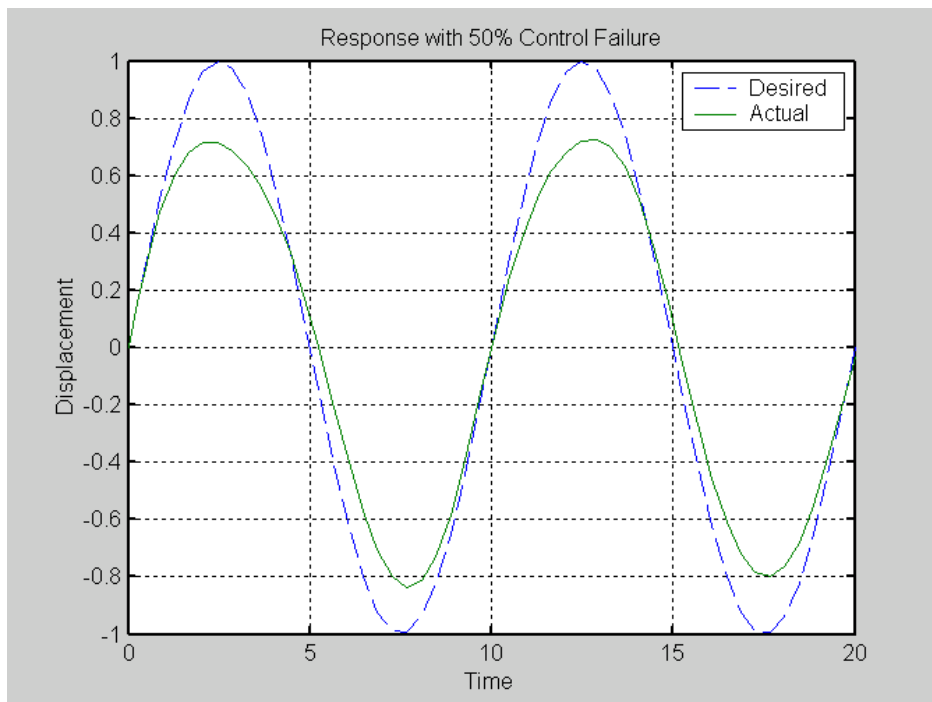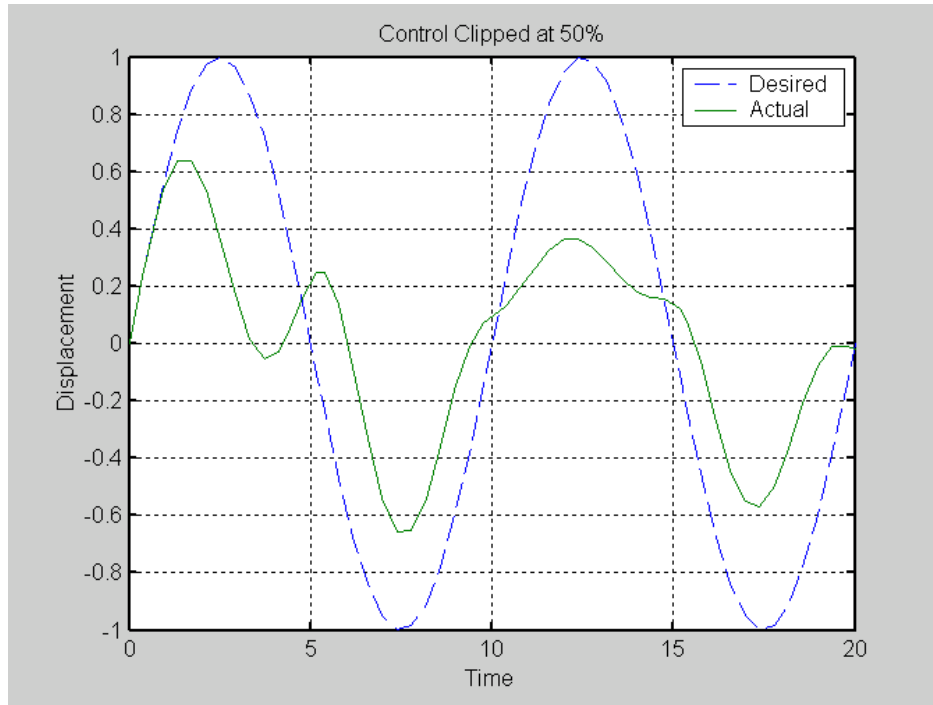SINUSOIDAL INPUT

FIGURE 2b. CONTROL FAILURE WITH CORRECTION

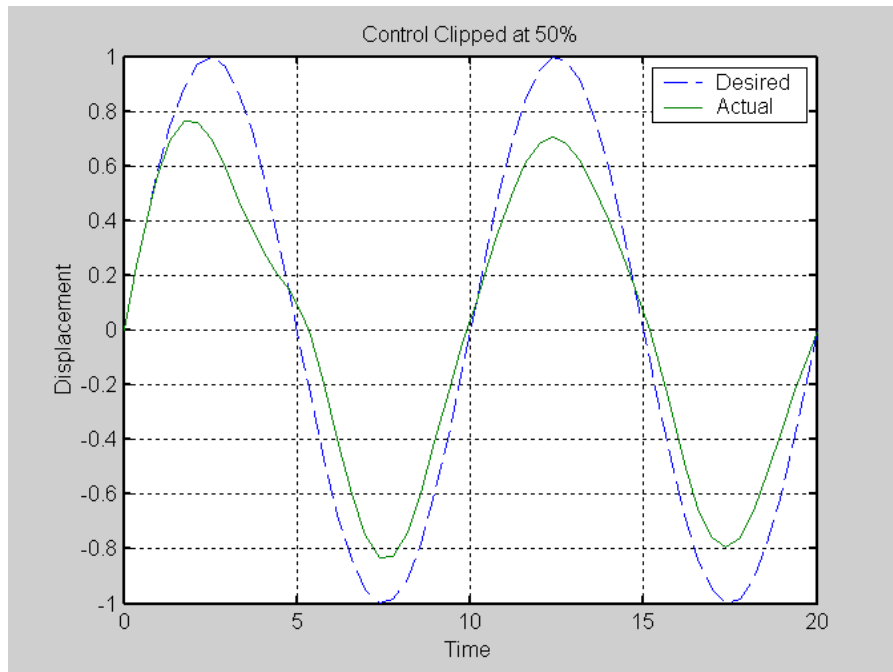FIGURE 2c. CONTROL SATURATION—RESULTS FOR 1-DOF SYSTEM WITH
SINUSOIDAL INPUT

FIGURE 2d. CONTROL SATURATION WITH CORRECTION

The longitudinal aircraft model was based on the following nonlinear equations in the body-fixed x- and z-axis directions and the pitching-moment equation.

$$\dot{U} + g\sin\theta = -WQ + \frac{1}{m}F_{A_x} + \frac{1}{m}T\cos\phi_T + \frac{1}{m}F_{x_{\delta e}}\delta_e \tag{5}$$

$$\dot{W} - g\cos\theta = UQ + \frac{1}{m}F_{A_z} - \frac{1}{m}T\sin\phi_T + \frac{1}{m}F_{z_{\delta e}}\delta_e \tag{6}$$

$$\ddot{\theta} = \frac{1}{I_{yy}}M_A - \frac{1}{I_{yy}}Td_T + \frac{1}{I_{yy}}M_{\delta_e}\delta_e \tag{7}$$

The above equations are consistent with Roskam's [12] notation. The complete MATLAB Simulink model, including the aircraft model, the inverse controller, and the ANNs, is shown in figure 3. The aircraft block uses the Simulink Aerospace 6-DOF Block with a nonlinear drag polar and linear lift and pitching-moment models. The lateral forces and moments are set to zero.

The inverse flight controller was derived from equations 21 and 22, using the definition of true airspeed $V_p^2 = U^2 + W^2$ and assuming $C_{D_{\delta_e}} \approx 0$. Details of this derivation are given in section 7.

$$T_c = \frac{1}{\cos(\alpha + \phi_T)}\left(m\dot{V}_c + mg\sin\gamma + \bar{q}S(C_D)_{\delta_e=0}\right) \tag{8}$$

$$\delta_{e_c} = \frac{C_m(\delta_e)}{C_{m_{\delta_e}}} = \frac{I_{yy}}{\bar{q}S\bar{c}C_{m_{\delta_e}}}(\ddot{\gamma}_c \, or \, \ddot{\theta}_c) - \frac{(C_m)_{\delta_e=0}}{C_{m_{\delta_e}}}$$
$$+ \frac{d_T}{\bar{q}S\bar{c}\cos(\alpha + \phi_T)C_{m_{\delta_e}}}\left(m\dot{V}_c + mg\sin\gamma + \bar{q}S(C_D)_{\delta_e=0}\right) \tag{9}$$

where

$$(C_D)_{\delta_e=0} = C_{D_0} + C_{D_K} \cdot (C_{L_0} + C_{L_\alpha}\alpha - C_{L_1})^2 \tag{10}$$

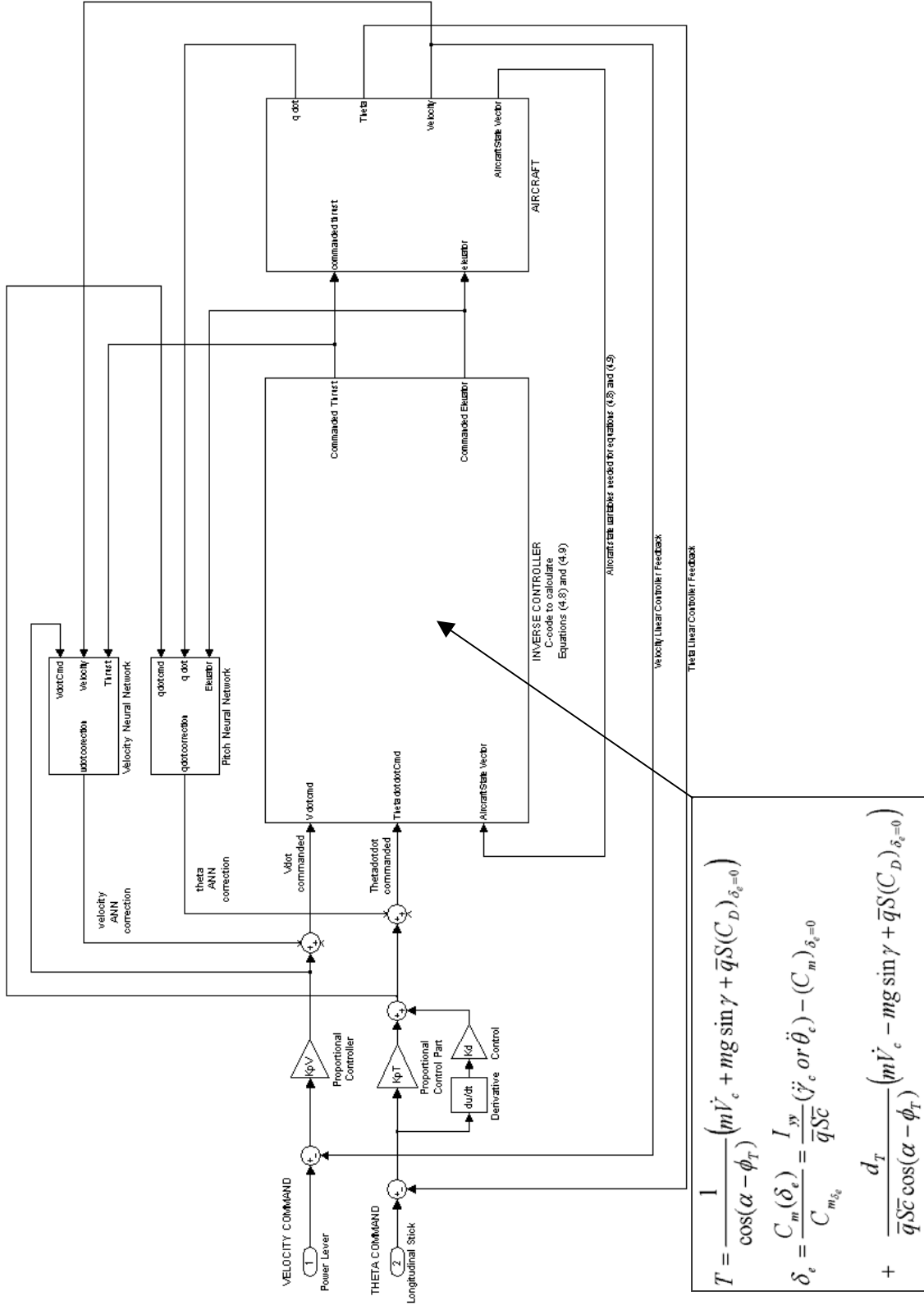$$(C_M)_{\delta_e=0} = C_{M_0} + C_{M_\alpha}\alpha + C_{M_q}\hat{q} \tag{11}$$

FIGURE 3. LONGITUDINAL CONTROLLER BLOCK DIAGRAM

$$T = \frac{1}{\cos(\alpha - \phi_T)}\left(m\dot{V}_c + mg\sin\gamma + \bar{q}S(C_D)_{\delta_e=0}\right)$$

$$\delta_e = \frac{C_m(\delta_e)}{C_{m_{\delta_e}}} = \frac{I_{yy}}{\bar{q}S\bar{c}}(\ddot{\gamma}_c \ or \ \ddot{\theta}_c) - (C_m)_{\delta_e=0}$$

$$+ \frac{d_T}{\bar{q}S\bar{c}\cos(\alpha - \phi_T)}\left(m\dot{V}_c - mg\sin\gamma + \bar{q}S(C_D)_{\delta_e=0}\right)$$

The command variables are the rate of change of flight speed $\dot{V}_c$ and the pitch acceleration $\ddot{\theta}_c$ or flight path angle acceleration $\ddot{\gamma}_c$. These values are computed with separate linear controllers based on the pilot input. Longitudinally, the stick position corresponds to the desired flight path angle, which translates to vertical speed. The throttle position corresponds to the desired flight speed. A linear feedback proportional, integral, derivative (PID) controller was used to track commanded θ_c (or γ_c) yielding $\ddot{\theta}_c$ (or $\ddot{\gamma}_c$). A linear feedback proportional controller is used to track commanded V_c yielding $\dot{V}_c$.

When the inverse controller is added to the aircraft and if there is no modeling error, the combination becomes (1) for the theta or gamma tracking, a simple double integrator block ($1/s^2$ in the Laplace Domain) that takes a commanded angular acceleration and outputs the resulting aircraft angular displacement or (2) for the velocity tracking, a simple integrator block ($1/s$ in the Laplace Domain) that takes a commanded velocity rate (acceleration) and outputs the resulting aircraft velocity. A PID controller was used in the theta or gamma tracking linear feedback loop. Theta tracking is straightforward because the theta acceleration is simply commanded in the inverse controller of equation 4-9. As indicated in this equation, to track gamma, theta acceleration was simply replaced by gamma acceleration. Implicit in this was the assumption that angle-of-attack acceleration is small since theta gamma and alpha are linearly related for longitudinal flight. During the gentle maneuvers of the flight control system, the assumption has proven to be valid based on plots of theta, gamma, and alpha accelerations. Also, the experience Noel Duerksen has had with the EZ-fly system on the Raytheon Beech Bonanza F33C is that gamma tracks theta very closely. The parameters of the PID controller can be chosen to provide any second-order system response characteristics desired, i.e., rise time overshoot damping ratio etc., can be specified. But since the elevator had deflection limits and its actuator had its own response time, there is a physical limit to how fast a rise time can actually be achieved, and how much the faster short period of the aircraft can be damped. The theta or gamma response PID controller parameters were chosen to give a damping ratio of 0.9 and a natural frequency of 0.25 giving a rise time of 5 seconds. A proportional controller was used in the velocity feedback loop since this had a first-order system response. The constant of proportionality can be chosen to give any desired time constant for the aircraft response. Again, since the throttle has limits, there is a physical limit to how fast the aircraft can speed up or slow down. The time constant for the velocity response was chosen to be 5.0 seconds. In retrospect, following discussions with members of the project team, the velocity response chosen above was too fast and should be changed to a time constant of something around 10 seconds, and the pitch angle response was too slow and should be chosen to be on the order of 1-2 seconds rise time.

A note of caution should be inserted here related to when this work is extended to include turbulence, wind, and wind shear. The various terms in the inverse controller must be carefully examined as to whether they are inertial terms or air mass velocity terms. For example, alpha and airspeed are air mass-referenced, whereas theta and commanded acceleration (velocity change) are inertial-referenced.

The ANNs that compensate for nonlinearities and adapt the control system in the partial loss or failure of actuators, or during other changes in flight characteristics, were developed to work together with the inverse controller. For simulation purposes, the aircraft model and the

controller are initialized by mathematically trimming the aircraft in level flight in a MATLAB routine.

4.4 ARTIFICIAL NEURAL NETWORKS FOR FLIGHT CONTROLLER ERROR COMPENSATION.

In the longitudinal flight controller, two ANNs are used to compensate for modeling errors, partial actuator failure, and other changes in flight characteristics. There are separate networks for pitch angle control and for flight velocity control. The ANN Simulink Block containing the velocity ANN shown in figure 4. At every time step, the ANN Simulink Block receives the following information: the current acceleration command from the PD-controller, $\dot{V}_c(t)$, the output of the neural network itself one time step ago (the error compensation signal, e(t-1)), and the actual acceleration of the aircraft one time step ago, $\dot{V}(t$-1). The need for the derivative of airspeed needed in figure 4 may be a problem in the event of turbulence or sensor noise. If the disturbance is random with a zero mean, then, in general, learning algorithms have been shown to be robust in filtering this out, if the learning rate is set properly. Of course this remains to be shown for this application. The ANN Simulink Block contains two functions: training the network and computing the error compensation signal. The ANN contained in this block has one input and one output. For training, $\dot{V}_c(t$-1) is used as the input, and the desired output is e(t-1). There was a correlation between these two signals because the aircraft response is caused by the sum of the error compensation signal and the PD-controller's commanded value. This ANN (that is continually trained to perform this mapping) was used to calculate the error compensation signal used in the controller for the current time step. The current commanded acceleration $\dot{V}_c(t)$ is assigned as the input to the ANNs and the network output is the compensation signal needed for this time step. The architecture of the neural network was developed by testing numerous different designs and choosing one that was sufficiently robust and provided fast adaptation to modeling errors and control failures.
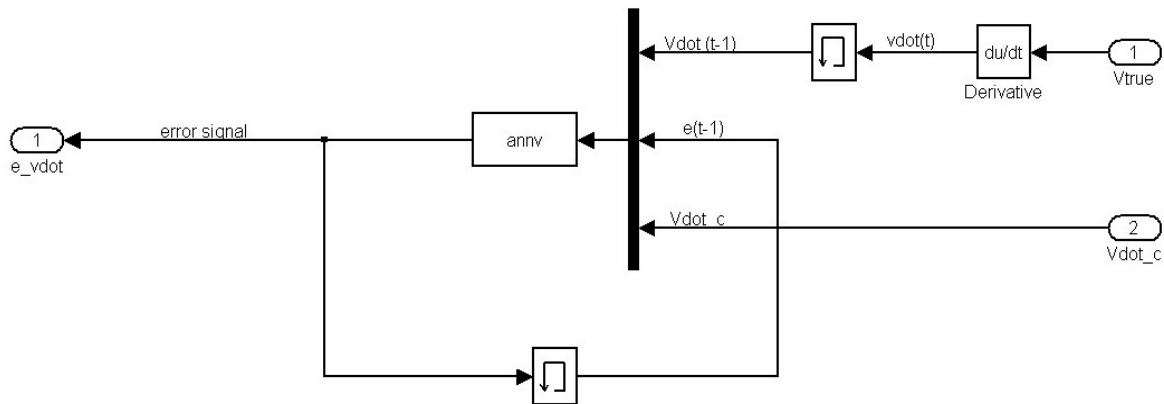

FIGURE 4. ARTIFICIAL NEURAL NETWORK FUNCTION FOR VELOCITY CONTROL CORRECTIONS

This neural network contains two hidden layers with five neurons in each hidden layer. The first hidden layer consists of neurons with tan-sigmoid activation functions. The second hidden layer and the output neuron have linear activation functions. The training technique used was the Levenberg-Marquardt algorithm with a learning rate of 0.7. The ANN used for pitch angle control is identical in structure to the velocity controller network. The linear acceleration (velocity rate) values are simply replaced with pitch angle acceleration. Figure 5 shows the details of the neural network structure. Each layer takes an input vector, multiplies it by a weight vector, adds a bias vector, and then processes each element of the resulting vector through the activation function shown. This result is then passed on to the next layer, or if it is the final layer, this becomes the network output. The weight and bias vectors for each layer are the parameters that are updated by the training algorithm. The training algorithm, at each simulation time step, compares the network output with the correct output fed to the network block. This gives an error signal that is used to adjust the weight and bias vectors a small amount in a direction that makes the error smaller. The process is called gradient descent and is an algorithm that changes the weights in the direction of the negative of the gradient, which is in the negative or downhill directions of the derivative of the output error with respect to each of the weight and bias parameters. The key is that the weights change only a small amount with each training step, but over many time steps the weights are changed in directions that makes the output converge to give the correct answer.
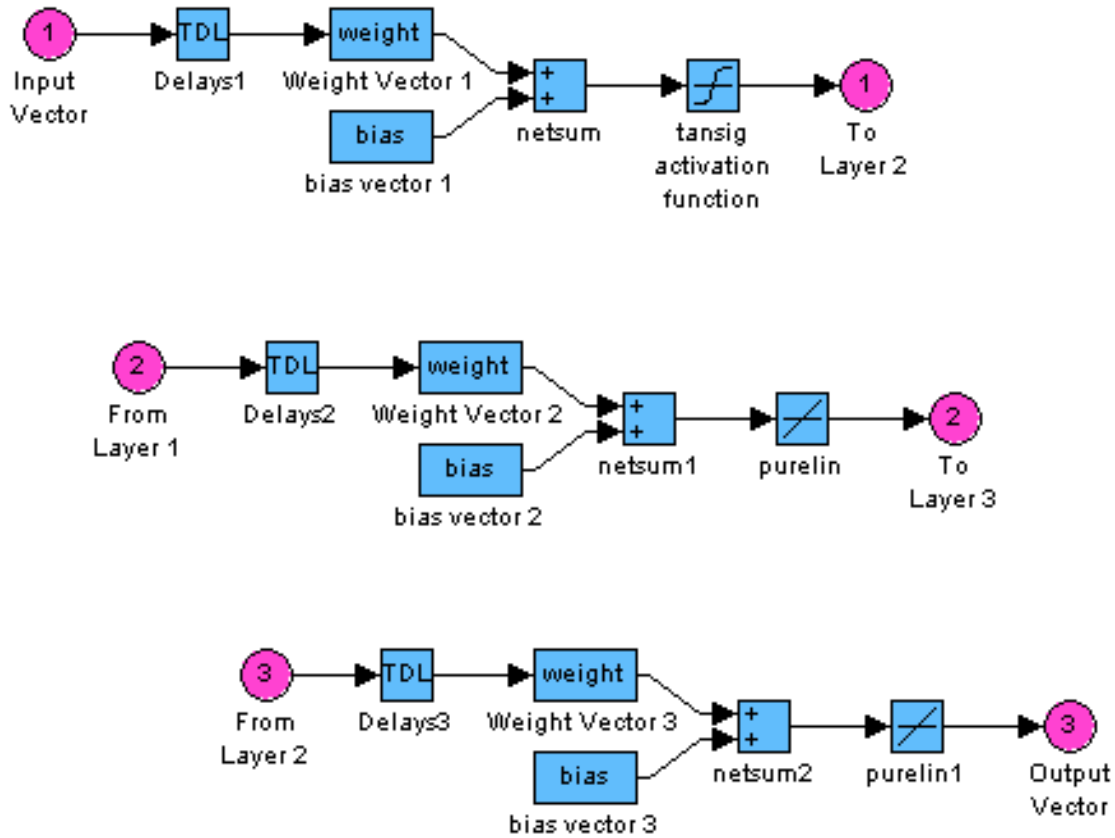

FIGURE 5. NEURAL NETWORK ARCHITECTURE DETAILS

The adaptive inverse controller was tested in MATLAB Simulink with a Raytheon Beech Bonanza F33C model in trimmed flight and with speed and pitch angle commands, both with modeling errors and partial control system failures. The ANNs were able to immediately start adapting to sudden changes in flight characteristics and during partial elevator or engine failure.

Figures 6 through 11 show the response of the Raytheon Beech Bonanza F33C in trimmed flight, where the aircraft C_{m_α} drops suddenly by 10% at 5 seconds, and the throttle/thrust gearing drops suddenly by 25% at 15 seconds (i.e., the actual engine thrust drops by 25%) It is important to clarify figure 9 and the corresponding thrust figures in the following examples. Commanded thrust shown is the thrust the inverse controller is commanding, not the actual thrust delivered by the engine. After the C_{m_α} change occurs, the pitch angle varies by less than 1 degree during the recovery that takes about 10 seconds. Figure 10 shows that the neural network starts instantly feeding in an error correction signal when the C_{m_α} change occurs. With the aid of the ANN, the aircraft response matches the commanded value again within 1 second's time. The necessary change in the elevator command is shown in figure 8. Similarly, the throttle gearing change causes only a 0.3-ft/s change in velocity (figure 7) with the ANN compensation shown in figure 11. Additional tests showed that the ANNs allow for a C_{m_α} change of up to 40% with rapid recovery and without notable oscillation.
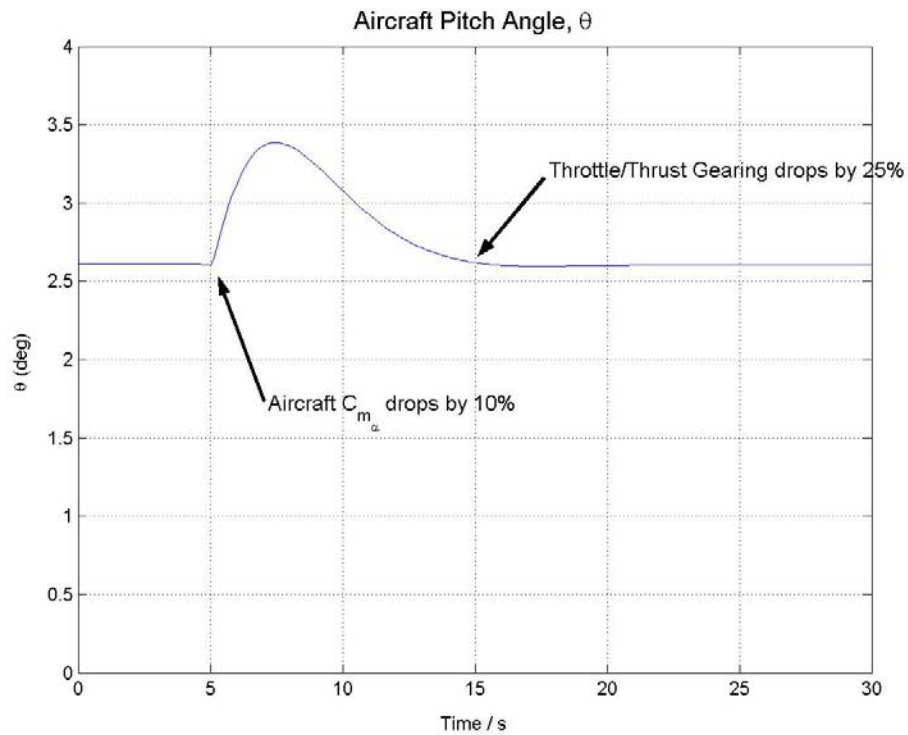

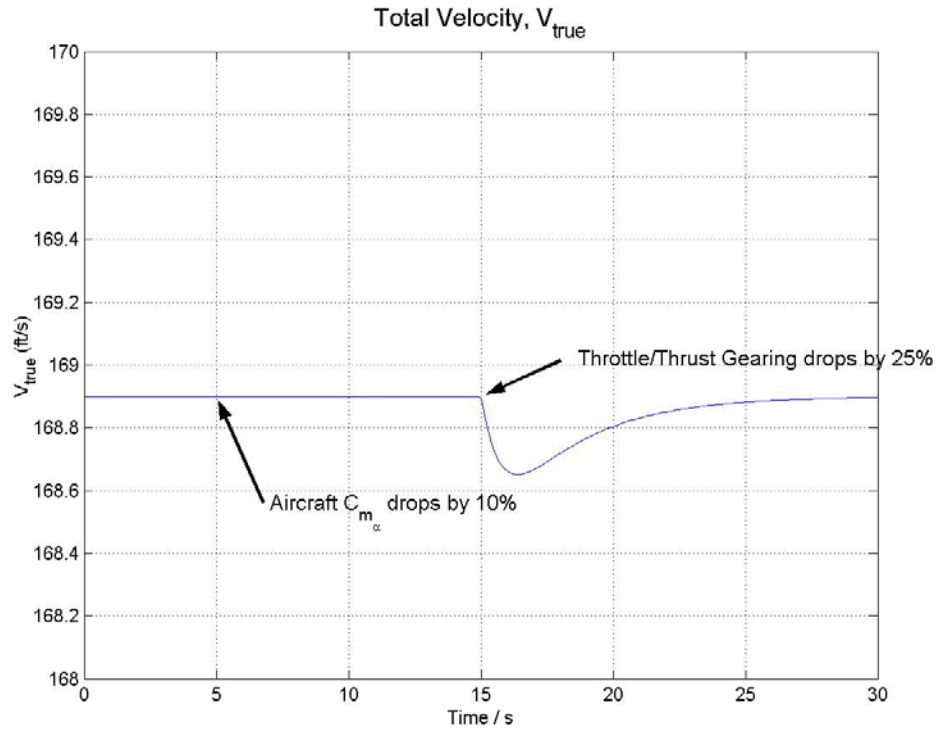FIGURE 6. PITCH ANGLE FOR TRIMMED FLIGHT WITH UNANTICIPATED FAILURES

FIGURE 7. VELOCITY FOR TRIMMED FLIGHT WITH
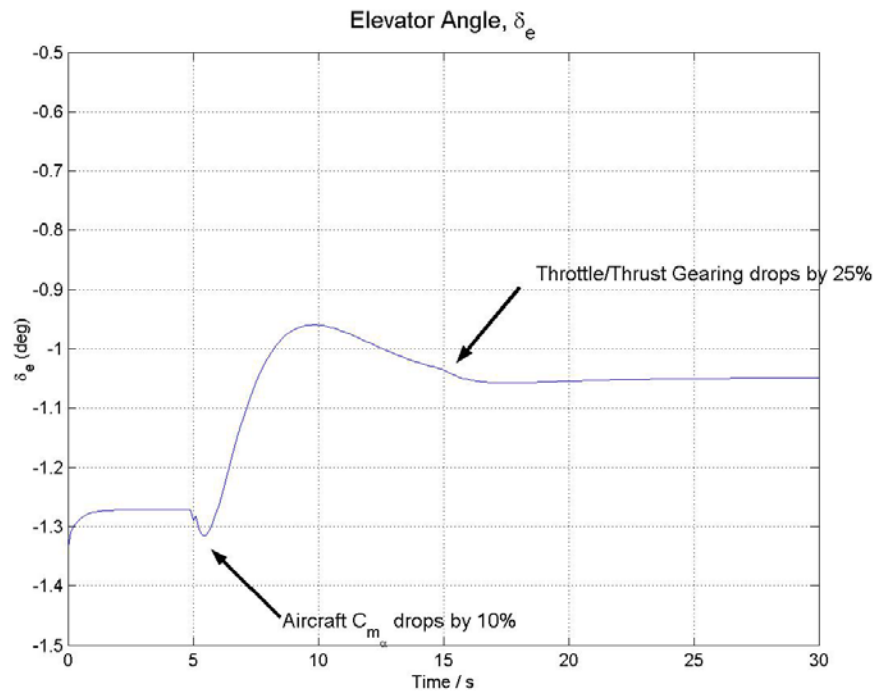UNANTICIPATED FAILURES

FIGURE 8. COMMANDED ELEVATOR POSITION FOR TRIMMED FLIGHT WITH
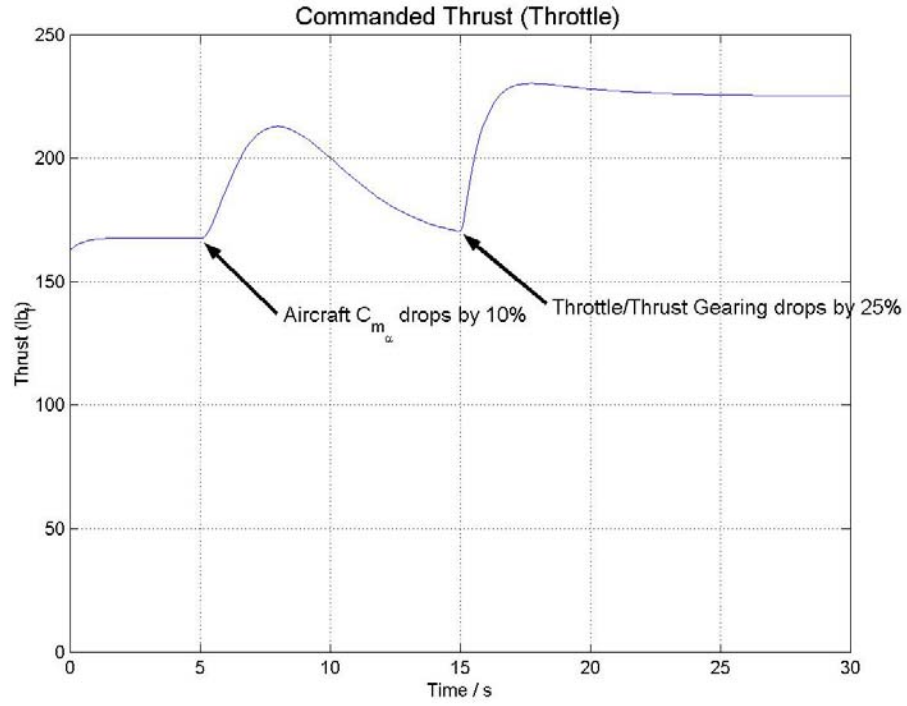UNANTICIPATED FAILURES

FIGURE 9. COMMANDED THRUST FOR TRIMMED FLIGHT WITH
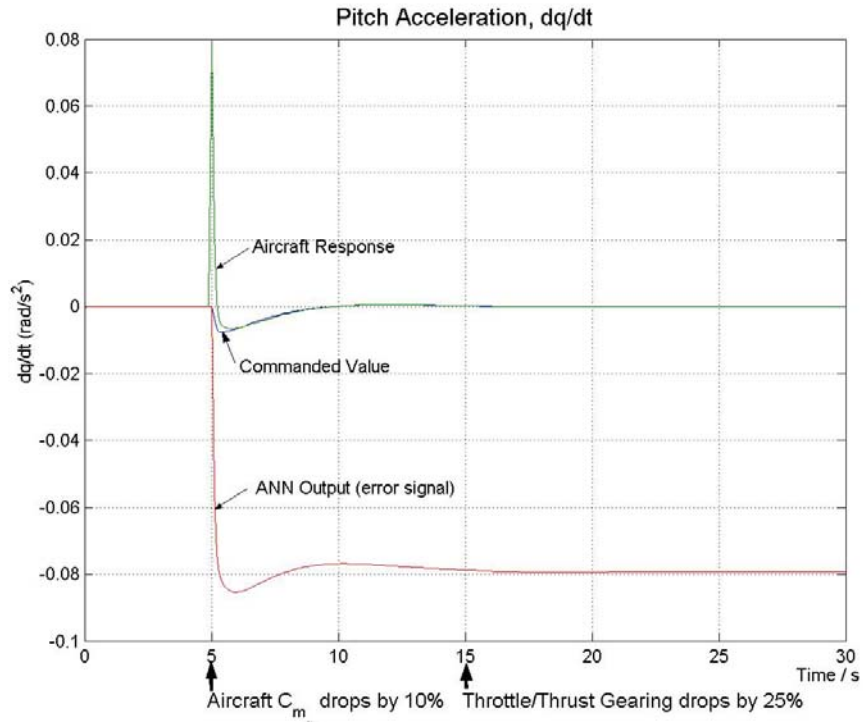UNANTICIPATED FAILURES

FIGURE 10. PITCH ACCELERATION FOR TRIMMED FLIGHT WITH
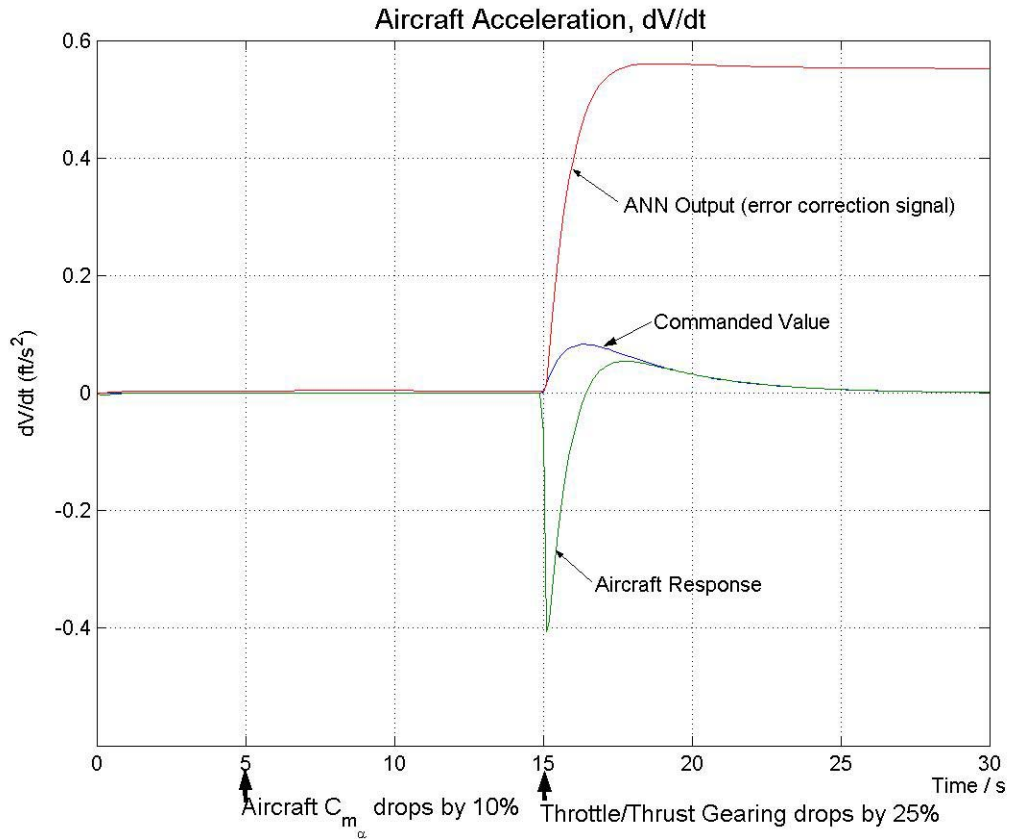UNANTICIPATED FAILURES

FIGURE 11. AIRCRAFT ACCELERATION FOR TRIMMED FLIGHT WITH
UNANTICIPATED FAILURES

Figures 12 through 17 show the response of the Raytheon Beech Bonanza F33C in controlled flight, with the same failures as above (where the aircraft C_{m_α} drops suddenly by 10% at 5 seconds and the throttle/thrust gearing drops suddenly by 25% at 15 seconds). While these changes are occurring, the pilot commands a 5 degree increase in pitch angle at 8 seconds and a 20-ft/sec increase in flight speed at 20 seconds. The ANNs make the necessary corrections with similar robustness and accuracy as in the test for the trimmed flight condition.
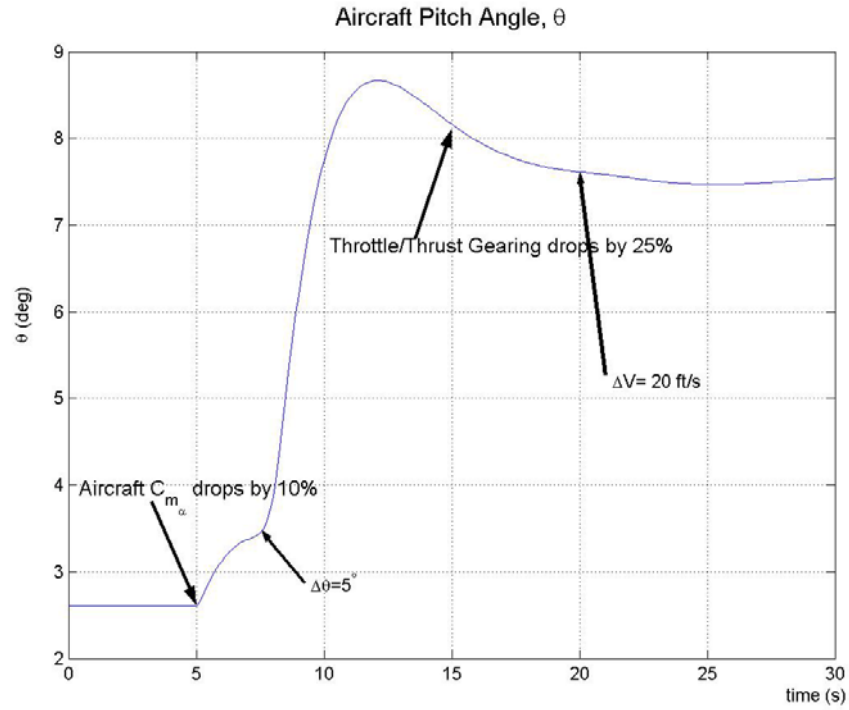
FIGURE 12. PITCH ANGLE FOR PILOT INPUTS WITH UNANTICIPATED FAILURES
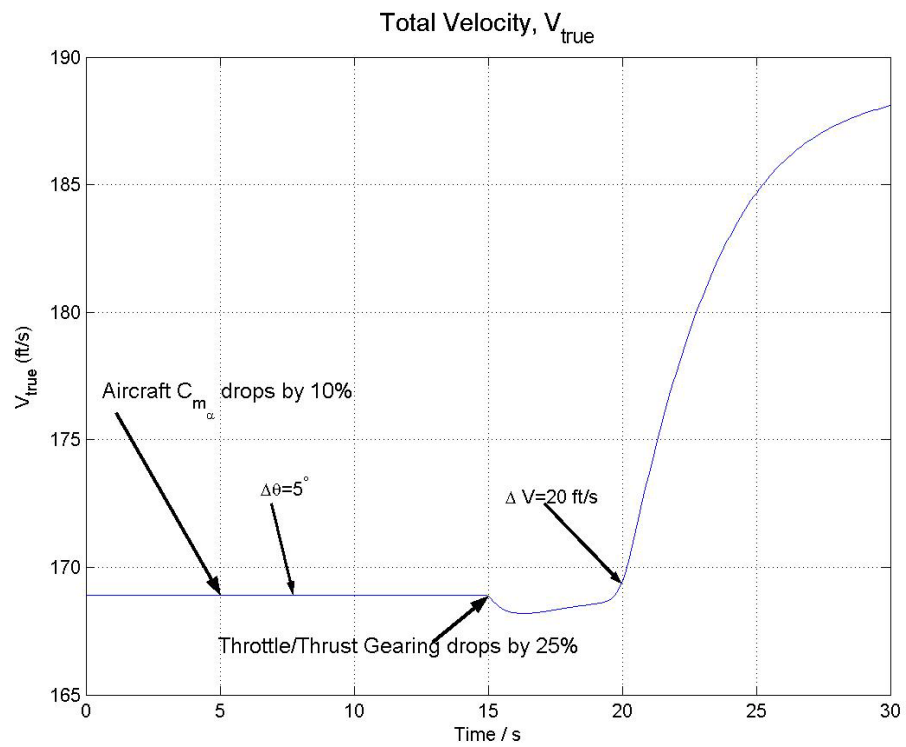

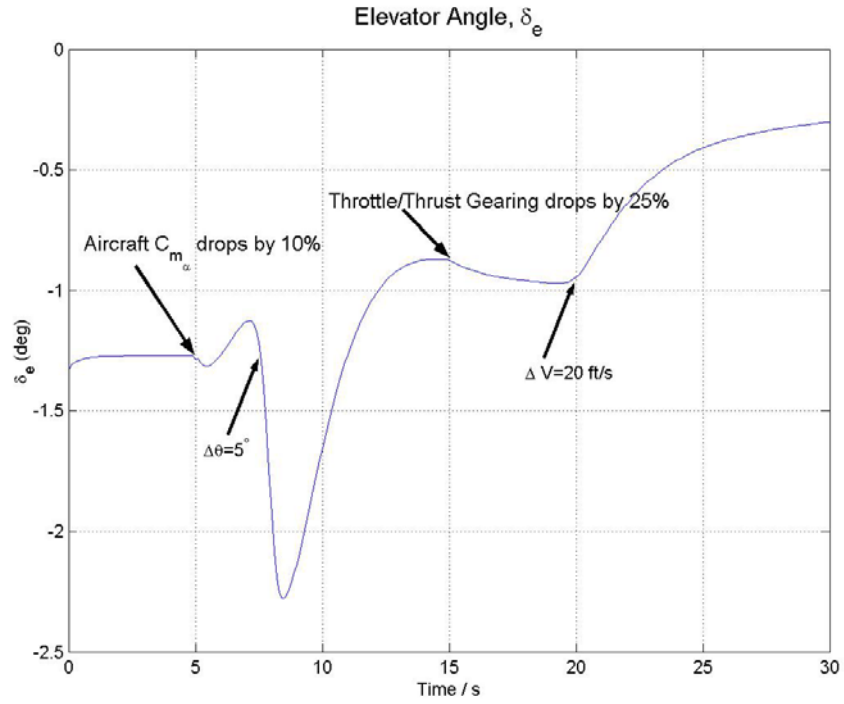FIGURE 13. VELOCITY FOR PILOT INPUTS WITH UNANTICIPATED FAILURES

FIGURE 14. COMMANDED ELEVATOR POSITION FOR PILOT INPUTS WITH
UNANTICIPATED FAILURES
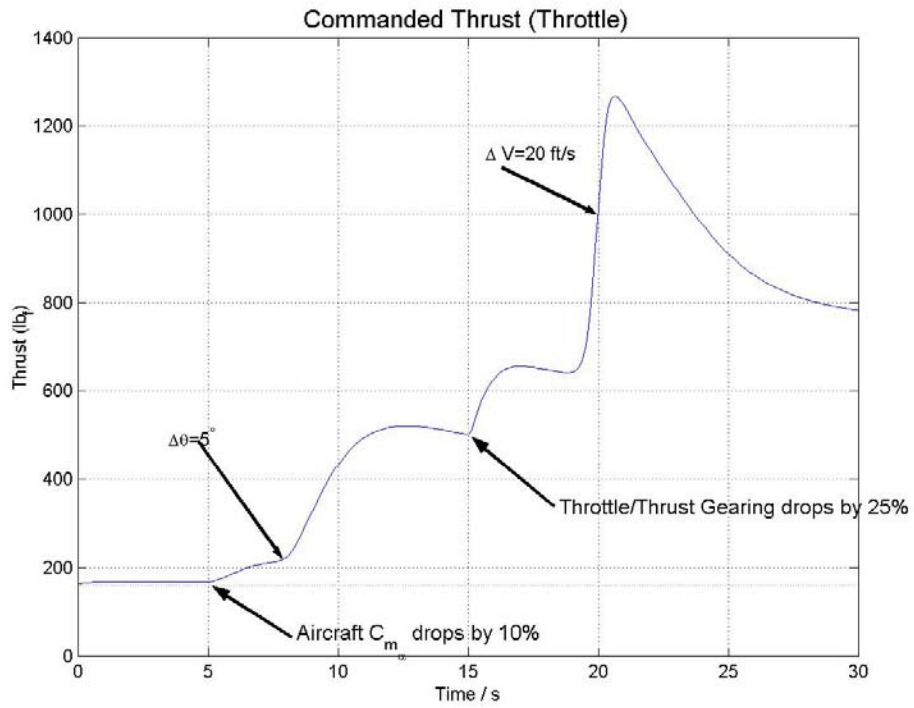

FIGURE 15. COMMANDED THRUST FOR PILOT INPUTS WITH
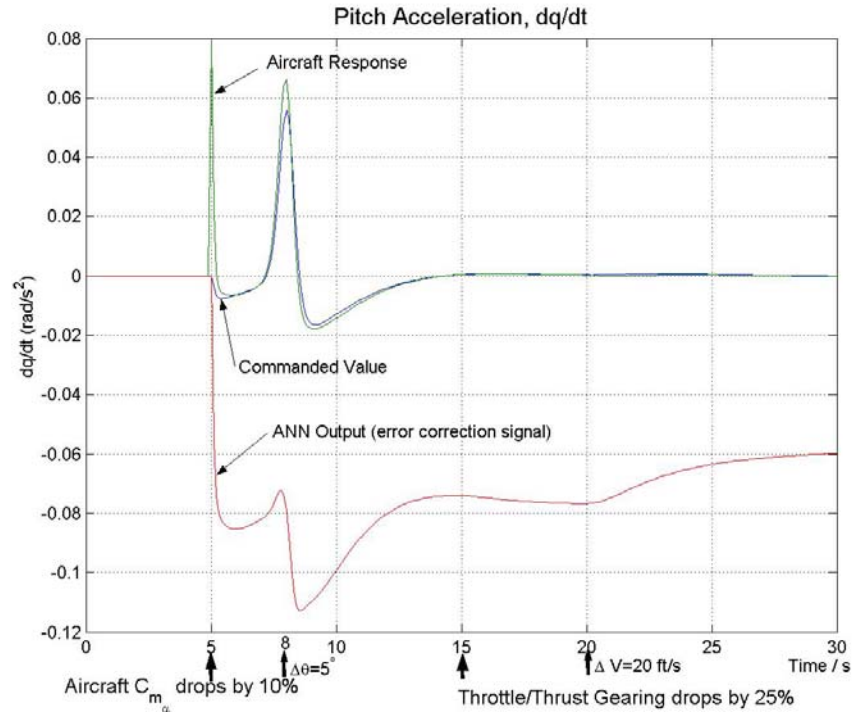UNANTICIPATED FAILURES

22

FIGURE 16. PITCH ACCELERATION FOR PILOT INPUTS WITH
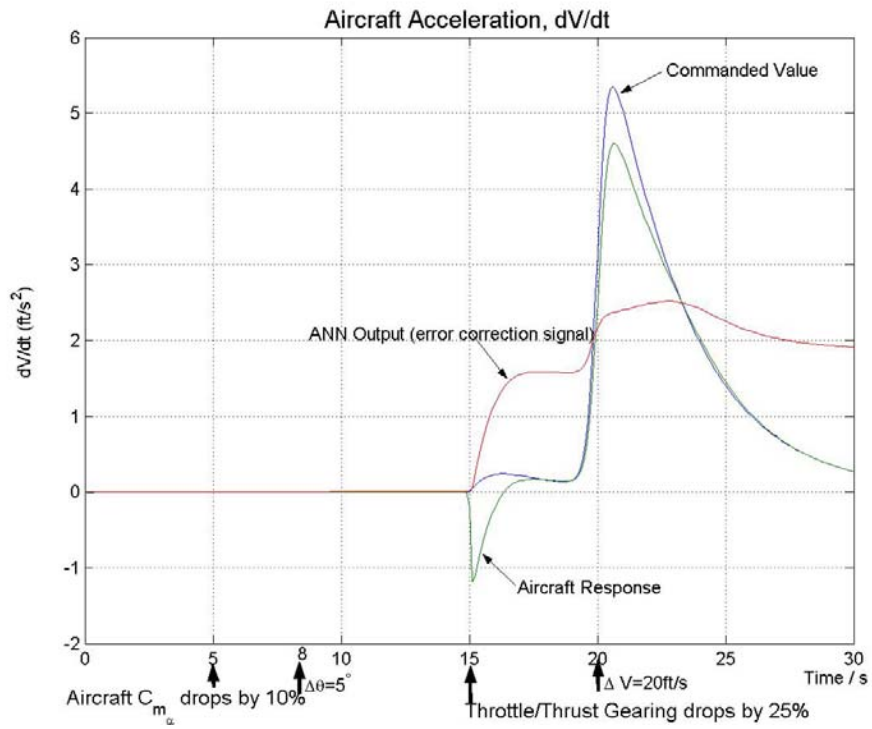UNANTICIPATED FAILURES

FIGURE 17. AIRCRAFT ACCELERATION FOR PILOT INPUTS WITH
UNANTICIPATED FAILURES

Figures 18 through 23 show the response of the Raytheon Beech Bonanza F33C when half of the elevator becomes infective (breaks off) at 0 sec., i.e., $C_{m_{\delta_e}}$ and $C_{L_{\delta_e}}$ are reduced by 50%. Here, the lateral-directional effects of the asymmetrical elevator are neglected. The figures show the ANNs make the necessary corrections, the pitch angle varies by less than 3 degrees, and the aircraft returns to the trimmed state in 60 seconds. Some initial oscillation of other variables occurs, but the amplitudes are very small. The lack of velocity perturbation with the loss of elevator was due to the fact that a change in elevator effectiveness did not significantly affect the drag equation, which was solved for thrust, so there was no significant modeling error in the velocity control loop. The pitch equation was greatly affected; therefore, a large pitch transient occurs since there was a large modeling error in the inverse control calculated for the elevator. During the pitch transient, the neural network was learning to successfully compensate for this modeling error. Note that the pitch neural network (response shown in figure 22) learns a steady-state value to compensate for the elevator failure. Also note that the velocity neural network (response shown in figure 23) also learns a very small (note 10^{-3} scale on the y axis) steady-state correction due to the fact that the aircraft trims at a slightly different angle of attack and elevator due to the change in elevator derivatives.
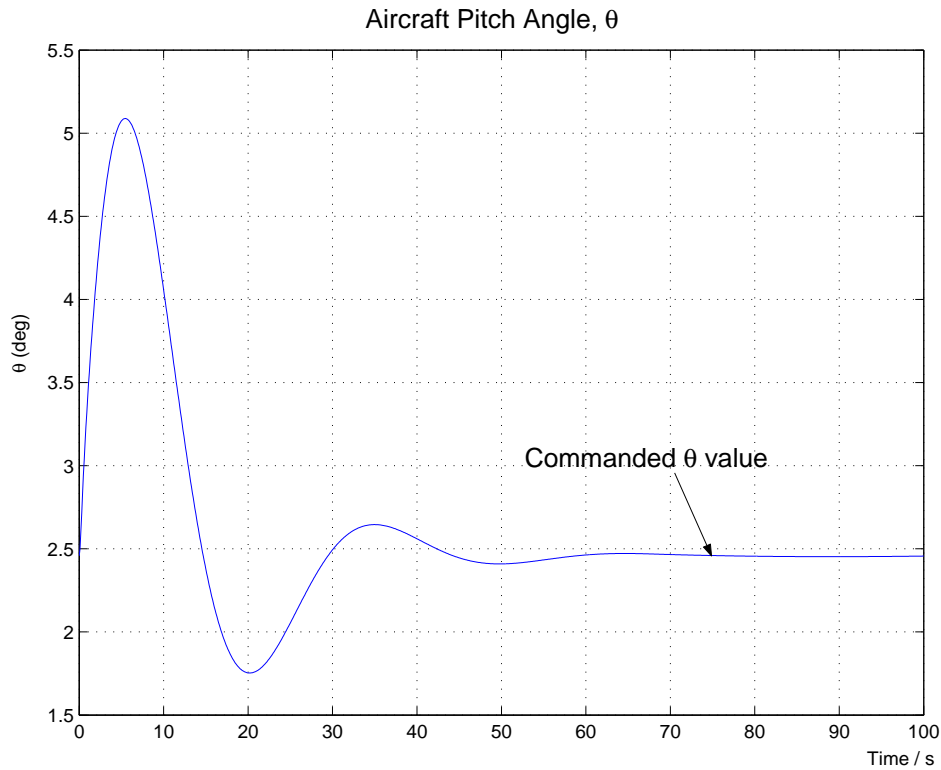

FIGURE 18. PITCH ANGLE FOR 50% LOSS OF ELEVATOR EFFECTIVENESS
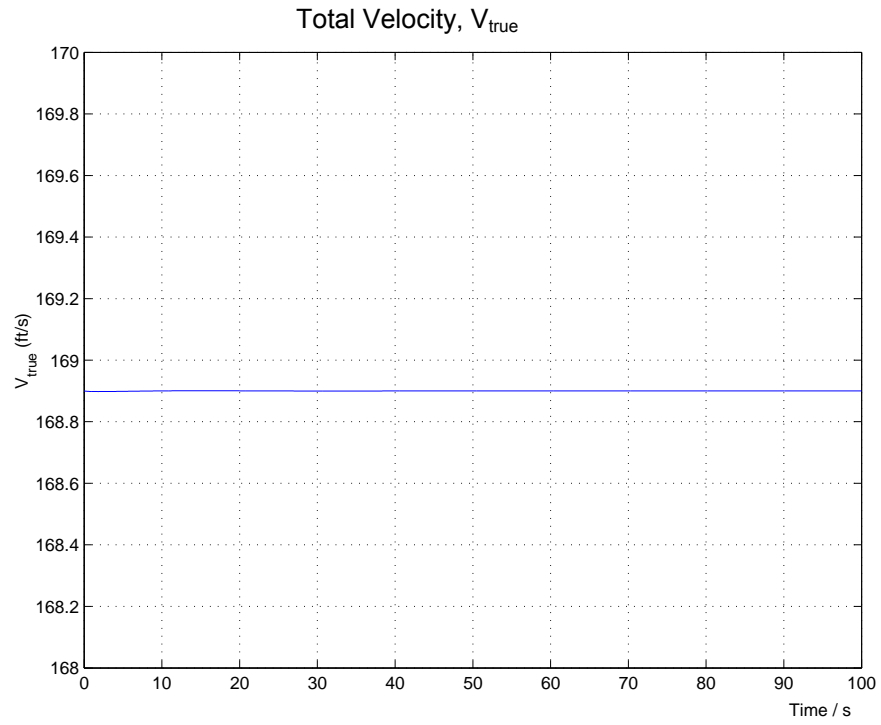
FIGURE 19. VELOCITY FOR 50% LOSS OF ELEVATOR EFFECTIVENESS
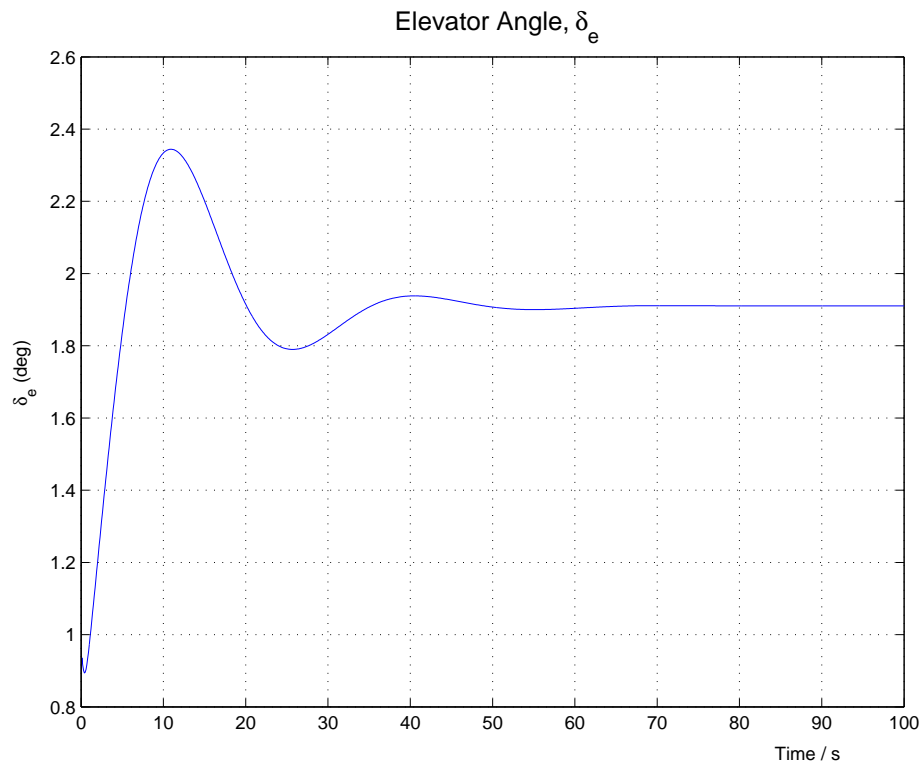

FIGURE 20. COMMANDED ELEVATOR POSITION FOR 50% LOSS OF
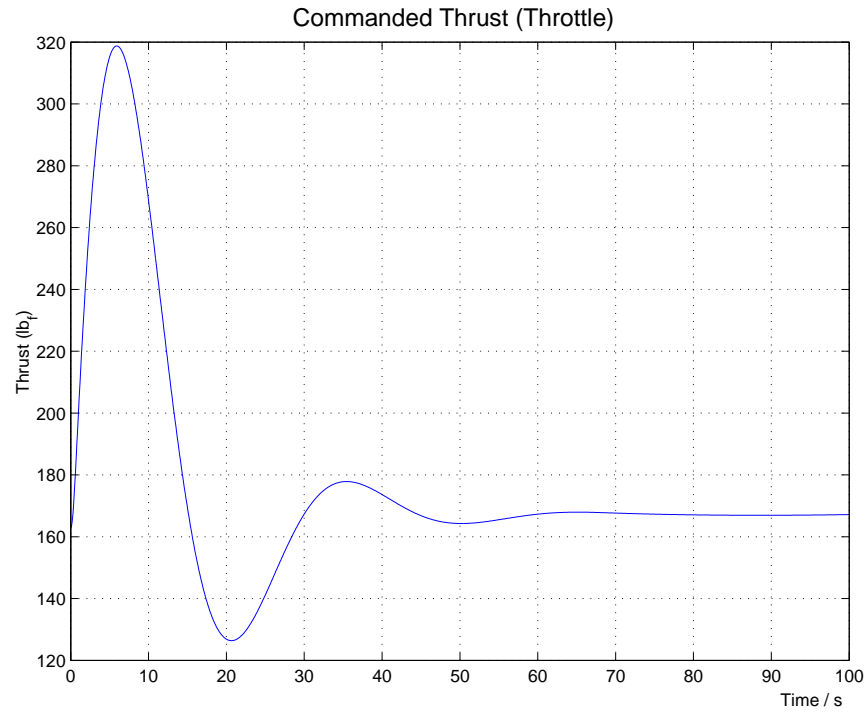ELEVATOR EFFECTIVENESS

FIGURE 21. COMMANDED THRUST FOR 50% LOSS OF ELEVATOR EFFECTIVENESS
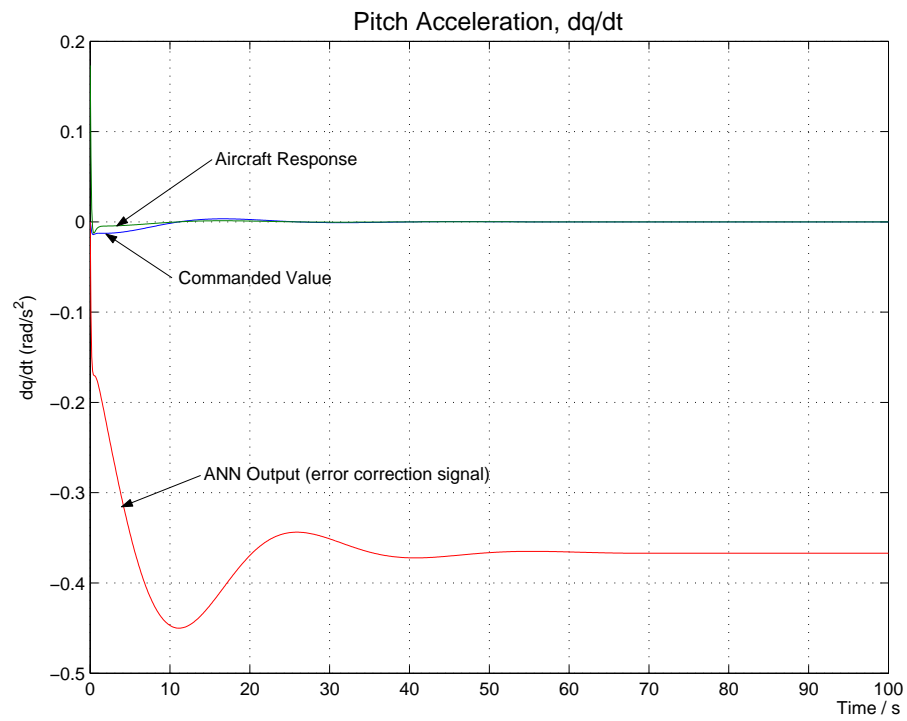

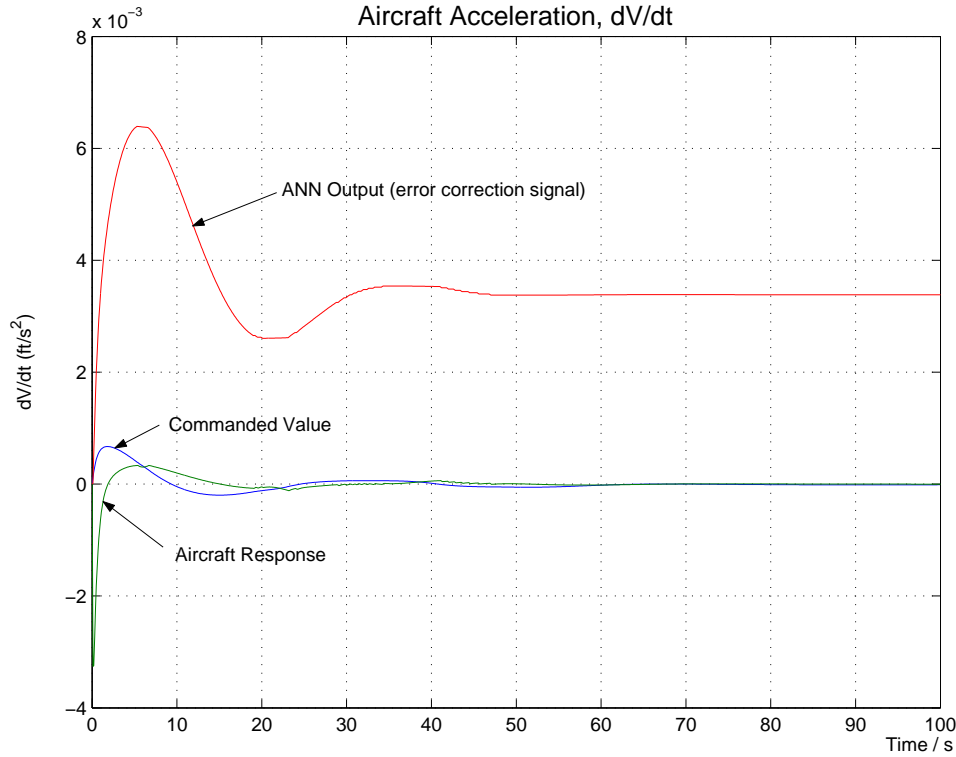FIGURE 22. PITCH ACCELERATION FOR 50% LOSS OF ELEVATOR EFFECTIVENESS

FIGURE 23. AIRCRAFT ACCELERATION FOR 50% LOSS OF ELEVATOR
EFFECTIVENESS

Gamma tracking was achieved with as much success as theta tracking. Figures 24 through 28 show the response of the Bonanza F33C responding to a commanded level flight path angle of 0 degrees and a commanded decrease in flight velocity of 70 ft/sec at 3 seconds into the simulation with the following failures occurring: the aircraft C_{m_α} drops suddenly by 10% at 20 seconds and the throttle/thrust gearing drops suddenly by 25% at 25 seconds. Notice that the neural networks learn to compensate for the failures while tracking the commanded flight path and velocity. When the throttle saturates at its minimum value, the elevator controller continues to maintain the pitch angle. The velocity controller was calculating negative-commanded thrust correctly, but since the aircraft cannot respond, due to the thrust saturation, the response was essentially open loop in velocity control. As soon as the thrust becomes unsaturated, however, the velocity control becomes effective again and tracks the velocity command.

Windup effects due to the throttle saturation have been observed in the early design without windup compensation. When the throttle saturates, the neural network sees the aircraft not responding to the commanded throttle (the controller is commanding a negative throttle to slow the aircraft down). The network sees this as a modeling error and adapts to try to compensate for it in much the same way as traditional integral control will windup trying to compensate for a steady-state error in the presence of control saturation. The network cannot compensate since the throttle is saturated, and the network weights train to erroneously huge values (similar to integral windup) that cause large speed changes when the throttle does finally unsaturate. Therefore an antiwindup method was implemented in the final design. The antiwindup method

used for the neural network was to stop training when the control associated with that neural network was saturated. When the control becomes unsaturated, the neural network begins to train again from where it left off with no re-initialization. Rysdyk has investigated a more elegant antiwindup method called pseudocontrol hedging, which limits the commanded accelerations to prevent control saturation using a model-following controller. This has not been implemented in the controller, but should be investigated extensively in the near future.

This final simulation only includes a first-order engine model (time constant = 1/2 sec.) and an elevator actuator model (time constant = 1/5 sec.). These models are included in the aircraft simulation block, but are not included in the inverse control calculations. Adding these to the inverse controller has been discussed, but no satisfactory method has yet been developed. The commanded thrust shown in figure 28 includes the engine model, since the thrust shown is saturated at its lower limit at about 18 seconds. The thrust calculated by the inverse controller follows the plot shown, except that it goes very negative during the time between 18 and 50 seconds.
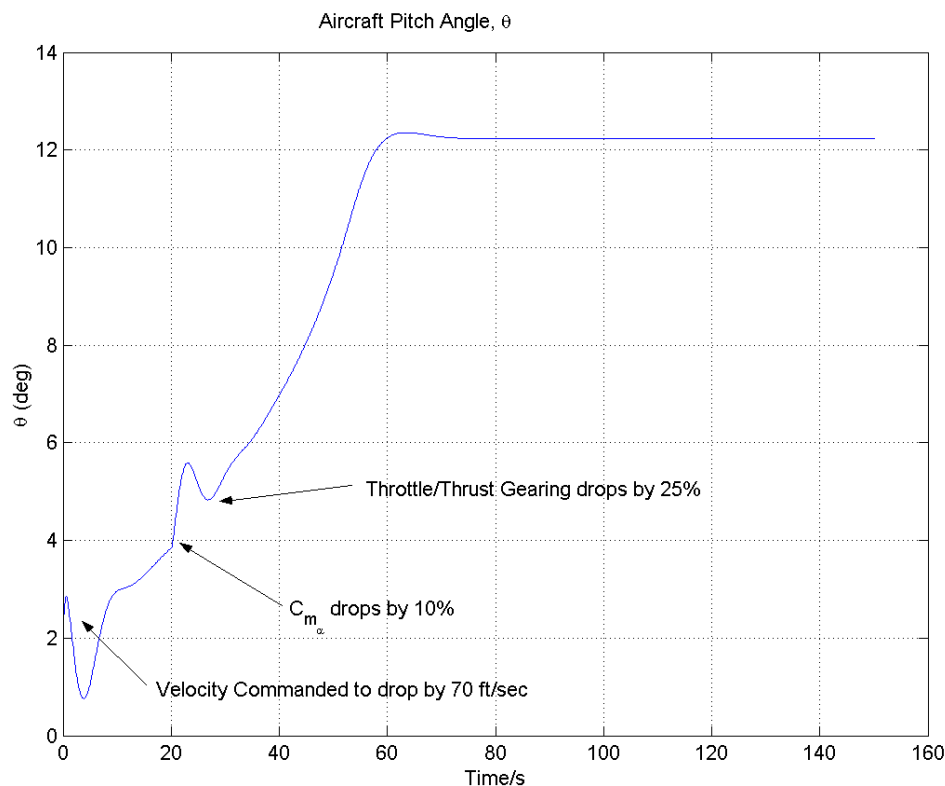
FIGURE 24. PITCH ANGLE FOR GAMMA TRACKING WITH VELOCITY AND C_{m_α} FAILURES
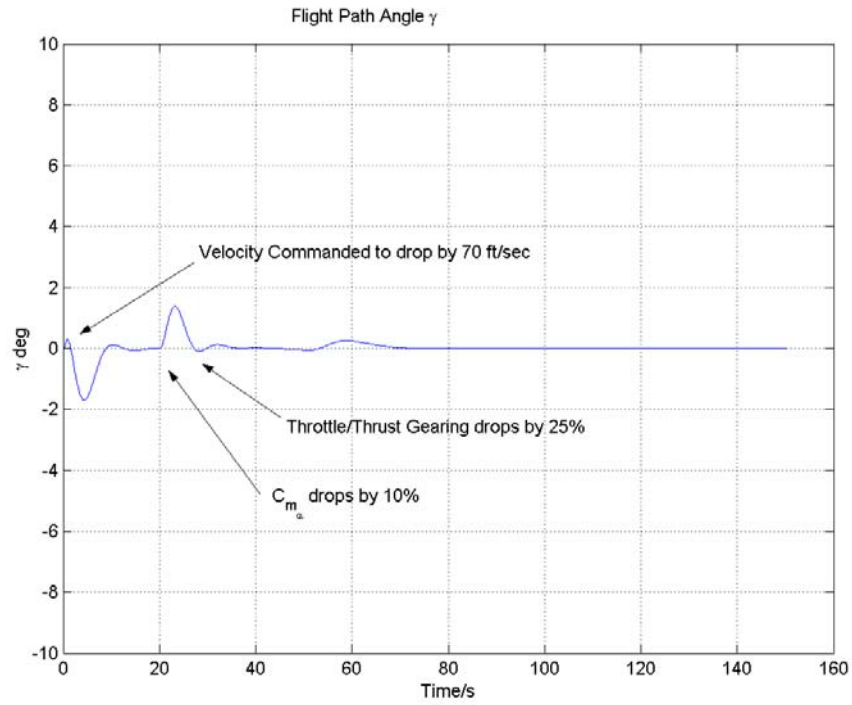
FIGURE 25. FLIGHT PATH ANGLE FOR GAMMA TRACKING WITH VELOCITY
AND C_{m_α} FAILURES
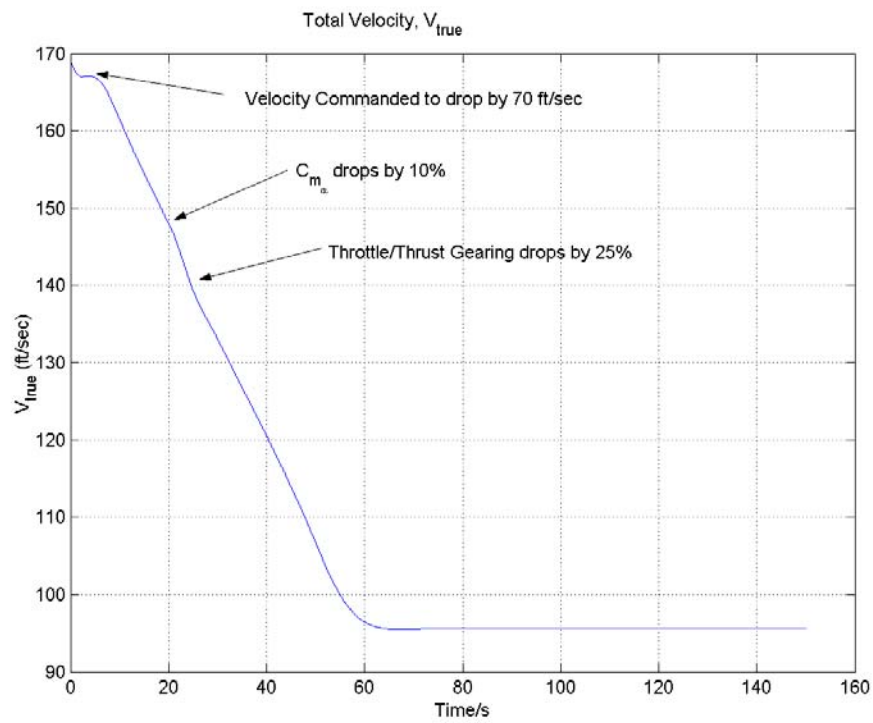
FIGURE 26. VELOCITY FOR GAMMA TRACKING WITH VELOCITY
AND C_{m_α} FAILURES
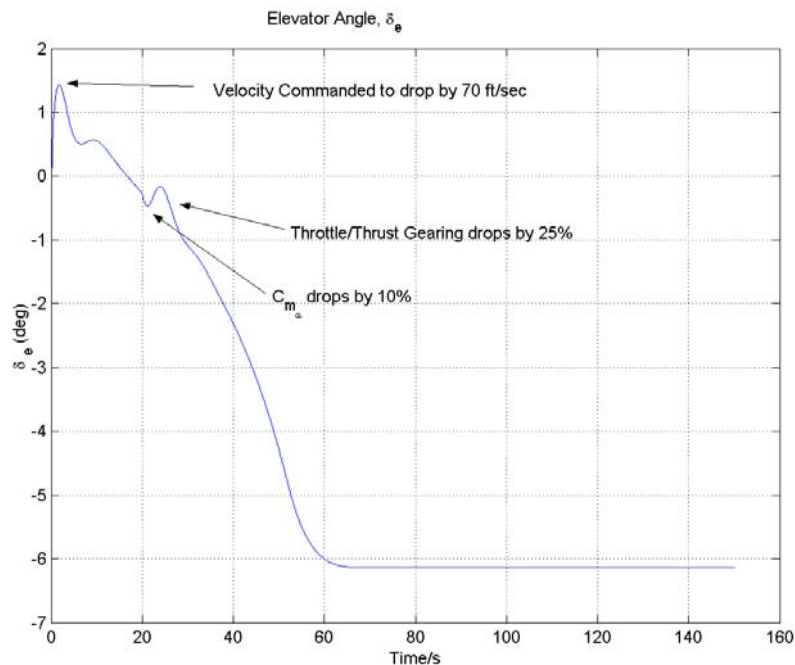
FIGURE 27. COMMANDED ELEVATOR POSITION FOR GAMMA TRACKING WITH
VELOCITY AND C_{m_α} FAILURES
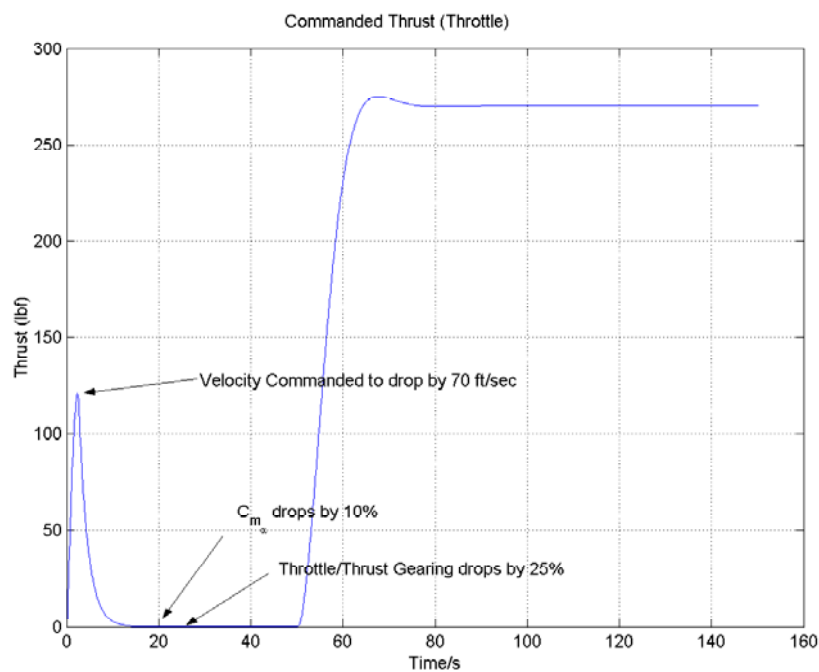
FIGURE 28. COMMANDED THRUST FOR GAMMA TRACKING WITH VELOCITY
AND C_{m_α} FAILURES

For reference purposes, figures 29 and 30 show the aircraft response to a 5 degree commanded change in pitch angle at 8 seconds with no failures. Figures 31 and 32 show the aircraft response to a commanded 20-ft/sec increase in velocity at 3 seconds with no failures.
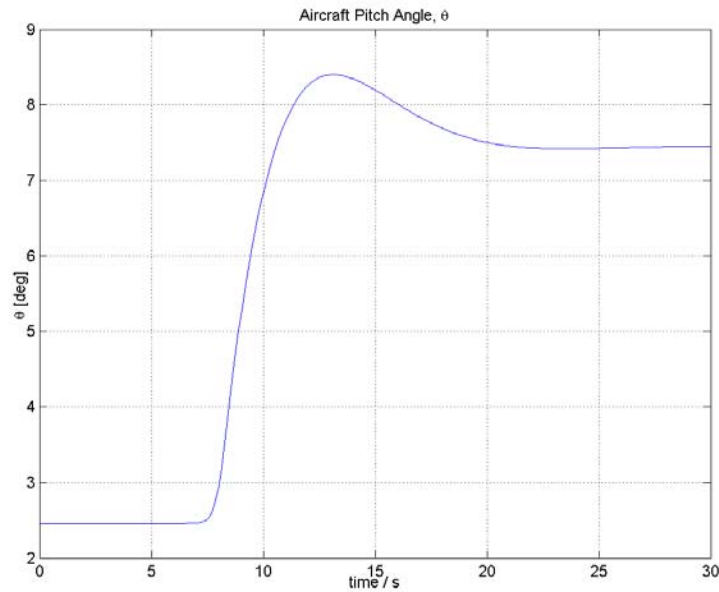
FIGURE 29. PITCH ANGLE RESPONSE TO A 5 DEGREE PITCH ANGLE CHANGE WITH NO FAILURES
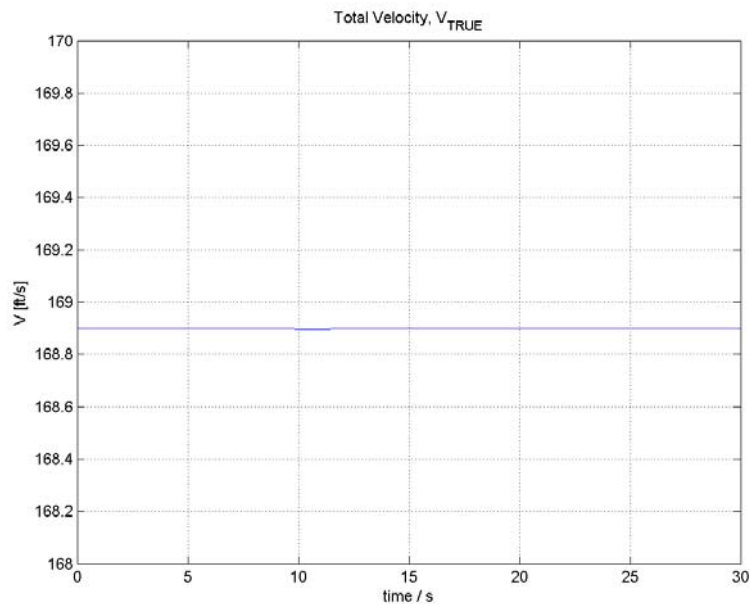
FIGURE 30. VELOCITY RESPONSE TO A 5 DEGREE PITCH ANGLE CHANGE WITH NO FAILURES

31
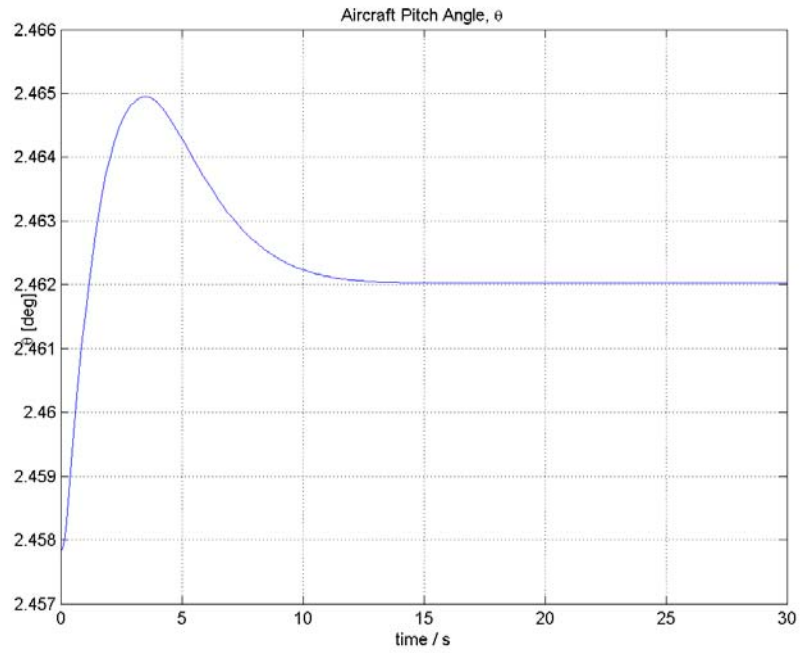
FIGURE 31. PITCH ANGLE RESPONSE TO A 20-ft/sec VELOCITY CHANGE WITH
NO FAILURES
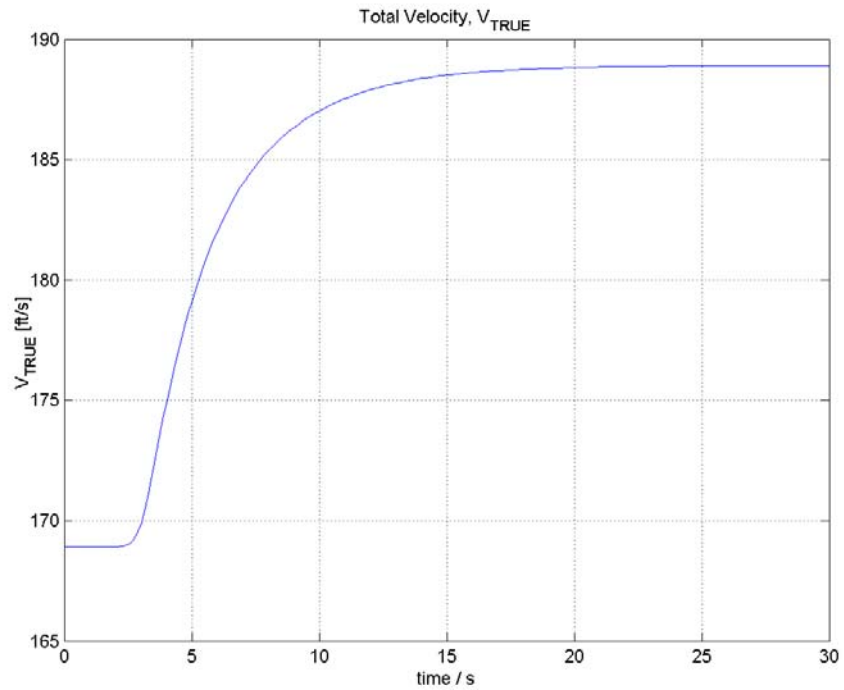
FIGURE 32. VELOCITY RESPONSE TO A 20-ft/sec VELOCITY CHANGE WITH
NO FAILURES

5. LONGITUDINAL CONTROLLER FLIGHT TESTING.

The longitudinal controller has been tested initially in the MATLAB environment. The controller is currently in the process of being implemented as C-code (ANSI-C format) on the actual aircraft. The nonlinear adaptive control architecture was incorporated into the existing Raytheon Bonanza Velocity Vector Command with Envelope Protection control system (EZ-fly developed by Tony Lambregts [13 and14] and modified and implemented on the Raytheon Beech Bonanza F33C by Noel Duerksen). The Raytheon Beech Bonanza F33C fly-by-wire test bed is a development platform for advanced controls, advanced displays, flight controls integrated with Pilot Fault Display and Multifunction Display, Attitude and Heading Reference System/Air Data Computer, and Datalink.

The MATLAB Simulink system development environment was used to develop and test the adaptive inverse controller. The Real-Time Workshop toolbox was used to dump embedded system C code directly from the Simulink control system model. This code was then compiled as a subroutine embedded in the existing Raytheon Beech Bonanza F33C flight computer EZ-fly software. This system replaced portions of two subroutines in the EZ-fly software that calculates engine throttle position and elevator position that will be output from the flight computer to command engine throttle and elevator actuators.

Flight testing on the Raytheon Beech Bonanza F33C is currently planned for spring 2003. The following goals are set for flight testing:

- To demonstrate velocity vector command with nonlinear adaptive control architecture.

- To demonstrate adaptive control effectiveness with changing configuration and payloads.

- To investigate the system performance in the presence of external disturbances such as atmospheric turbulence.

6. MATLAB SIMULINK CONTROL SOFTWARE CERTIFICATION.

6.1 OVERVIEW OF THE DO-178B.

Radio Technical Commission for Aeronautics, RTCA/DO-178B (Software Considerations in Airborne Systems and Equipment Certification) is the standard used by most applicants for approval of airborne software in certification projects. To quickly summarize, DO-178B requires the following tasks:

- Determine the software level based on the aircraft-level safety assessment.

- Define the software development process and document in plans.

- Define the software requirements based on the system requirements.

- Implement the requirements (coding, integration, etc.)

- Verify that the requirements are met.

- Carryout the integral processes (configuration management, verification, and quality assurance) throughout the development project.

6.2 THE DO-178B CERTIFICATION REQUIREMENTS.

Since this project was research-based, rather than an actual certification project, DO-178B was not used. If an applicant desires to certify this project, DO-178B would be applicable for the applicants. There are several approaches that an applicant might consider, using the Simulink models.

1. The Simulink models may be considered as the high-level software requirements. They may then be used in the applicant's typical development process—using traditional requirements-to-design-to-code process.

2. The Simulink tool may be used to generate the source from the models. Then the source code could be verified by the developer. If the output of the tool is verified, tool qualification is not required.

3. Simulink models may be ported to a qualified development tool for source code generation.

4. Simulink may be qualified as a development tool.

Each of the suggested approaches has pros and cons that should be carefully considered by the applicant.

6.3 ALGORITHM DETERMINISM.

One of the largest certification barriers is the determinism of the algorithms. In addition to developing the software to meet DO-178B objectives, the algorithms of the flight controls must be proven to be deterministic to the required level (i.e., the required safety level). Once the algorithms are shown to be deterministic, implementation of DO-178B is like any other project. This project has not addressed this deterministic aspect, but it is important to realize that it must be addressed before the approach can be used in projects seeking certification.

6.4 ISSUES PERTAINING TO THE CURRENT PROJECT.

The above discussion points to two issues. First, MATLAB is used extensively in the development and testing of flight control systems, therefore, the ability to generate reliable code for these types of system needs to be addressed. Second, the algorithms used in the control systems are straightforward and deterministic, except for the ANN learning, which depends on the learning data it is presented. This is not to say that the neural network learning algorithm is not a straightforward calculation, because it is a relatively simple calculation without any decision branches. However, the cumulative result of the learning algorithm being applied to numerous data resulting from the response of the aircraft to previous calculations of the network does not seem to lend itself to straightforward testing, leading to certification. This needs to be

examined carefully from the standpoint of certification, especially from the standpoint of theoretical results, which have established guaranteed bounds for the control system, including the learning algorithm for the adaptation of the weights in the neural network.

7. LONGITUDINAL INVERSE CONTROLLER FORMULATION DETAILS.

The derivation of the inverse controller for the longitudinal motion involves specifying the commanded aircraft accelerations and solving for the control settings required for achieving these accelerations. This is opposite of the normal forward aircraft model where the control settings are specified and the resulting motion of the aircraft is solved for.

In body coordinates, the equations of motion for longitudinal maneuvers with the level wings are [12]

$$\dot{U} + g\sin(\theta) = -WQ + \frac{1}{m}F_{A_x} + \frac{1}{m}T\cos(\varphi_T) + \frac{1}{m}F_{X_{\delta_e}}\delta_e \tag{12}$$

$$\dot{W} - g\cos(\theta) = +UQ + \frac{1}{m}F_{A_z} - \frac{1}{m}T\sin(\varphi_T) + \frac{1}{m}F_{Z_{\delta_e}}\delta_e \tag{13}$$

$$\ddot{\theta} = \frac{1}{I_{yy}}M_A - \frac{1}{I_{yy}}Td_T + \frac{1}{I_{yy}}M_{\delta_e}\delta_e \tag{14}$$

where d_T is the thrust moment arm and φ_T is the thrust angle relative to the fuselage axis. In addition, the true airspeed and flight path angle γ are defined as

$$V_p = \sqrt{U^2 + W^2} \tag{15}$$

$$\theta = \gamma + \tan^{-1}\left(\frac{W}{U}\right) \tag{16}$$

An output feedback controller is set up to track the flight path angle and the flight speed. The tracking error for each is fed to a PID controller that outputs commanded accelerations $\dot{V}_p$ and $\ddot{\gamma}$ that are inputs to the inverse controller. Therefore

Given:　　　　$\dot{V}_p$ and $\ddot{\gamma}$ or $\ddot{\theta}$
Find:　　　　$\theta, T, \delta_e, U, W$

This constitutes a mathematically well-posed problem with five equations and five unknown parameters. The following three statements constitute auxiliary equations that go along with the above.

$$Q = \dot{\theta} \tag{17}$$

$$\dot{\theta} = \dot{\gamma} + \dot{\alpha} = \dot{\gamma} + \frac{\dot{W}U - \dot{U}W}{V_p^2} \tag{18}$$

$$\dot{V}_p V_p = \dot{U}U + \dot{W}W \tag{19}$$

Equations 12 through 16 must be solved simultaneously for given time histories of V_p and γ. Equations 12 through 14 are in the body-fixed axis system (X,Z). The commanded forward acceleration $\dot{V}_p$ and normal acceleration resulting in a flight path angular acceleration $\ddot{\gamma}$ are in an axis system parallel and normal to the velocity vector (stability coordinates), therefore, equations 12 and 13 are transformed into the V_p and γ coordinate system.

Multiply equation 12 by $\left(\dfrac{W}{V_p^2}\right)$, equation 13 by $\left(\dfrac{U}{V_p^2}\right)$, and subtract equation 12 from equation 13

$$\frac{U\dot{W} - W\dot{U}}{V_p^2} - \frac{g}{V_p}\cos(\theta - \alpha) = Q + \frac{1}{mV_p}\left[F_{A_z}\cos(\alpha) - F_{A_x}\sin(\alpha)\right]$$

$$-\frac{T}{mV_p}\sin(\alpha + \varphi_T) + \frac{1}{mV_p}\left[F_{Z_{\delta_e}}\cos(\alpha) - F_{X_{\delta_e}}\sin(\alpha)\right]\delta_e.$$

Note that
$$\cos(\theta - \alpha) = \cos(\gamma)$$
$$Q = \dot{\gamma} + \dot{\alpha}$$
$$F_{A_z}\cos(\alpha) - F_{A_x}\sin(\alpha) = -\bar{q}SC_L\big|_{\delta_e=0}$$
$$F_{Z_{\delta_e}}\cos(\alpha) - F_{X_{\delta_e}}\sin(\alpha) = -\bar{q}SC_{L_{\delta_e}}$$

Therefore, rearranging and simplifying results in

$$\dot{\gamma} + \frac{g}{V_p}\cos(\gamma) - \frac{\bar{q}S}{mV_p}C_L\big|_{\delta_e=0} - \frac{T}{mV_p}\sin(\alpha + \varphi_T) - \frac{\bar{q}S}{mV_p}C_{L_{\delta_e}}\delta_e = 0. \tag{20}$$

In the inverse problem, this equation involves three unknowns, $\alpha, T,$ and δ_e. Two other equations with the same unknowns are needed.

Now, multiply equation 12 by $\left(\dfrac{U}{V_p^2}\right)$ and equation 13 by $\left(\dfrac{W}{V_p^2}\right)$ and add

$$\frac{\dot{V}_p}{V_p} - \frac{g}{V_p}\sin(\theta - \alpha) = \frac{1}{mV_p}\left[F_{A_z}\sin(\alpha) + F_{A_x}\cos(\alpha)\right]$$

36

$$+\frac{T}{mV_p}\cos(\alpha+\varphi_T)+\frac{1}{mV_p}\left[F_{Z_{\delta_e}}\sin(\alpha)+F_{X_{\delta_e}}\cos(\alpha)\right]\delta_e$$

Note that
$$F_{A_z}\sin(\alpha)+F_{A_x}\cos(\alpha)=-qSC_D\big|_{\delta_e=0}$$
$$F_{Z_{\delta_e}}\sin(\alpha)+F_{X_{\delta_e}}\cos(\alpha)=-qSC_{D_{\delta_e}}$$

Then, again rearranging and simplifying results in

$$-\dot{V}_p-g\sin(\gamma)-\frac{qS}{m}C_D\big|_{\delta_e=0}+\frac{T}{m}\cos(\alpha+\varphi_T)-\frac{qS}{m}C_{D_{\delta_e}}\delta_e=0. \tag{21}$$

From equation 14, since
$$\ddot{\theta}=\ddot{\gamma}+\ddot{\alpha}$$

$$\ddot{\alpha}+\ddot{\gamma}=\ddot{\theta}=\frac{qS\overline{c}}{I_{yy}}\left[C_m-C_T\frac{d_T}{\overline{c}}+C_{m_{\delta_e}}\delta_e\right] \tag{22}$$

A speed control lever (replacing the cockpit throttle) was set to command airspeed V_F. When this is compared with the feedback of actual airspeed, a linear proportional controller that outputs a commanded acceleration processes the resulting error $\dot{V}_p$. Similarly, longitudinal stick can be set up to commands flight path angle γ or pitch angle θ. The tracking error of either of these variables was fed to a PD-controller that outputs a commanded angular acceleration $\ddot{\theta}$ or $\ddot{\gamma}$. Then equations 20 through 22 can be solved simultaneously for $\gamma, T,$ and δ_e. Other variables can be determined from the auxiliary equations.

An approximate solution can be obtained if one assumes $C_{D_{\delta_e}}\approx 0$. Then equation 21 can be solved for T, from which equation 22 can be solved for δ_e. Plugging these into equation 20 results in a single differential equation for γ. Note that these equations are linear in terms of the controls, as long as control saturation does not occur.

As mentioned in section 4, a note of caution must be inserted here related to when this work is extended to include turbulence, wind, and wind shear. The various terms in the inverse controller must be carefully examined as to whether they are inertial terms or air mass velocity terms. For example, alpha and airspeed are air mass-referenced, whereas theta and commanded acceleration (velocity change) are inertial-referenced.

8. LATERAL-DIRECTIONAL INVERSE CONTROLLER FORMULATION.

Adding control of lateral-directional flight of the aircraft requires considering the full 6 DOF of the aircraft equations of motion. The EZ-fly velocity vector command system extends to full flight control by keeping the longitudinal control strategy of controlling speed with a speed control lever and flight path angle (altitude rate) with longitudinal stick. In addition, a coordinated turn is controlled by commanded bank angle (turn rate) with lateral stick along with

requiring the turn be coordinated with feet on the floor (no rudder pedal). This leads to another linear feedback loop to control bank angle, and an inverse controller that takes the resulting commanded bank angle acceleration, coupled with the commanded yaw acceleration required to coordinate the turn, and calculates the required aileron and rudder deflections. The longitudinal inverse controller is extended to include the coupling due to the nonzero bank angle, but can still take the commanded speed acceleration and pitch angle acceleration and calculate the required thrust and elevator. These two inverse controllers are coupled through yawing and rolling moments due to thrust, which could be used to advantage should engine-only backup control be required. The derivation of both of these inverse controllers is detailed below.

8.1 GENERAL EQUATIONS OF MOTION.

a.   Problem Statement:

One would like to have direct control of

- Flight path angle, γ, via longitudinal stick position
- Bank angle, φ, via lateral stick position
- Flight speed, V_p

Constraint: coordinated maneuvers throughout.

b.   Interpretation of the Problem Statement:

Define: δ_{sx} – longitudinal stick position
δ_{sy} – lateral stick position
δ_{sT} – throttle control position

One would like to have the following relations for control:

$$\gamma = f(\delta_{sx}) = C_1 \delta_{sx} \tag{23}$$
$$\varphi = g(\delta_{sy}) = C_2 \delta_{sy} \tag{24}$$
$$V_p = h(\delta_{sT}) = C_3 \delta_{sT} \tag{25}$$

Therefore, assume that the stick position defines the desired

$\gamma(t)$	$\dot{\gamma}(t)$	$\ddot{\gamma}(t)$	$V_p(t)$
$\varphi(t)$	$\dot{\varphi}(t)$	$\ddot{\varphi}(t)$	$\dot{V}_p(t)$

These may or may not be obtainable in reality because of a number of physical limitations (additional constraints?).

Assume: no such physical limitations yet.

c.   Raw Equations of Motion:

Force equations

$$m\left(\dot{U} + QW - RV\right) = -mg\sin\theta + F_{A_x} + F_{T_x} \tag{26}$$
$$m\left(\dot{V} + RU - PW\right) = +mg\cos\theta\sin\varphi + F_{A_y} + F_{T_y} \tag{27}$$
$$m\left(\dot{W} + PV - QU\right) = +mg\cos\theta\cos\varphi + F_{A_z} + F_{T_z} \tag{28}$$

Moment equations

$$I_{xx}\dot{P} - I_{xz}\dot{R} + QR\left(I_{zz} - I_{yy}\right) - PQI_{xz} = L_A + L_T \tag{29}$$
$$I_{yy}\dot{Q} + PR\left(I_{xx} - I_{zz}\right) + I_{xz}\left(P^2 - R^2\right) = M_A + M_T \tag{30}$$
$$-I_{xz}\dot{P} + I_{zz}\dot{R} + PQ\left(I_{yy} - I_{xx}\right) + I_{xz}QR = N_A + N_T \tag{31}$$

Coupling equations

$$P = \dot{\varphi} - \dot{\psi}\sin\theta \tag{32}$$
$$Q = \dot{\theta}\cos\varphi - \dot{\psi}\cos\theta\sin\varphi \tag{33}$$
$$R = \dot{\psi}\cos\theta\cos\varphi - \dot{\theta}\sin\varphi \tag{34}$$

These equations constitute nine equations involving the following 13 variables

$$U,\ V,\ W,\ P,\ Q,\ R,\ \theta,\ \varphi,\ \psi,\ \delta_e,\ \delta_r,\ \delta_a,\ T$$

The four additional equations needed for unique solution are equations 23 through 25 and the constraint on coordinated maneuvers.

d.   Auxiliary Equations:

From figure 33

$$V_p = \sqrt{U^2 + V^2 + W^2} \tag{35}$$

$$\alpha = \sin^{-1}\left(\frac{W}{V_p\cos\beta}\right) = \sin^{-1}\left(\frac{W}{\sqrt{U^2 + W^2}}\right) = \cos^{-1}\left(\frac{U}{\sqrt{U^2 + W^2}}\right) = \tan^{-1}\left(\frac{W}{U}\right) \tag{36}$$

$$\beta = \sin^{-1}\left(\frac{V}{V_p}\right) = \tan^{-1}\left(\frac{V}{\sqrt{U^2 + W^2}}\right) \tag{37}$$
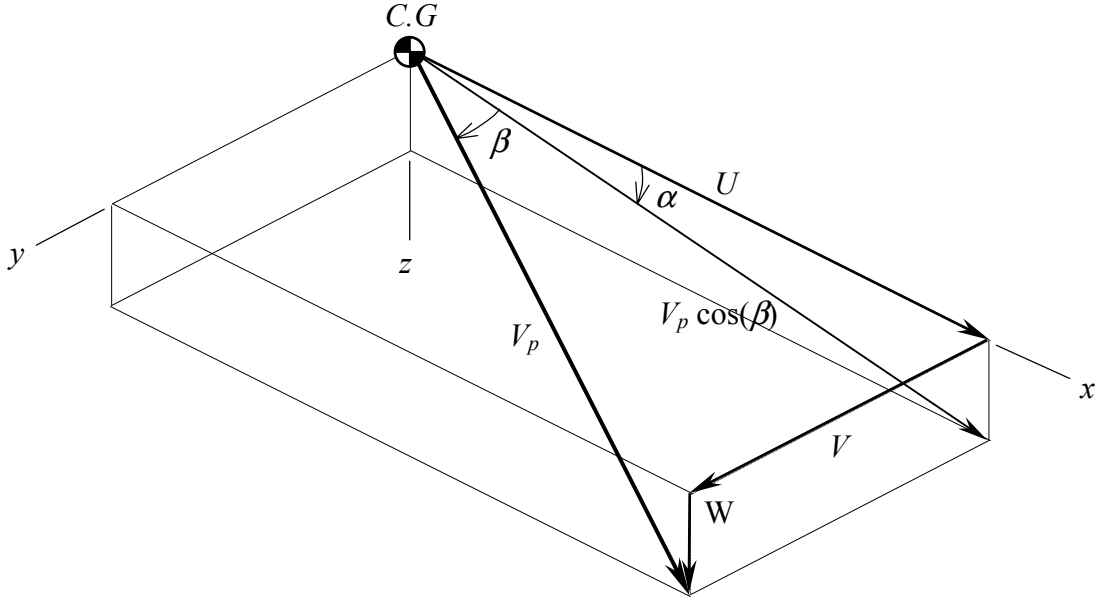
FIGURE 33. KINEMATIC DIAGRAM OF THE VELOCITY VECTOR IN
BODY COORDINATES

e.   Aerodynamic Terms:

Forces

$$F_{A_x} = qS\left(C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\dot\alpha}}\hat{\dot\alpha} + C_{X_q}\hat{Q} + C_{X_{\delta_e}}\delta_e\right) \tag{38}$$

$$F_{A_y} = qS\left(C_{Y_0} + C_{Y_\beta}\beta + C_{Y_r}\hat{R} + C_{Y_p}\hat{P} + C_{Y_{\delta_r}}\delta_r\right) \tag{39}$$

$$F_{A_Z} = qS\left(C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_{\dot\alpha}}\hat{\dot\alpha} + C_{Z_q}\hat{Q} + C_{Z_{\delta_e}}\delta_e\right) \tag{40}$$

Moments

$$L_A = qSb\left(C_{l_0} + C_{l_\beta}\beta + C_{l_r}\hat{R} + C_{l_p}\hat{P} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r\right) \tag{41}$$

$$M_A = qS\bar{c}\left(C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot\alpha}}\hat{\dot\alpha} + C_{m_q}\hat{Q} + C_{m_{\delta_e}}\delta_e\right) \tag{42}$$

$$N_A = qSb\left(C_{n_0} + C_{n_\beta}\beta + C_{n_r}\hat{R} + C_{n_p}\hat{P} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\right) \tag{43}$$

8.2 FORMULATION OF THE INVERSE LATERAL-DIRECTIONAL MOTION.

• In the following development, the right-hand sides are assumed to be known a priori when the equation is marked by (#).

- In this formulation, it is assumed that $\theta(t)$, $\varphi(t)$, and $V_p(t)$ are known.

a. Turn Rate, $\dot\psi$:

For a coordinated maneuver, from the side-force equation, must have

$$m\left(\dot V + RU - PW\right) = +mg\cos\theta\sin\varphi \tag{44}$$

Now from the coupling equations

$$R = \dot\psi\cos\theta\cos\varphi - \dot\theta\sin\varphi \tag{45}$$
$$P = \dot\varphi - \dot\psi\sin\theta \tag{46}$$

substitute into equation 44 and solve for the turn rate

$$\dot\psi = \frac{g\sin(\varphi)\cos(\theta) - \dot V + \dot\theta U\sin(\varphi) + \dot\varphi W}{U\cos(\theta)\cos(\varphi) - W\sin(\theta)} \;\# \tag{47}$$

b. Body Rates, P, Q, R:

These are the coupling equations

$$P = \dot\varphi - \dot\psi\sin\theta \;\# \tag{48}$$
$$Q = \dot\theta\cos\varphi + \dot\psi\cos\theta\sin\varphi \;\# \tag{49}$$
$$R = \dot\psi\cos\theta\cos\varphi - \dot\theta\sin\varphi \;\# \tag{50}$$

c. Lateral-directional controls and side-slip, δ_a, δ_r, and β

From the combination of the equations for side-force, rolling moment, and yawing moment

$$C_l = C_{l_0} + C_{l_\beta}\beta + C_{l_r}\hat R + C_{l_p}\hat P + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r = \left(L_I - L_T\right)/\bar q Sb \tag{51}$$
$$C_n = C_{n_0} + C_{n_\beta}\beta + C_{n_r}\hat R + C_{n_p}\hat P + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r = \left(N_I - N_T\right)/\bar q Sb \tag{52}$$
$$C_Y = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_r}\hat R + C_{Y_p}\hat P + C_{Y_{\delta_r}}\delta_r = -C_{Y_T} \tag{53}$$

where the inertial terms are given as

$$L_I = I_{xx}\dot P - I_{xz}\dot R + QR\left(I_{zz} - I_{yy}\right) - PQI_{xz} \tag{54}$$
$$N_I = -I_{xz}\dot P + I_{zz}\dot R + PQ\left(I_{yy} - I_{xx}\right) + I_{xz}QR \tag{55}$$

41

Casting equations 51 through 53 in matrix form, one can solve for δ_a, δ_r, and β simultaneously.

$$\begin{bmatrix} C_{n_\beta} & C_{n_{\delta_a}} & C_{n_{\delta_r}} \\ C_{l_\beta} & C_{l_{\delta_a}} & C_{l_{\delta_r}} \\ C_{Y_\beta} & C_{Y_{\delta_a}} & C_{Y_{\delta_r}} \end{bmatrix} \begin{Bmatrix} \beta \\ \delta_a \\ \delta_r \end{Bmatrix} = \begin{Bmatrix} (N_I - N_T)/qSb - \left(C_{n_0} + C_{n_r}\hat{R} + C_{n_p}\hat{P} \right) \\ (L_I - L_T)/qSb - \left(C_{l_0} + C_{l_r}\hat{R} + C_{l_p}\hat{P} \right) \\ -\left(C_{Y_0} + C_{Y_r}\hat{R} + C_{Y_p}\hat{P} \right) \end{Bmatrix}$$

(56) #
(57) #
(58) #

d.   Climb Angle, γ.

1.   Definition. The smallest angle between the flight path and the horizontal plane.

2.   Formulation. Therefore, from the kinematic diagram in figure 34

$$\sin(\gamma) = \frac{-dz'/dt}{V_p} = \frac{U}{V_p}\sin(\theta) - \frac{V}{V_p}\sin(\varphi)\cos(\theta) - \frac{W}{V_p}\cos(\varphi)\cos(\theta) \tag{59}$$
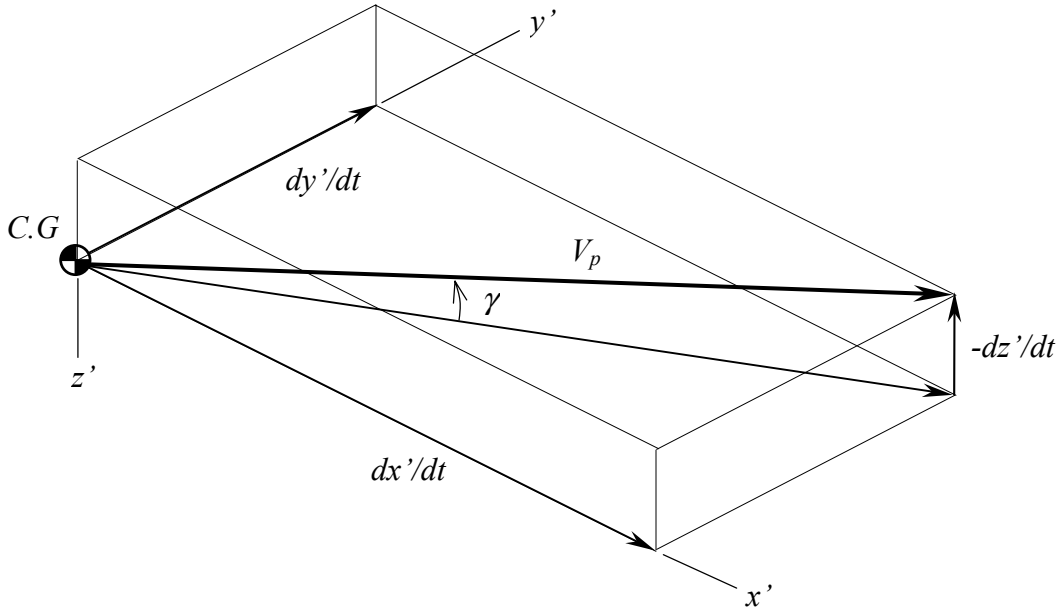
FIGURE 34. KINEMATIC DIAGRAM OF THE VELOCITY VECTOR IN THE INERTIAL REFERENCE FRAME

From the kinematic diagram in figure 29

$$\frac{U}{V_p} = \cos(\alpha)\cos(\beta) \tag{60}$$

$$\frac{V}{V_p} = \sin(\beta) \qquad (61)$$

$$\frac{W}{V_p} = \sin(\alpha)\cos(\beta) \qquad (62)$$

Therefore

$$\sin(\gamma) = \cos(\alpha)\cos(\beta)\sin(\theta) - \sin(\beta)\sin(\varphi)\cos(\theta) - \sin(\alpha)\cos(\beta)\cos(\varphi)\cos(\theta) \qquad (63)$$

3. Checking:

- Wings-Level

$$\varphi = 0 \quad \Rightarrow \quad \sin(\gamma) = \cos(\beta)[\cos(\alpha)\sin(\theta) - \sin(\alpha)\cos(\theta)]$$

Therefore

$$\sin(\gamma) = \cos(\beta)\sin(\theta - \alpha)$$

And if there is no side-slip

$$\beta = 0 \quad \Rightarrow \quad \gamma = \theta - \alpha$$

This is the same as that for the longitudinal case.

- Knife-Edge Turn

$$\varphi = \pi/2 \quad \Rightarrow \quad \alpha = 0?$$

Therefore

$$\sin(\gamma) = \cos(\beta)\sin(\theta) - \sin(\beta)\cos(\theta)$$

Resulting in

$$\gamma = \theta - \beta$$

This also checks.

e.   Angle of Attack, α.

Following the same procedure as in the longitudinal case, one starts with

$$m(\dot{U} + QW - RV) = -mg\sin\theta + F_{A_x} + F_{T_x} \tag{64}$$

$$m(\dot{V} + RU - PW) = +mg\cos\theta\sin\varphi + F_{A_y} + F_{T_y} \tag{65}$$

$$m(\dot{W} + PV - QU) = +mg\cos\theta\cos\varphi + F_{A_z} + F_{T_z} \tag{66}$$

Multiply equation 64 by $W/(U^2 + W^2)$, equation 66 by $U/(U^2 + W^2)$, and divide through by the mass

$$\frac{W}{U^2 + W^2}(\dot{U} + QW - RV) = \frac{W}{U^2 + W^2}\left[-g\sin\theta + \frac{F_{A_x} + F_{T_x}}{m}\right] \tag{67}$$

$$\frac{U}{U^2 + W^2}(\dot{W} + PV - QU) = \frac{U}{U^2 + W^2}\left[+g\cos\theta\cos\varphi + \frac{F_{A_z} + F_{T_z}}{m}\right] \tag{68}$$

Subtract equations 67 from equation 68

$$\frac{U\dot{W} - W\dot{U}}{U^2 + W^2} + \frac{V}{U^2 + W^2}(PU + RW) - \frac{Q}{U^2 + W^2}(U^2 + W^2)$$

$$= \frac{g}{U^2 + W^2}[U\cos(\theta)\cos(\varphi) + W\sin(\theta)] + \frac{(F_{A_z} + F_{T_z})U}{m(U^2 + W^2)} - \frac{(F_{A_x} + F_{T_x})W}{m(U^2 + W^2)} \tag{69}$$

From the kinematic diagram

$$\alpha = \tan^{-1}\left(\frac{W}{U}\right) \tag{70}$$

Therefore

$$\dot{\alpha} = \frac{\dot{W}U - \dot{U}W}{U^2 + W^2} \tag{71}$$

Also

$$\frac{V}{\sqrt{U^2 + W^2}} = \tan(\beta) \tag{72}$$

$$\frac{W}{\sqrt{U^2 + W^2}} = \sin(\alpha) \tag{73}$$

$$\frac{U}{\sqrt{U^2 + W^2}} = \cos(\alpha) \tag{74}$$

Furthermore, for reasonably small side-slip angle, β

$$\sqrt{U^2 + W^2} \approx \sqrt{U^2 + W^2 + V^2} = V_p \tag{75}$$

Therefore, noting equation 72 and considering the rest of equation 69 term by term

$$\frac{V}{U^2 + W^2}(PU + RW) = \tan(\beta)[P\cos(\alpha) + R\sin(\alpha)] = P_V \tan(\beta) \tag{76}$$

where P_V is the roll rate about the velocity vector.

$$\frac{g}{U^2 + W^2}[U\cos(\theta)\cos(\varphi) + W\sin(\theta)] = \frac{g}{V_p}[\cos(\alpha)\cos(\theta)\cos(\varphi) + \sin(\alpha)\sin(\varphi)] \tag{77}$$

$$\frac{F_{A_z}U - F_{A_x}W}{m(U^2 + W^2)} \approx \frac{qS}{mV_p}\left[C_{A_z}\cos(\alpha) - C_{A_x}\sin(\alpha)\right] = \frac{-qS}{mV_p}C_L \tag{78}$$

$$\frac{F_{T_z}U - F_{T_x}W}{m(U^2 + W^2)} \approx \frac{T}{mV_p}[\sin(\varphi_T)\cos(\alpha) - \cos(\varphi_T)\sin(\alpha)] = \frac{qS}{mV_p}C_T\sin(\varphi_T - \alpha) \tag{79}$$

Substituting these into equation 69 results in

$$\dot{\alpha} \approx -P_V\tan(\beta) + Q + \frac{g}{V_p}[\cos(\alpha)\cos(\theta)\cos(\varphi) + \sin(\alpha)\sin(\varphi)] \\ + \frac{qS}{mV_p}[C_L + C_T\sin(\varphi_T - \alpha)] \qquad\qquad \# \tag{80}$$

Again, $C_T\sin(\varphi_T - \alpha)$ is probably negligible.

f.   Airspeed, V_p:

Multiply equation 64 by U/V_p, equation 65 by V/V_p, equation 66 by W/V_p, and add the three equations

45

$$\frac{m}{V_p}\left(U\dot{U}+V\dot{V}+W\dot{W}\right)=\frac{mg}{V_p}\left[-U\sin(\theta)+V\sin(\varphi)\cos(\theta)+W\cos(\varphi)\cos(\theta)\right]$$
$$+\frac{\left(F_{A_x}+F_{T_x}\right)U}{V_p}+\frac{\left(F_{A_y}+F_{T_y}\right)V}{V_p}+\frac{\left(F_{A_z}+F_{T_z}\right)W}{V_p} \tag{81}$$

But

$$\frac{\left(U\dot{U}+V\dot{V}+W\dot{W}\right)}{V_p}=\dot{V}_p \tag{82}$$

and

$$\frac{1}{V_p}\left[-U\sin(\theta)+V\sin(\varphi)\cos(\theta)+W\cos(\varphi)\cos(\theta)\right]=\sin(\gamma) \tag{83}$$

so equation 81 reduces to

$$\dot{V}_p=g\sin(\gamma)+\frac{qS}{m}\left[\begin{array}{l}\left(C_{A_x}+C_{T_X}\right)\cos(\alpha)\cos(\beta)+\\\left(C_{A_y}+C_{T_y}\right)\sin(\beta)+\left(C_{A_z}+C_{T_z}\right)\sin(\alpha)\cos(\beta)\end{array}\right] \tag{84}$$

But also

$$C_{A_x}\cos(\alpha)\cos(\beta)+C_{A_z}\sin(\alpha)\cos(\beta)=-C_D\cos(\beta) \tag{85}$$

$$C_{T_x}\cos(\alpha)\cos(\beta)+C_{T_z}\sin(\alpha)\cos(\beta)=C_T\left[\cos(\varphi_T)\cos(\alpha)+\sin(\varphi_T)\sin(\alpha)\right]\cos(\beta)$$
$$=C_T\cos(\varphi_T-\alpha)\cos(\beta) \tag{86}$$

Therefore

$$\dot{V}_p=g\sin(\gamma)+\frac{qS}{m}\left[\left(C_T\cos(\varphi_T-\alpha)-C_D\right)\cos(\beta)+\left(C_{A_y}+C_{T_y}\right)\sin(\beta)\right]\# \tag{87}$$

g.   Elevator Deflection Angle, δ_e

From the pitching-moment equation

$$\dot{Q}+PR\left(\frac{I_{xx}-I_{zz}}{I_{yy}}\right)+\frac{I_{xz}}{I_{yy}}\left(P^2-R^2\right)=\frac{qS\overline{c}}{I_{yy}}\left(C_m+C_T\frac{d_T}{\overline{c}}\right)\# \tag{88}$$

46

where

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot\alpha}}\hat{\dot\alpha} + C_{m_q}\hat{Q} + C_{m_{\delta_e}}\delta_e \tag{89}$$

9. RESULTS AND FURTHER WORK.

Proprietary geometry and force and moment data for the Raytheon Beech Bonanza F33C were used to develop longitudinal and full 6 degree-of-freedom MATLAB Simulink dynamic models of the aircraft. This data was also used to develop a fixed base visual flight simulation model in the x-plane flight simulation software [15]. An inverse controller was formulated for velocity vector control in the longitudinal case. Pilot-commanded speed and flight path angle were tracked by the controller. It was verified that this multi-input multi-output controller did function properly when there was no modeling error between the aircraft and the model used in the inverse controller. Neural networks were then added to the controller to accommodate modeling errors. Various modeling errors were then introduced by simulating configuration changes and failure modes of the aircraft. The presented artificial neural network (ANN) approach was proven to be very effective in designing an adaptive inverse controller. A change in pitching-moment coefficient with angle of attack (C_{m_α}) of up to 40%, loss of 25% engine thrust, and even the loss of 50% of the elevator effectiveness were successfully compensated for by the neural networks and would allow the pilot to continue safely controlling the aircraft without any adjustments. The inverse controller theory was extended to include lateral-directional flight. A review of existing military developments and standards on advanced flight control systems was reported, and certification issues that should be examined were discussed, which were related to code generation from MATLAB Simulink and possible tool qualification. Methods to dump the controller code from MATLAB, which is compatible with the Raytheon Beech Bonanza F33C aircraft flight computer, were tested.

Performing the tasks stated earlier in this report revealed the necessity for further investigation in several areas. These tasks can be divided into two categories: (1) the issues involving modeling and control system architecture and (2) matters pertaining to certification and developing and evaluating the safety enhancements of this type of flight control system.

In the first category, the present MATLAB Simulink controller should be extended to include the lateral-directional modes of motion as well as the longitudinal flight, to cover the two predominant modes of motion in aircraft. Furthermore, the system's response to external disturbances, including wind shear and atmospheric turbulence, needs to be investigated because it is critical to assess the robustness of the control algorithm in the presence of random external inputs. In terms of the ANN application, issues of network windup and pseudocontrol hedging need to be addressed since these occur when a control is at its limit. Also, alternate neural network architectures should be examined to verify that the Calise method is best for this application. The models and control architectures need to be verified in simulation first and then flight tested to assess their performance in real-life situations. Of course simulation prior to flight test is absolutely necessary to alleviate the risks associated with the latter.

Regarding safety, methods need to be developed for assigning control effort to multiple control surfaces that control the same flight variable, i.e., flaps, elevator, trim tab, and thrust for pitch

control. Also relevant to safety and certification are the identification of the critical issues for implementation of the software in flight testing and how nonlinear adaptive control affects the required level of redundancy in design and safety features of the flight control system. Boundaries need to be established that would define levels of failure that can be tolerated by this system in order to evaluate the safety benefits of the adaptive flight control system. Investigation of these factors can result in proposed changes to DO-178B relevant to Title 14 Code of Federal Regulations Part 23 regarding safety benefits of advanced flight control systems across the fleet as opposed to individual hardware part reliability.

It would be useful to interface the work at Wichita State University on MATLAB Control systems with Federal Aviation Administration-sponsored work on qualification of software tools, specifically to investigate techniques for tool qualification as applied to MATLAB Simulink as a software development tool and how MATLAB tool qualification can be applied to the advanced flight controls project. Also, it would be useful to look at how the software generated by MATLAB real-time workshop could be certified and then look at how this process can be applied to the advanced flight controls project.

10. REFERENCES.

1. Brinker, J. and Wise, K., "Stability and Flying Qualities Robustness of a Dynamic Inversion Aircraft Control Law," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, November-December 1996, pp. 1270-1277.

2. McFarland, M. and Calise, A., "Robust Adaptive Control of Uncertain Nonlinear Systems Using Neural Networks," *IEEE Transactions on Automatic Control*, July 1997, Short Paper.

3. McFarland, M. and Calise, A., "Adaptive Nonlinear Control of Agile Antiair Missiles Using Neural Networks," *IEEE Transactions on Control Systems Technology*, Vol. 8, No. 5, September 2000, pp. 749-756.

4. Rysdyk, R., Nardi, F., and Calise, A., "Robust Adaptive Nonlinear Flight Control Applications Using Neural Networks," *Proceedings of the American Control Conference*, pp. 2595-2599.

5. Rysdyk, R., "Adaptive Nonlinear Flight Control," PhD thesis, Georgia Institute of Technology, November 1998.

6. Rysdyk, R., Calise, A., and Chen, R., "Nonlinear Adaptive Control of Tiltrotor Aircraft Using Neural Networks," World Aviation Congress, Anaheim, CA, Paper 975613.

7. Rysdyk, R. and Agarwal, R., "Nonlinear Adaptive Flight Path and Speed Control Using Energy Principles," AIAA 2002-4440, August 2002.

8. Soloway, D. and Haley, P., "Aircraft Reconfiguration Using Neural Generalized Predictive Control," *American Control Conference Proceedings*, Vol. 4, pp. 2924 -2929.

9.   Kim, B. and Calise, A., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, January-February 1997, pp. 26-33.

10.   Lewis, F., Liu, K., and Yesildirek, A., "Neural Net Robot Controller With Guaranteed Tracking Performance," *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, May 1995, pp. 703-715.

11.   Rolf Rysdyk, "Simulation and Flight Test Assessment of Safety Benefits and Certification Aspects of Advanced Flight Control Systems," for the period September 15, 2000-August 31, 2001, University of Washington, Seattle, WA, Interim Report to the FAA William J. Hughes Technical Center, December 2001.

12.   Roskam, J., "Airplane Flight Dynamics and Automatic Flight Controls, Part I," DARcorporation, Lawrence, KS, 3rd ed., 2001, ISBN 1-884885-17-9.

13.   Lambregts, T., "Vertical Flight Path and Speed Autopilot Design Using Total Energy Principles," AIAA 83-2239CP, August 1983.

14.   Lambregts, T., "Operational Aspects of the Integrated Vertical Flight Path and Speed Control System," SAE Paper 831420, October 1983.

15.   http://www.x-plane.com/   x-plane is the world's most comprehensive, powerful flight simulator, and has the most realistic flight model available for personal computers.