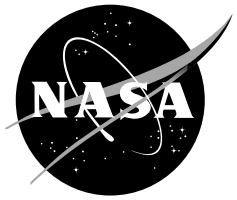
NASA/TM-2003-212028



Integration of Online Parameter Identification and Neural Network for In-Flight Adaptive Control

Jacob J. Hageman, Mark S. Smith, and Susan Stachowiak
NASA Dryden Flight Research Center
Edwards, California

October 2003

The NASA STI Program Office…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results…even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

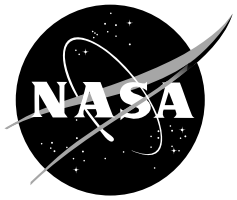
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2003-212028



Integration of Online Parameter Identification and Neural Network for In-Flight Adaptive Control

Jacob J. Hageman, Mark S. Smith, and Susan Stachowiak
NASA Dryden Flight Research Center
Edwards, California

National Aeronautics and
Space Administration

Dryden Flight Research Center
Edwards, California 93523-0273

October 2003

NOTICE

Use of trade names or names of manufacturers in this document does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

ABSTRACT

An indirect adaptive system has been constructed for robust control of an aircraft with uncertain aerodynamic characteristics. This system consists of a multilayer perceptron pre-trained neural network, online stability and control derivative identification, a dynamic cell structure online learning neural network, and a model following control system based on the stochastic optimal feedforward and feedback technique. The pre-trained neural network and model following control system have been flight-tested, but the online parameter identification and online learning neural network are new additions used for in-flight adaptation of the control system model. A description of the modification and integration of these two stand-alone software packages into the complete system in preparation for initial flight tests is presented. Open-loop results using both simulation and flight data, as well as closed-loop performance of the complete system in a nonlinear, six-degree-of-freedom, flight validated simulation, are analyzed. Results show that this online learning system, in contrast to the nonlearning system, has the ability to adapt to changes in aerodynamic characteristics in a real-time, closed-loop, piloted simulation, resulting in improved flying qualities.

NOMENCLATURE

C_{m_α}	coefficient of pitching moment due to angle of attack
DCS	dynamic cell structure neural network
IFCS	Intelligent Flight Control System
PID	real-time parameter identification
PTNN	pre-trained neural network
SOFFT	stochastic optimal feedforward and feedback technique

INTRODUCTION

Advances in computational power and aircraft parameter estimation techniques have introduced the opportunity for real-time online estimation of aerodynamic stability and control derivatives for use in adaptive control. The Intelligent Flight Control System (IFCS) project,[1,2,3] combining the efforts of NASA, academia, and industry, has proposed a set of control systems, estimation algorithms, and neural networks for flight-testing in an effort to show real-time adaptation, allowing for safe, predictable control of an aircraft with uncertain aerodynamic stability and control derivatives.

The software suite currently being tested is comprised of a pre-trained neural network[4] (PTNN) for calculating the baseline aerodynamic stability and control derivatives, and real-time parameter identification[5,6] (PID) paired with a dynamic cell structure neural network[7,8](DCS). These routines provide near real-time estimates of the current aircraft derivatives to the research flight control system. The control system is based on the stochastic optimal feedforward and feedback technique[9,10] (SOFFT). The complete system is categorized as an indirect adaptive system, because adaptation modifies the aircraft plant model residing in the SOFFT controller, where a direct adaptive system uses feedback signals to generate command augmentation signals. The SOFFT controller was flight-tested in 1999

using only the baseline derivatives from the PTNN. The PID and DCS software packages are new additions to this system.

This report describes the progression of the stand-alone PID and DCS algorithms into a complete system, including preflight integration and testing with extensive simulation use, and final preparation for open-loop in-flight testing. The history of required modifications and enhancements for the PID and DCS are summarized, in addition to intermediate results and background discussions regarding the tradeoffs and configuration choices. Open-loop results using both simulation and flight data and closed-loop performance of the complete system in a nonlinear, six-degree-of-freedom simulation are presented.

BACKGROUND

The IFCS project goal is to develop neural network–based adaptive control systems for flight. A highly modified preproduction F-15B aircraft (Boeing Company, St. Louis, Missouri) with canards, axisymmetric thrust-vectoring pitch and yaw balance beam nozzles, and a full authority digital flight control system was chosen as the test vehicle. For this project, the canards are used for control and assistance with simulating failures, but the vectoring nozzles are not used. An updated avionics package has been added to the aircraft to supply required aircraft parameters, and the experimental software is run in a new airborne research test system known as the ARTS II. In figure 1, the dark lines represent the systems configuration of the aircraft in 1999, and the light lines represent new additions to the system.

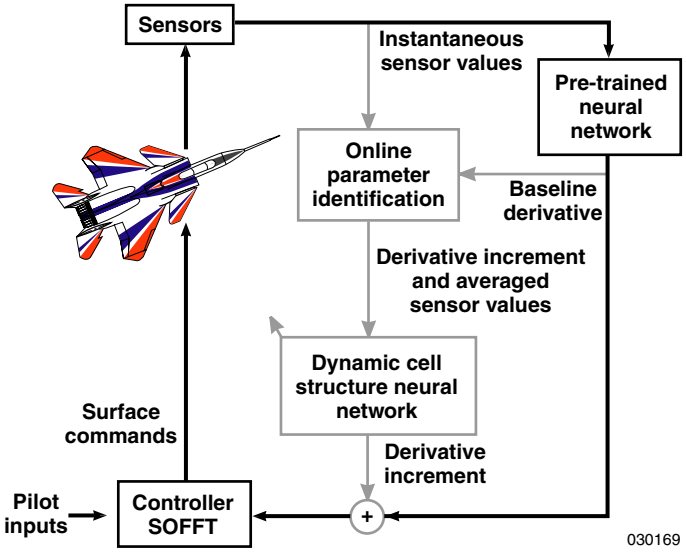


Figure 1. System configuration.

In the system shown in figure 1, the PTNN provides the wind-tunnel derived prediction of 26 stability and control derivatives based on flight condition and surface positions. Training is performed on the ground before flight, and the algorithm has no online learning capability. The PTNN is classified as a multilayer perceptron (MLP) neural network. The training method uses a trust region variation of the Levenberg-Marquardt algorithm to optimize the network weights for a minimum mean square error. This method has been used at the NASA Ames Research Center (Moffett Field, California) for prediction of aircraft coefficients during wind-tunnel operations.[1]

The SOFFT controller uses 26 stability and control derivatives with the aircraft states and pilot inputs to calculate surface commands. The main feature of the SOFFT control design methodology is the separation of the feedforward and feedback portions of the controller to allow fast and smooth tracking response while suppressing noise and disturbances through the use of two different cost functions.[9,10] The feedforward part of the SOFFT controller uses a pilot selectable command model for achieving specific handling characteristics (that is, frequency and damping). The feedback portion uses an online Riccati solver for stabilization and disturbance rejection. Both the feedforward and feedback portions of the controller rely on aircraft plant models formulated from aerodynamic stability and control derivatives. The ability to modify the aircraft plant model in flight in near real time using updated aerodynamic stability and control derivatives enables this controller to indirectly adapt to uncertain aerodynamics.[6]

For the 1999 flights, to show the validity of using neural networks in flight, the PTNN derivatives were fed into the SOFFT controller to update the aircraft plant model depending on flight conditions. At this stage no online learning of the neural network existed; the system would only recall estimates of the derivatives from the wind-tunnel data on which the PTNN had been trained. This system performed very well in the lateral axis where the commanded roll rate and actual roll rate very closely matched. In comparing the conventional control system to the PTNN and SOFFT controller, pilots commented that the new controller had similar or slightly improved handling qualities characteristics.[11,12] Pilots noted degraded handling qualities in the pitch axis and commented that the commands did not track actual rates as well as they did in the lateral axis. The suspected cause of this degradation is a difference between the original wind-tunnel model estimate of C_{m_α}, based on a previous rectangular engine nozzle design used to train the PTNN, and the current configuration of the aircraft that uses round engine nozzles.

During the phase of flights completed in the spring of 2003, data were gathered for offline testing of the parameter estimation and learning neural network. In the next phase of flights, the aerodynamic stability and control derivatives will be estimated online in real time. The differences between the online estimates and the PTNN estimates will be stored as a function of flight condition with another neural network. This online learning portion of the system will be run open loop on the aircraft, allowing for an initial comparison of the derivatives with the PTNN and postflight analysis routines before closing the loop. In the subsequent flight phase, the online estimates will be used to update the aircraft plant model in the SOFFT controller, thereby closing the loop. At this stage, the ability of the entire system to actively adapt to changing aerodynamic stability and control derivatives can be tested and analyzed.

Identification of the in-flight aircraft stability and control parameters is accomplished using PID, which involves correlating aircraft responses to control surface inputs. Various techniques have been used postflight to calculate the aerodynamic stability and control derivatives, but the IFCS concept requires near real-time parameter identification during flight. The PID technique used for this study is a real-time equation error method that operates in the frequency domain called Fourier transform

regression (FTR).[5] Modifications were made to the original technique for use in this study, including a reduction in the number of calculated derivatives and operation on a time-based window of data.[6]

A few properties of the PID algorithm must be considered. First, the PID has no long-term storage of estimated derivatives, because it works only on a window of recent data. Another result of working on a window of recent data is that estimated derivatives are not valid for the current flight condition. Instead, the estimates are correlated to averaged flight conditions over the data window. Also, the PID is only able to compute valid estimates with sufficient system excitation, so the estimates should not be used under certain conditions. For long-term storage and to fill in the gaps where PID results are not valid, the DCS has been added to the system to learn the estimated derivative corrections and recall instantaneous values for use in the SOFFT controller.

The DCS belongs to the class of topology representing networks with mixed unit insertions depending on approximation error and stimulus distance. A competitive, nonsymmetric Hebbian rule is used for topology learning, and an error-modulated Kohonen rule is used for center adaptation, in which the goal is a uniform distribution of resource values.[7] The DCS has been used in a direct adaptive aircraft control method with favorable piloted simulation results.[8]

The simulation data shown in this report come from a six-degree-of-freedom, flight validated, nonlinear simulation using modified second-order Runge-Kutta integration.[13] Oblate Earth equations are used with a gravity model based on the Goddard Earth Model-10 and a standard day atmospheric model. Fixed-base, real-time, piloted flight and batch operation modes are available, and both are used for validation and verification of the research systems.

REAL-TIME PARAMETER IDENTIFICATION (PID)

A major goal for using PID in a real-time flight environment is to be able to calculate aerodynamic derivative information using typical pilot inputs and still obtain rapid convergence. As a result of using pilot inputs, enough system excitation might not always exist to calculate accurate answers, or calculated errors might be large, thus a measure of confidence in the derivatives is needed. Furthermore, the derivatives must be updated with respect to flight condition and aircraft configuration changes.

Postflight parameter identification methods can be used to calculate the same 26 aerodynamic stability and control derivatives estimated by the PTNN using independent stacked sine inputs to the surfaces to decouple the dynamic effects.[14] Canards are symmetrically scheduled with angle of attack, and multiple surfaces are used for roll and yaw commands. These surface correlations cause difficulty in calculating independent surface contributions. To account for this difficulty, the number of derivatives estimated by the online PID was reduced to 13, and the other 13 derivatives are assumed known from the PTNN. This assumption creates an undesirable dependence on the PTNN for the individual derivative results, but because the focus is on the complete model using coordinated commands, the final result is acceptable.

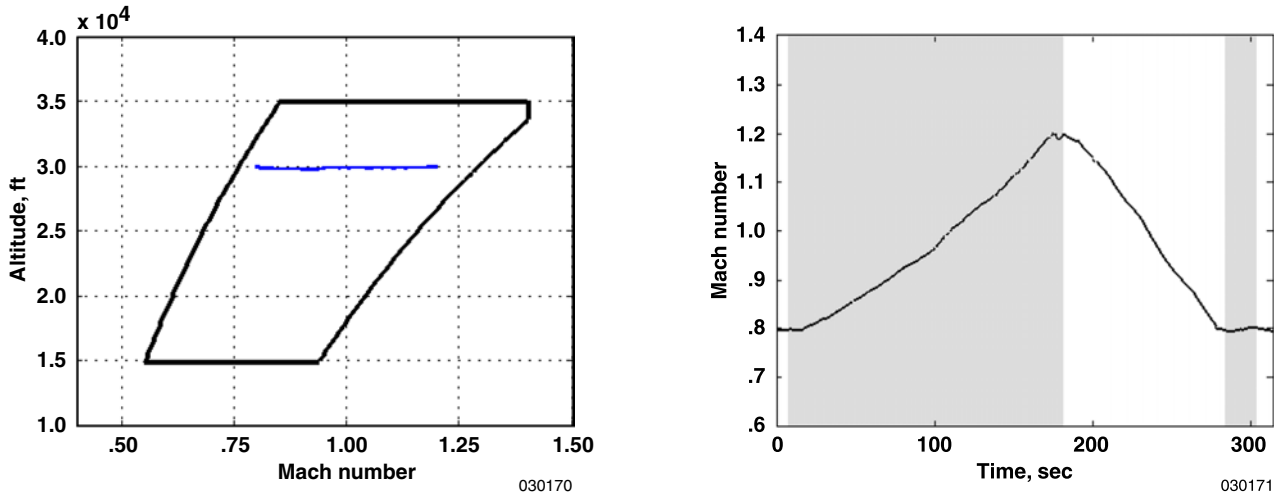
The postflight version of the PID also works on the complete time history of data given to the system. To estimate derivatives for a specific flight condition, only that section of data would be given to the PID. Single averaged derivatives for that section of data would be output, whereas for the real-time system, derivatives must be updated as the flight condition changes. To perform the real-time updates, an adjustable data buffer stores the recent input information for the PID calculations. The lower size limit of

this buffer is driven by the amount of information needed for the PID to converge. The upper limit affects the response time of the PID and the ability of the PID to converge because of changing conditions. For all results presented in this report, a 10-second data buffer is used. Because the derivatives are actually an average over the data in the buffer, the PID also averages the flight conditions to use as independent inputs for DCS training. Because the PID is always running while the research system is active on the aircraft, the inputs are tested and limited to avoid numerical errors in the matrix inversion routines of the algorithm.

To determine the validity of the PID estimates for the current conditions, four types of tests are performed. The first two tests measure the standard and relative error calculated for each derivative. The third test checks for a sufficient amount of system excitation per axis. The fourth and final test requires that the system pass the standard error test for a specific number of frames before a derivative is considered valid for DCS training.[5]

For the flight test, many parameters that control the various functions of the PID are read from a configuration file, allowing for tuning of the algorithm between flights without requiring software recompilation. Data buffer size, validity test limits, filter, and calibration values are variables included in the configuration file.

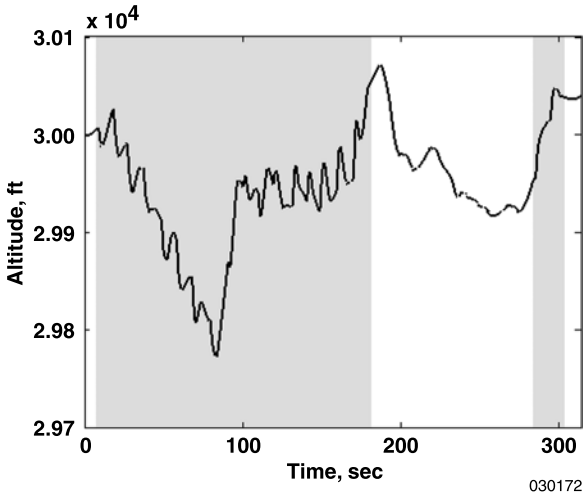
Figure 2 shows the F-15 IFCS piloted simulation flight profile used for open-loop PID testing. The flight consisted of an acceleration with pitch, roll, and yaw doublets approximately every 7 seconds followed by a smooth deceleration and three doublets at the end, all at a constant altitude. The Mach number ranged from 0.8 to 1.2, spanning the experimental envelope with the acceleration occurring in the first 175 seconds, followed by a deceleration back to initial conditions. For the plots presented here, highlighted portions of the profile indicate when the system was being excited by pilot inputs.



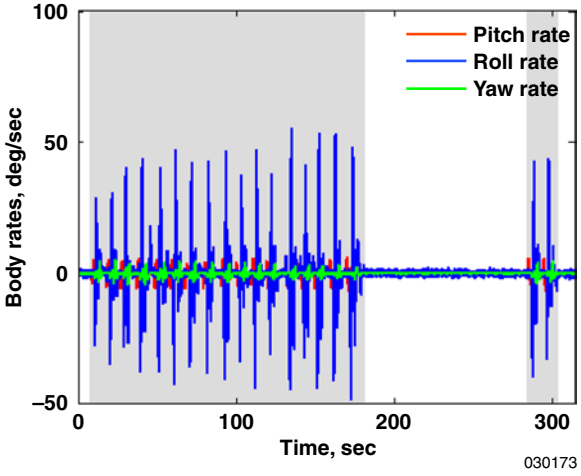
(a) Flight profile in experimental envelope.

(b) Flight profile Mach number.

Figure 2. Simulation open-loop flight profile.



(c) Flight profile altitude.



(d) System excitation.

Figure 2. Concluded.

Data from this simulation are used to calculate the full 26 PTNN and 13 PID estimates. Figure 3 shows the PID estimate of the coefficient of pitching moment due to angle of attack, C_{m_α}, chosen because of the interesting dependence on Mach number. For all the simulation data plots, the PID is started from an initial state with an empty data buffer.

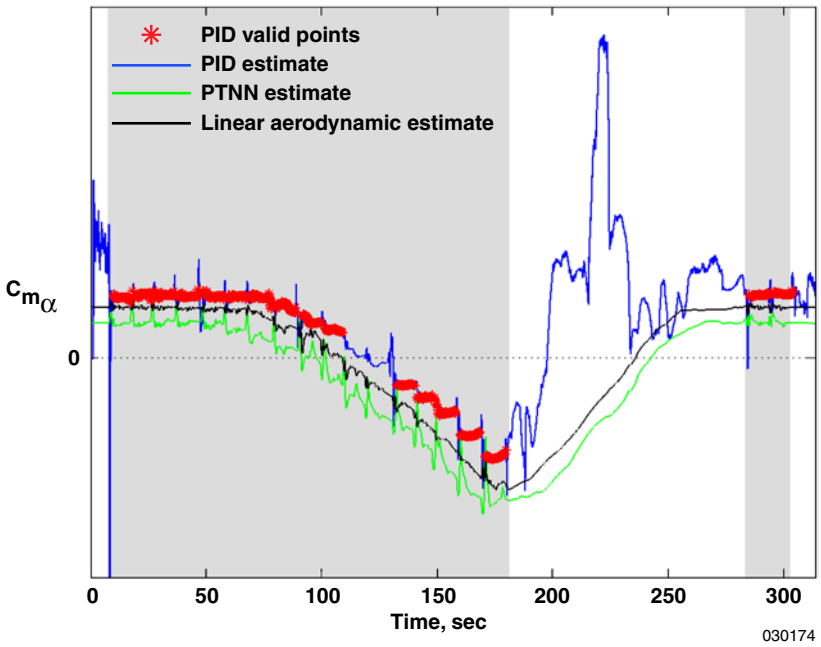


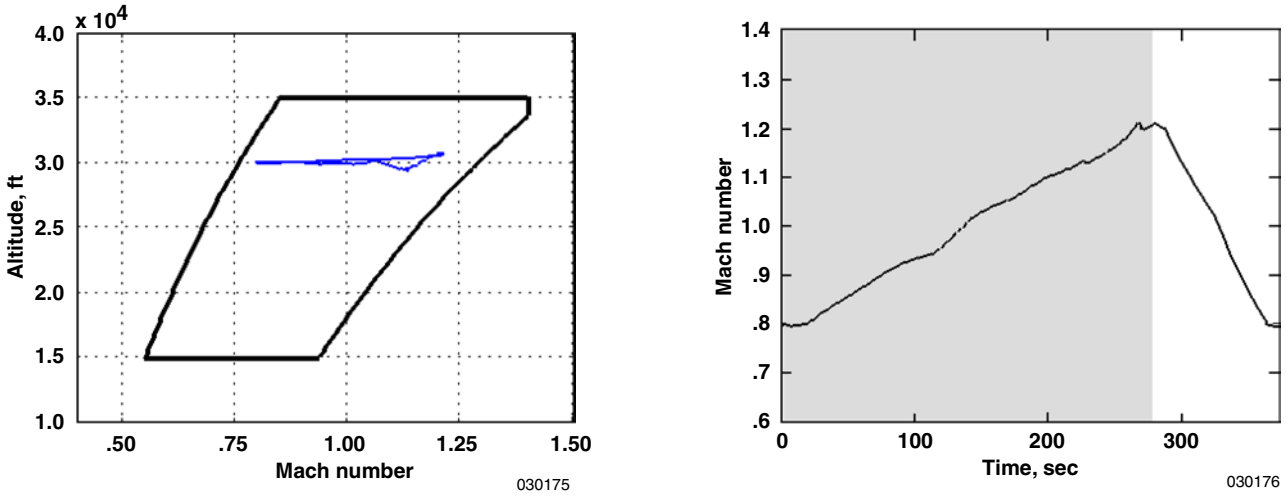
Figure 3. Simulation open-loop PID results.

The linear aerodynamic estimation shown in figure 3 is calculated using the aerodynamic model in the simulation and perturbing it to find the linear derivative for the current condition. This linear aerodynamic estimation is used as a comparison model.

As expected, the PID, PTNN, and linear aerodynamic estimates are very similar for this configuration, because the aircraft is in a healthy unmodified configuration. The pitch, roll, and yaw doublets during the acceleration and at the end of the run excite the system, allowing the PID to estimate the derivative with high confidence, marked by the PID valid points. During deceleration, the lack of inputs cause the PID estimate to become erratic and fail the validity tests. Because of the multiaxis pilot inputs, these results are common across all 13 of the PID-estimated derivatives.

For the simulation data presented in figure 4, an increment dependent on Mach number has been added to C_{m_α} in the aerodynamic model to simulate a poorly modeled, uncertain, or damage-influenced derivative. A similar profile is shown with a piloted acceleration during the first 270 seconds containing pitch, roll, and yaw doublets approximately every 10 seconds, and a smooth deceleration without the doublets at the end, all at a constant altitude.

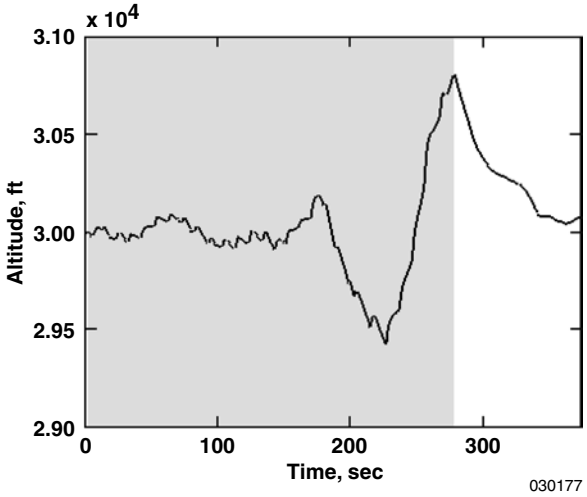
As figure 5 shows, the PID estimate again is very close to the linear aerodynamic estimate with the modified aerodynamic characteristics, but because the PTNN is not an online learning system, it estimates the original unmodified derivative. For the control system flight-tested in 1999, the PTNN estimates were the only available information, and an aircraft modification such as this would cause degraded and possibly unstable flight characteristics. With the addition of the DCS, which acts as a long-term storage mechanism, the valid PID results can be saved and used to correct the PTNN values as derivative inputs to the SOFFT.



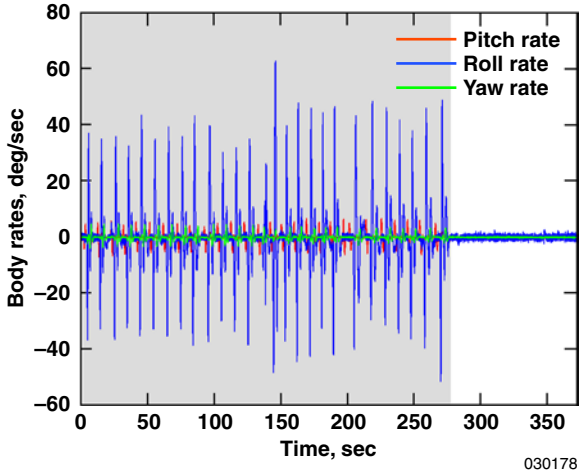
(a) Flight profile in experimental envelope.

(b) Flight profile Mach number.

Figure 4. Simulation open-loop flight profile with modified aerodynamics.



(c) Flight profile altitude.



(d) System excitation.

Figure 4. Concluded.

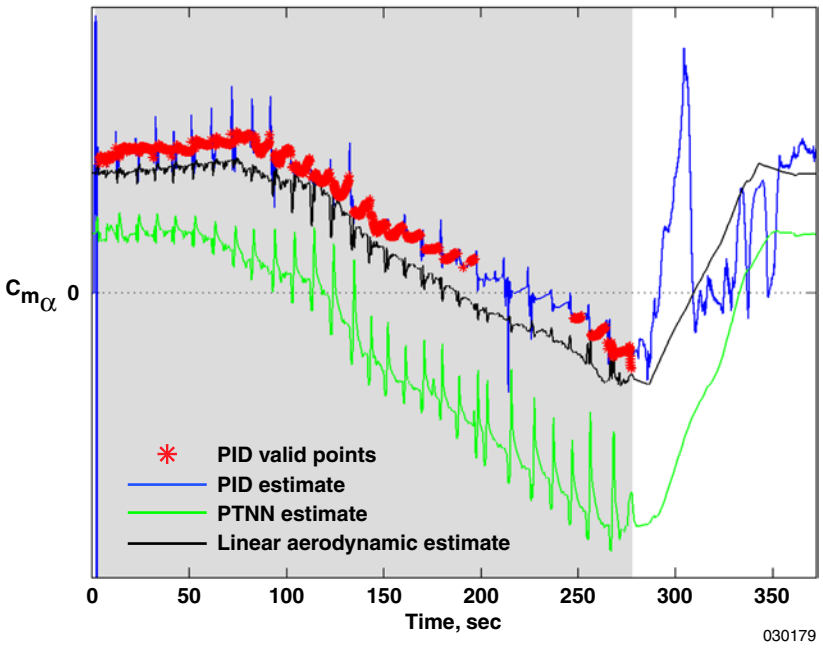


Figure 5. Simulation open-loop PID results with modified aerodynamics.

DYNAMIC CELL STRUCTURE NEURAL NETWORK (DCS)

The role of the DCS is to use valid information from the PID to store and recall an estimate of the derivative increment based on the current flight conditions. The derivative increment learned by the DCS is the PID estimate minus an averaged PTNN estimate over the same PID window of data. The flight conditions and surface positions are average values corresponding to the window of data used by the PID to calculate the derivative estimate. When information is recalled from the DCS, flight conditions and surface positions are used as inputs, and the outputs are added to the instantaneous PTNN estimates and fed to the SOFFT controller. These corrected derivatives are used to update the model that is used by the controller to calculate surface gains.

To use the DCS as the online learning mechanism in this system, many modifications to the generic algorithm were necessary. The original DCS contained all 26 derivative increments in 1 very large network, resulting in extensive training time and undesirable interaction between different axes. An individual network for each derivative increment was deemed too computationally expensive, so the final configuration involved grouping the derivative increments by axis (the three moments, side force, and normal force) for a total of five networks. Creating additional difficulty is the fact that validity is independently calculated for each derivative increment, and all increments in an axis are not always valid at the same instant. Because training is important whenever a derivative increment is valid, if other derivative increments in that axis are invalid, they are marked as unknown so they will not adversely effect the network. All calculations are done on the current valid derivative increments only, and if none exist for that axis, no training is performed.

A very important ability of the network is to scale both the independent and dependent inputs before calculations are done. This ability allows for adjusting the importance of inputs for determination of errors, node addition location, and recall of derivative increments from the network. Typically, Mach number is scaled to be the most important, followed by angle of attack, with sideslip, altitude, and surface positions of lesser importance. The typical topology of the network mainly depends on the independent inputs as in a typical aerodynamic table formulation, but more nodes are also desirable in areas where the derivative increments are rapidly changing as in the transonic range. The scaling of the derivative increments is an involved process and very dependent on the input data used. Fortunately, like the PID, the DCS has a configuration file to adjust many of the important parameters including the scaling between flights. An initial set of scaling values has been chosen, but when in-flight testing begins, updates are expected for optimal performance.

Included in the configuration file are other values that can be independently modified per axis and affect the internal operation of the DCS. The way data are inserted into the training set and criteria for what data are inserted is controlled by two of the values in the configuration file. The error level for adding a new node, edge forgetting behavior, and amount of adaptation for best matching and neighboring nodes are also included. These values control the topology, size, speed, adaptability, and long-term storage behavior of the neural network.

All the network data are periodically saved during testing. Between runs the network can be reset to a zero condition, restored to a previously saved network, or allowed to continue training. Currently, when training occurs from a zero condition, and the first derivative increment is valid, all zero nodes are set to these initial values. For any derivative increments that are not valid, all nodes receive that increment set as soon as validity is achieved. This method decreases the initial response time of the DCS and eliminates

the condition of having nodes near the untrained zero condition. Before this modification was implemented, the recall data would occasionally jump to the insufficiently trained nodes, causing unfavorable results.

Figure 6 shows the performance of the network using an initial configuration file. The C_{m_α} increment is shown for the modified aerodynamic model at the same acceleration, deceleration, and constant altitude shown in figure 4 and was used to create figure 5. The goal of the DCS is to smoothly and quickly learn the derivative increments calculated by the PID and store that information. For valid derivative increments the network is trained and a recall is performed using only the independent variables. For invalid derivative increments only the recall is performed.

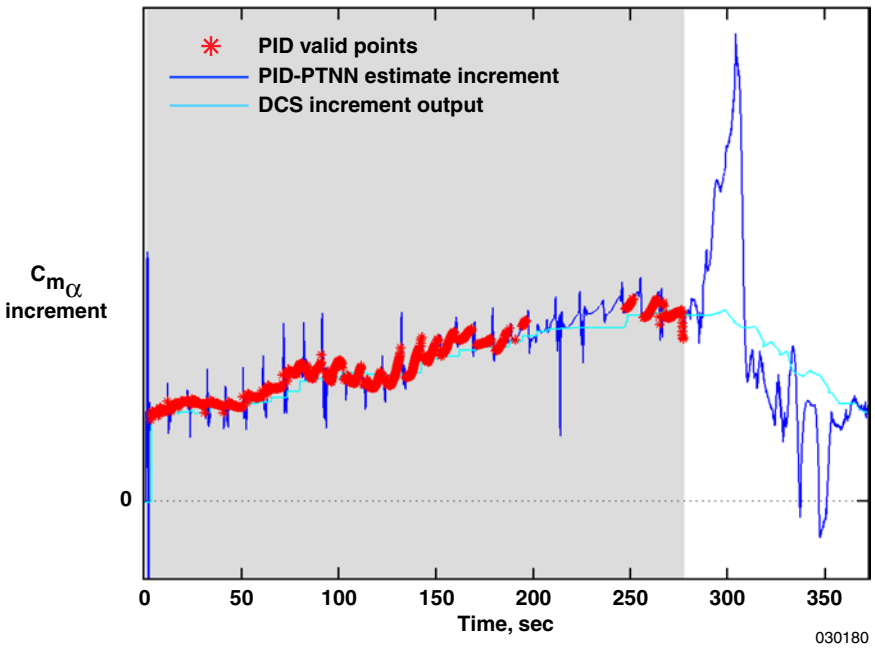


Figure 6. Simulation open-loop DCS results with modified aerodynamics.

For this maneuver, the network is started from the zero state and ends with 19 nodes and 68 edges. On the smooth deceleration where the C_{m_α} derivative increment is not valid, the DCS recalls a value using the current conditions and surface positions. Figure 7 shows the comparison among the result of adding the recalled derivative increment to the PTNN estimate, the linear aerodynamic estimate, and the original PTNN estimate for the data shown in figure 4.

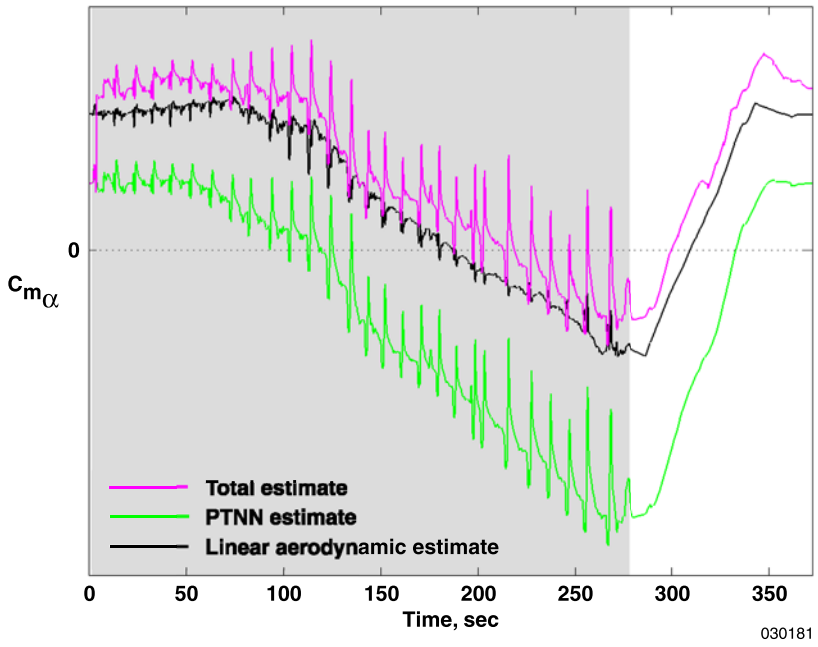


Figure 7. Simulation open-loop total estimated derivative with modified aerodynamics.

In a closed-loop configuration, the total estimate is sent to the SOFFT controller to update the model. Figure 7 illustrates how, at 4 seconds into the maneuver when the PID gives the DCS the first valid derivative increment, the total estimate quickly jumps to a value close to the linear aerodynamic estimate from the simulation. On the deceleration when the C_{m_α} PID estimate is invalid because of insufficient frequency information, the DCS successfully recalls the correct derivative increment. The roughness during the excitation phase comes from the PTNN, which is dependent on aircraft states and surface positions. The total estimated derivative is simply the sum of the PTNN estimate and the DCS derivative increment estimate.

Figure 8 shows the total estimated C_{m_α} as a function of Mach number. This figure illustrates the ability of the DCS to learn the valid derivative increment during the training portion of the flight and smoothly recall an estimate of the derivative increment when the PID estimates are invalid.

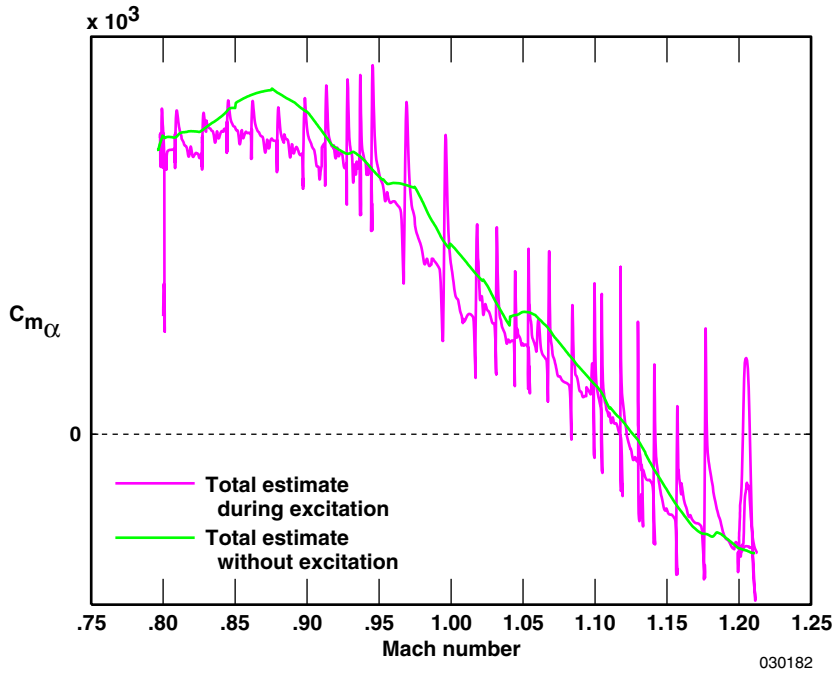


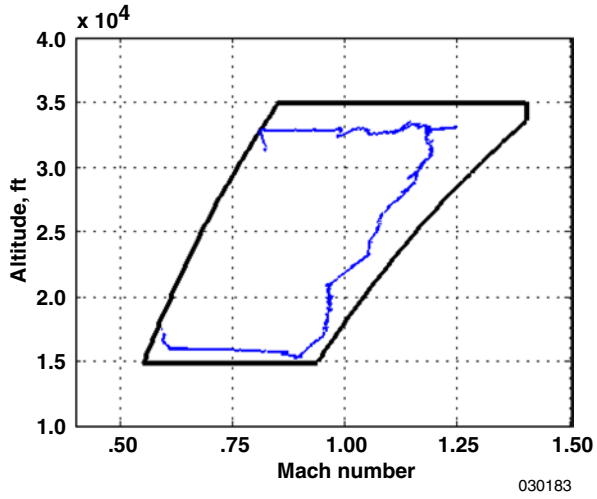
Figure 8. Simulation open-loop total estimated derivative versus Mach number with modified aerodynamics.

OPEN-LOOP FLIGHT DATA RESULTS

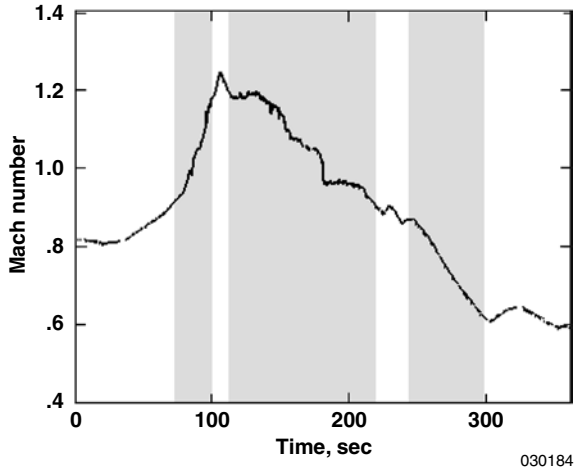
In early 2003, the F-15 aircraft was flown to obtain data for postflight analysis of the PID and DCS algorithms. These flights were conducted as a risk reduction step before the next phase in which the software will be moved onboard and run in real time in an open-loop configuration. The flight data used here are from a high-altitude acceleration followed by a fairly constant dynamic pressure decent to lower altitude, then a deceleration as shown in figure 9.

Four pitch, roll, and yaw doublets were executed by the pilot between 72 and 100 seconds into the test. Doublets were performed approximately every 5 seconds from 112 to 219 seconds, and from 243 to 298 seconds. Figure 10 shows the PID results.

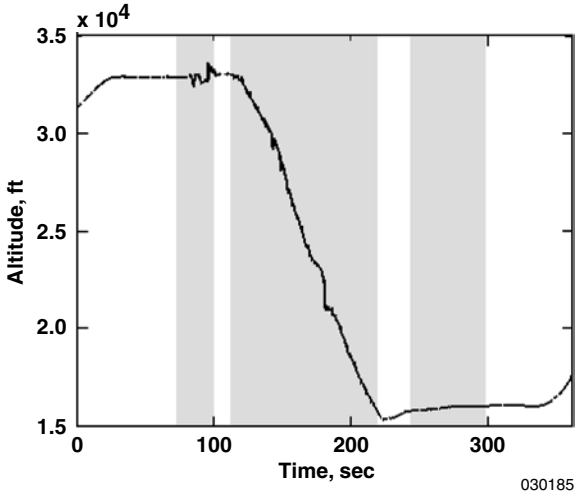
During the first excitation section, the PID does not compute any valid points because of the rapidly changing flight conditions. The PID estimates are fairly accurate, but because the PID is working on a 10-second window of input data, and angle of attack, Mach number, and C_{m_α} quickly change, higher standard errors are calculated. Valid data are seen for the first part of the second section and all of the third section, because data in the 10-second buffer are roughly at the same flight condition.



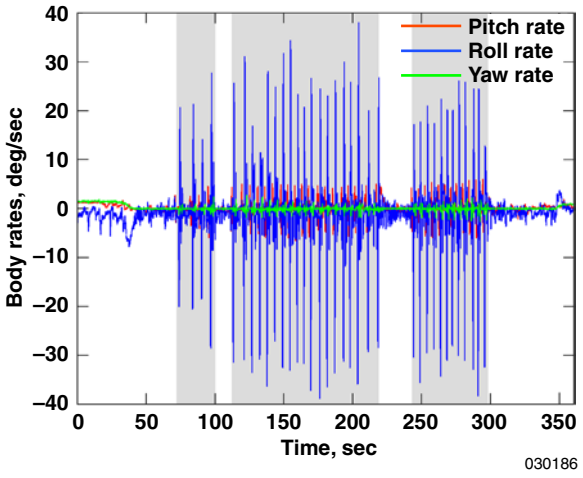
(a) Flight profile in experimental envelope.



(b) Flight profile Mach number.



(c) Flight profile altitude.



(d) System excitation.

Figure 9. Flight data open-loop flight profile.

For the actual aircraft, generating a true value for the derivatives is much more difficult. For the initial analysis the close match between PTNN estimates and PID estimates suggests that the PID satisfactorily estimates the derivatives online. Because the PID uses 13 PTNN derivatives to calculate the other 13 derivatives online, further indepth examination of the results is a complex process. The major difficulty is that if the PTNN is incorrect, the derivatives calculated online by the PID will compensate for the inaccuracies. More indepth exploration of the quality of PID derivative estimates for flight data is presented in references 6 and 14. For the analysis shown here, the fact that valid derivatives can be calculated is essential, but further tuning might be necessary for better results.

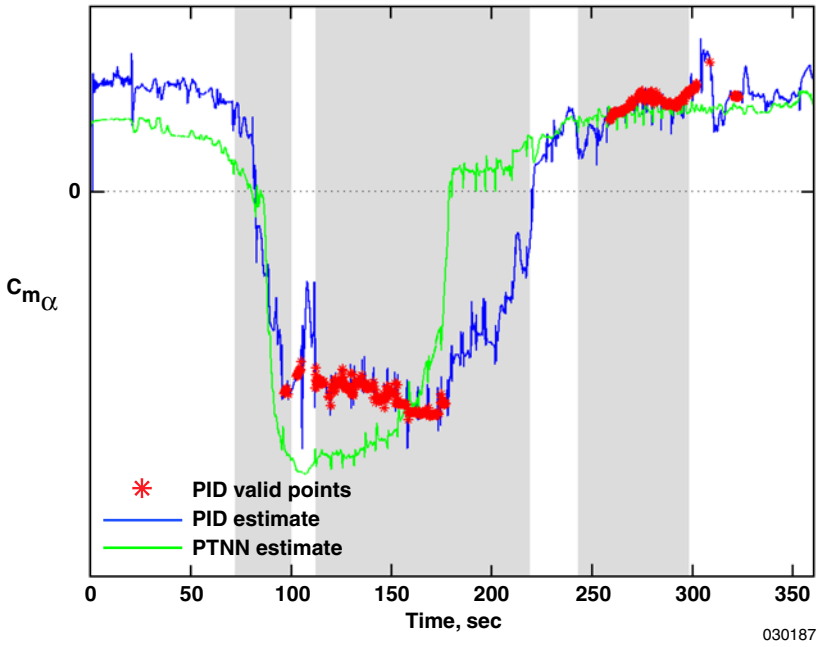


Figure 10. Flight data open-loop PID results.

Figure 11 shows the DCS performance on the C_{m_α} derivative increment for the same flight maneuver shown in figure 9. Because the research portion of the initial flights were primarily conducted to gather data for PID analysis, DCS maneuvers comparable to those shown during the simulation data analysis are not available.

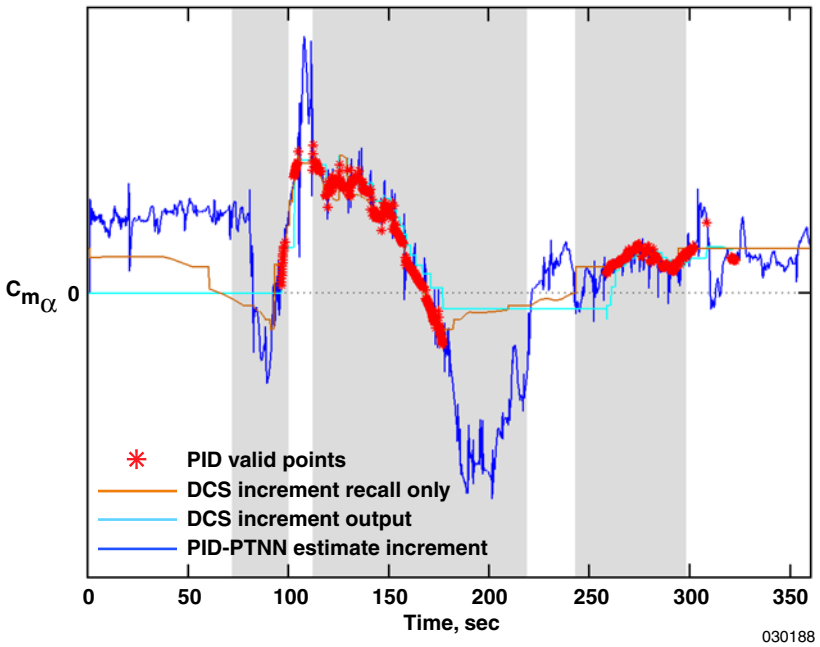


Figure 11. Flight data open-loop DCS results.

For this flight data the altitude and Mach number are constantly changing without returning to previous conditions; therefore, analyzing the learning of the DCS by looking only at the instantaneous recall results is difficult. To more accurately evaluate how well the DCS learns the derivative increment, the data are passed though the DCS twice, once with learning enabled and once with learning disabled. With learning enabled, the DCS is allowed to update the network as valid PID data are received and then estimates current derivatives using the independent inputs. For this data set 22 nodes and 70 edges are created during learning. With learning disabled, the DCS uses independent inputs to estimate current derivatives but does not update the network. The resulting performance can be analyzed by examining how well the DCS recalls the derivative increment for the same input values on which it was previously trained. The initial portion of the data and the second dip in the derivative are not learned by the DCS, because no valid derivative increments come through.

CLOSED-LOOP SIMULATION RESULTS

Closed-loop, piloted simulations have been executed to examine the effectiveness of the complete online learning system. The primary focus is to compare the nonlearning system (PTNN and SOFFT only) to the online learning system in terms of ability to adapt to a modified aerodynamic model. The major question to answer is, “Does the online learning system improve the performance of the SOFFT controller?” The performance of the SOFFT controller is determined by how well the desired model following characteristics are achieved. The specific case examined here shows the abilities of the two systems to match commanded pitch rate with actual pitch rate response of the closed-loop system.

Figure 12 shows a series of doublets without and with an aerodynamic increment added to C_{m_α} before and after activation of the online learning portion of the system. During the first 5 seconds, the pilot performed a pitch doublet with the unmodified aerodynamics using the nonlearning system with only PTNN estimates to show initial performance of the system. At 5 seconds, the aerodynamic increment was inserted and another pitch doublet was performed, still using the nonlearning system. At 10 seconds into the run, the online learning was activated with one set of pitch, roll, and yaw doublets inserted for training purposes. After 20 seconds, a final pitch doublet was inserted to show the performance of the system with online learning activated. The complete run was executed at a fairly constant altitude and Mach number to simplify the analysis.

During the first 5 seconds, the PTNN estimate, which is the same as the total estimate for the nonlearning system, is fairly close to the linear aerodynamic estimate, because the unmodified aerodynamics are used. At 5 seconds, the aerodynamic increment is observed by the change in the linear aerodynamic estimate. During this portion of the test, the total derivative estimate is again the PTNN estimate. Because the total derivative estimate is significantly different from the linear aerodynamic estimate, the SOFFT control system aircraft plant model will be inaccurate, and, as a result, degraded flight qualities are expected.When the learning system is activated and the doublets adequately excite the system, the PID estimates the derivative increments, which are learned by the DCS and added to the PTNN estimates to obtain the total estimates. The learning system quickly corrects for the aerodynamic increment, sending the improved derivative estimate to the SOFFT controller to update the model.

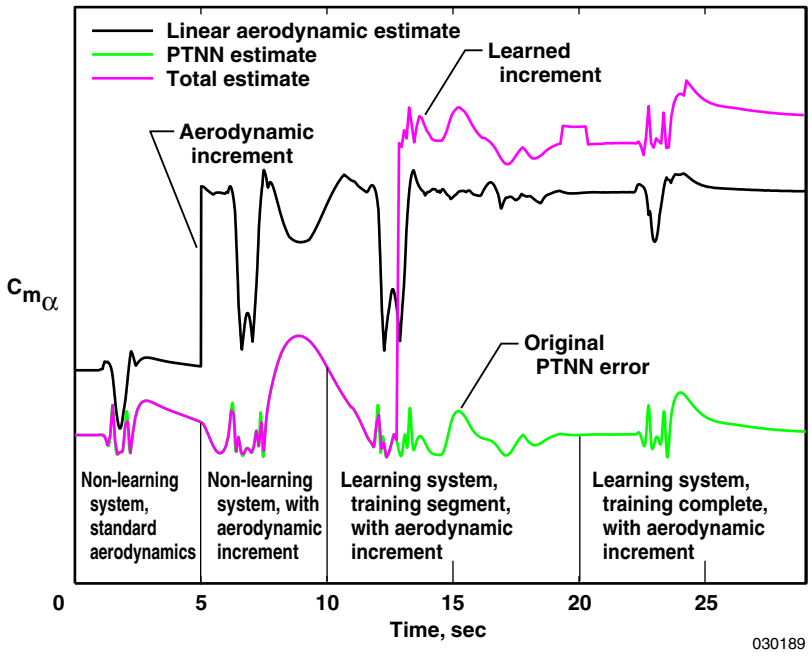


Figure 12. Simulation closed-loop total estimated derivative.

Figure 13 shows the actual pitch rate of the system compared to the commanded pitch rate. The closed-loop results of the nonlearning and learning systems are observed for the complete system.

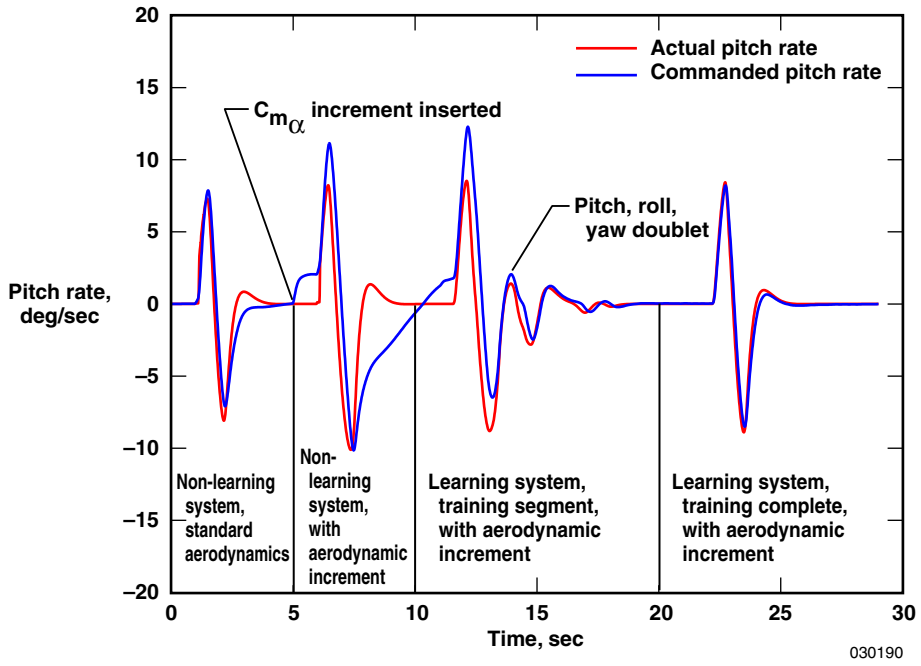


Figure 13. Simulation closed-loop pitch performance.

For the unmodified aerodynamics during the first doublet, the nonlearning system response is a fairly close match to the command. The second doublet shows that the nonlearning system has degraded control characteristics, and the response is significantly different from the command because of the aerodynamic modification. During the pitch, roll, and yaw doublets the online learning system immediately starts to improve the response of the aircraft. After exciting the system, the final pitch doublet shows excellent performance.

The performance improvement that occurs with the learning system and modified aerodynamics emphasizes that the PTNN does not exactly match the aerodynamic model in the standard simulation. As previously discussed, the suspected cause is that the PTNN was trained on an aerodynamic model using the original wind-tunnel model with rectangular engine nozzles. Although the PTNN is not perfectly accurate, the complete response of the online learning system compensates for both the PTNN inaccuracies and the modified aerodynamics, resulting in better performance than is achieved without learning.

CONCLUDING REMARKS

The online real-time parameter identification (PID) algorithm and dynamic cell structure neural network (DCS) have been successfully integrated in preparation for flight demonstration. This indirect adaptive system has worked well in the open-loop simulation data and flight data cases and the closed-loop simulation tests. The PID has demonstrated the ability to estimate the aerodynamic stability and control derivatives in real time in the flight environment. The DCS has demonstrated the ability to efficiently and effectively train and recall the derivative increment information. As a complete system, the online PID algorithm and DCS have produced promising results in a closed-loop simulation environment. The online learning system is being installed on the F-15 research aircraft for open-loop, in-flight testing. During this phase of flight tests, different properties of the PID and DCS can be explored on and offline through modifying configuration files over various flight conditions and pilot maneuvers. In-flight performance of the PID can be analyzed for estimation accuracy and validity testing. Continued effort is needed, but with further testing and analysis this system has the potential for effective use in flight projects requiring adaptive flight control systems.

REFERENCES

1. Kaneshige, John, and Karen Gundy-Burlet, “Integrated Neural Flight and Propulsion Control System,” AIAA-2001-4386, Aug. 2001.

2. Gundy-Burlet, Karen, K. Krishnakumar, Greg Limes, and Don Bryant, “Control Reallocation Strategies for Damage Adaptation in Transport Class Aircraft,” *AIAA Guidance, Navigation, and Control Conference and Exhibit,* AIAA-2003-5642, Aug. 2003.

3. Krishnakumar, K., G. Limes, K. Gundy-Burlet, and D. Bryant, “An Adaptive Critic Approach to Reference Model Adaptation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit,* AIAA-2003-5790, Aug. 2003.

4. Norgaard, Magnus, Charles C. Jorgensen, and James C. Ross, *Neural Network Prediction of New Aircraft Design Coefficients*, NASA TM-112197, May 1997.

5. Morelli, Eugene A., “Real-Time Parameter Estimation in the Frequency Domain,” *AIAA Atmospheric Flight Mechanics Conference*, AIAA-99-4043, Aug. 1999.

6. Smith, Mark S., Timothy R. Moes, and Eugene A. Morelli, “Real-Time Stability and Control Derivative Extraction from F-15 Flight Data,” *AIAA Atmospheric Flight Mechanics Conference*, AIAA-2003-5701, Aug. 2003.

7. Ahrns, Ingo, Jörg Bruske, and Gerald Sommer, “On-line Learning with Dynamic Cell Structures,” *Proceedings of ICANN’95*, vol. 2, 1995, pp.141–146.

8. Jorgensen, Charles C., *Direct Adaptive Aircraft Control Using Dynamic Cell Structure Neural Networks*, NASA TM-112198, May 1997.

9. Halyo, Nesim, and Haldun Direskeneli, *A Stochastic Optimal Feedforward and Feedback Control Methodology for Superagility*, NASA CR-4471, 1992.

10. Halyo, Nesim, *Integrated Control Using the SOFFT Control Structure*, NASA CR-4748, July 1996.

11. Urnes, James, Sr., Ron Davidson, Steve Jacobson, “A Damage Adaptive Flight Control System Using Neural Network Technology,” *2001 American Control Conference*, June 2001.

12. Lee, Chong O., Michael P. Thomson, and James M. Urnes, Sr., “Development of a Neural-Network-Based Fault-Tolerant Adaptive Flight Control System,” *2001 American Control Conference*, June 2001.

13. Norlin, Ken A., *Flight Simulation Software at NASA Dryden Flight Research Center*, NASA TM-104315, Oct. 1995.

14. Moes, Timothy R., Mark S. Smith, and Eugene A. Morelli, “Flight Investigation of Prescribed Simultaneous Independent Surface Excitations (PreSISE) for Real-Time Stability and Control Derivative Estimation,” *AIAA Atmospheric Flight Mechanics Conference*, AIAA-2003-5702, Aug. 2003.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 2003	Technical Memorandum

4. TITLE AND SUBTITLE

Integration of Online Parameter Identification and Neural Network for In-Flight Adaptive Control

5. FUNDING NUMBERS

WU 745-20-00-SE-40-00-IFS

6. AUTHOR(S)

Jacob J. Hageman, Mark S. Smith, and Susan Stachowiak

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

8. PERFORMING ORGANIZATION REPORT NUMBER

H-2543

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA/TM-2003-212028

11. SUPPLEMENTARY NOTES

Also presented at the AIAA Atmospheric Flight Mechanics Conference, Austin Texas, Aug. 11–14, 2003. AIAA-2003-5700

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified—Unlimited
Subject Category 08

This report is available at http://www.dfrc.nasa.gov/DTRS/

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

An indirect adaptive system has been constructed for robust control of an aircraft with uncertain aerodynamic characteristics. This system consists of a multilayer perceptron pre-trained neural network, online stability and control derivative identification, a dynamic cell structure online learning neural network, and a model following control system based on the stochastic optimal feedforward and feedback technique. The pre-trained neural network and model following control system have been flight-tested, but the online parameter identification and online learning neural network are new additions used for in-flight adaptation of the control system model. A description of the modification and integration of these two stand-alone software packages into the complete system in preparation for initial flight tests is presented. Open-loop results using both simulation and flight data, as well as closed-loop performance of the complete system in a nonlinear, six-degree-of-freedom, flight validated simulation, are analyzed. Results show that this online learning system, in contrast to the nonlearning system, has the ability to adapt to changes in aerodynamic characteristics in a real-time, closed-loop, piloted simulation, resulting in improved flying qualities.

14. SUBJECT TERMS	15. NUMBER OF PAGES
F-15 Aircraft, Indirect Adaptive Control, Intelligent Flight Control System, Neural Network, Parameter Identification	23
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited