Adaptive Control Using Neural Network Augmentation for a Modified F-15 Aircraft

John J. Burken, Peggy Williams-Hayes, John T. Kaneshige and Susan J. Stachowiak

Abstract— Description of the performance of a simplified dynamic inversion controller with neural network augmentation follows. Simulation studies focus on the results with and without neural network adaptation through the use of an F-15 aircraft simulator that has been modified to include canards. Simulated control law performance with a surface failure, in addition to an aerodynamic failure, is presented. The aircraft, with adaptation, attempts to minimize the inertial cross-coupling effect of the failure (a control derivative anomaly associated with a jammed control surface). The dynamic inversion controller calculates necessary surface commands to achieve desired rates. The dynamic inversion controller uses approximate short period and roll axis dynamics. The yaw axis controller is a sideslip rate command system. Methods are described to reduce the cross-coupling effect and maintain adequate tracking errors for control surface failures. The aerodynamic failure destabilizes the pitching moment due to angle of attack. The results show that control of the aircraft with the neural networks is easier (more damped) than without the neural networks. Simulation results show neural network augmentation of the controller improves performance with aerodynamic and control surface failures in terms of tracking error and cross-coupling reduction.

I. INTRODUCTION

THE objective of the NASA Intelligent Flight Control System (IFCS) program is to develop and flight-test schemes that enhance control during primary control surface failures or aerodynamic changes resulting from failures or modeling errors. The first flight phase, known as generation one (Gen 1), required aerodynamic parameter identification and is not discussed in this report (please see [1] for a discussion on the Gen 1 flight phase). The second flight phase, known as generation two (Gen 2), evaluates a neural flight control system that can provide adaptive control without explicit parameter identification. The Gen 2

approach does not require information on the nature or the extent of the failure, knowledge of the control surface positions, or information on aerodynamic failures or unmodeled parameters. The Gen 2 tracking controller adds direct adaptive neural network signals to the control law [2]–[6]. These neural networks are used to generate command augmentation signals to compensate for errors caused by unmodeled dynamics, including dynamics resulting from damage or failure. Flight demonstration began in early 2006 on the NASA F-15 aircraft. An F-15 six-degree-of-freedom (6-DOF) simulator was used in the evaluation test, which compared stabilator and canard failure compensation with the neural network algorithm. A canard failure emulates a change in pitching moment due to angle of attack, C_{m_α}, and is considered an aerodynamic malfunction caused by modeling errors or damage. The specific objectives of Gen 2 are to 1) implement and fly a direct adaptive neural network–based flight controller, 2) demonstrate the ability of the system to adapt to simulated system failures by suppressing transients associated with the failure, 3) reestablish sufficient control and handling of the vehicle for safe recovery, and 4) provide flight experience for the development of verification and validation processes for flight-critical neural network software.

II. DESCRIPTION OF THE GENERATION 2 CONTROLLER

The general control scheme of the Gen 2 controller is based on an adaptive neural controller that cancels errors associated with the dynamic inversion of the model. Initially, constant values of aerodynamic stability and control derivatives for a fixed condition in the flight envelope are used for model inversion. In addition, desired handling qualities are achieved with low-order reference models, which are based on pilot preferences [7]. The yaw axis controller is not a reference-based controller, but rather a classical yaw system. The Gen 2 control structure that uses a direct adaptive control method is shown in Fig. 1. The symbol *dd* in a subscript indicates direct derivative and *ad* indicates adaptation.

A. Reference Models

The pilot generates flight commands through

J. J. Burken is with the National Aeronautics and Space Administration (NASA) Dryden Flight Research Center, Edwards, CA 93523 USA (phone: 661-276-3726; fax: 661-276-2586; e-mail: john.burken@mail.dfrc.nasa.gov).

P. Williams-Hayes is with NASA Dryden Flight Research Center, Edwards, CA 93523 USA (e-mail: peggy.williams@mail.dfrc.nasa.gov).

J. T. Kaneshige is with NASA Ames Research Center, Moffett Field, CA 94035 USA (e-mail: j.kaneshige@mail.arc.nasa.gov).

S. J. Stachowiak is with NASA Johnson Space Center, Houston, TX 77058 USA (e-mail: susan.j.stachowiak@mail.jsc.nasa.gov).
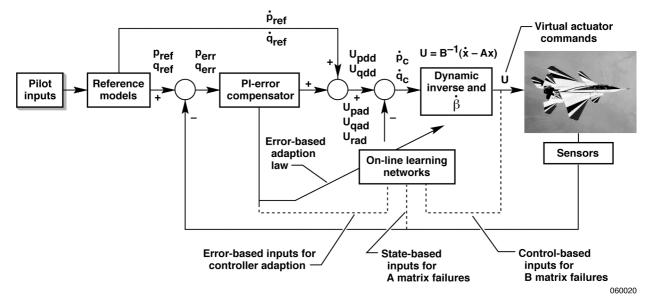
Fig. 1. Generation 2 architecture in which the inversion is applied to the *B* matrix.

longitudinal and lateral stick deflections {*dep*, *dap*} and rudder pedals {*drp*}. When these inputs are used, the reference model (*ref*) provides pitch (*p*), roll (*q*) and yaw (*r*) rate commands $\{p_{ref}, q_{ref}\}$ and acceleration commands $\{\dot{p}_{ref}, \dot{q}_{ref}\}$ by means of first-order roll rate and second-order pitch rate transfer functions, as shown in (1) and (2). The symbol τ_r is rise time, in seconds, and *s* is the Laplace operator. The symbol *K* is for constant gain, with the subscripts *lat* for lateral and *lon* for longitudinal. Short period frequency, ω_{sp}, and short period damping, ζ_{sp}, are in rad/sec. The symbol *L* represents neural network error-modification damping in rad/sec.

$$\frac{p_{ref}}{dap} = \frac{K_{lat}}{\tau_r s + 1} \tag{1}$$

$$\frac{q_{ref}}{dep} = \frac{K_{lon}\omega_{sp}^2(s + L_\alpha)}{s^2 + 2\zeta_{sp}\omega_{sp}s + \omega_{sp}^2} \tag{2}$$

These transfer functions are low-order equivalent systems designed to achieve level-one handling qualities [7].

B. Simplified Dynamic Inversion

The inputs to the dynamic inversion controller are commanded angular accelerations $\{\dot{p}_c, \dot{q}_c\}$, which are computed with the equation

$$\begin{bmatrix} \dot{p}_c \\ \dot{q}_c \end{bmatrix} = \begin{bmatrix} U_p \\ U_q \end{bmatrix} - \begin{bmatrix} U_{pad} \\ U_{qad} \end{bmatrix} \tag{3}$$

in which $\{U_{pad}, U_{qad}\}$ are augmentation commands generated by adaptive neural networks. The symbol *U* is controller command and the subscript *c* represents command. These augmentation commands compensate for the estimated errors resulting from the difference between reference (p_{ref} or q_{ref}) and aircraft angular rates (4).

$$p_{err} = p_{ref} - p$$
$$q_{err} = q_{ref} - q \tag{4}$$

The subscript *err* is for error. The pseudocontrol acceleration commands $\{U_p, U_q\}$ are computed with the equations

$$U_p = \left[K_{pp} + \frac{K_{ip}}{s}\right]p_{err} + \dot{p}_{ref} \tag{5}$$

$$U_q = \left[K_{pq} + \frac{K_{iq}}{s}\right]q_{err} + \dot{q}_{ref} \tag{6}$$

in which K_p and K_i are proportional (*p*) and integral (*i*) constants, respectively, for each axis. The computation process for the pseudocontrol acceleration commands is shown in Fig. 2. The simplified dynamic inversion
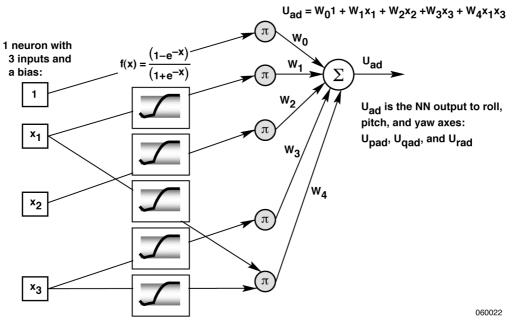
$$U_{ad} = W_0 1 + W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_1 x_3$$

Fig. 2. Simple sigma pi neural network structure (Σ = sum, π = product).

algorithm inverts a B matrix with states for modeling the short period and roll modes. The symbol B indicates control and T means transpose in (7). To protect against a poorly ranked B matrix, the inversion is generated by a pseudoinverse (7).

$$B^{-1} = \frac{B^T * \mathrm{Adjoint}(B * B^T)}{\mathrm{Determinant}(B * B^T)} \quad (7)$$

The inversion is used to determine necessary control surface deflections $\{\delta_a (\text{roll axis}), \delta_e (\text{pitch axis})\}$. Control surface commands $\{\delta_{a_c}, \delta_{e_c}\}$ are obtained with the equation

$$\begin{bmatrix} \delta_{a_c} \\ \delta_{e_c} \end{bmatrix} = B^{-1} \begin{bmatrix} \dot{p}_c - L_1 \\ \dot{q}_c - M_1 \end{bmatrix} \quad (8)$$

in which B is the state space system control matrix and the terms $\{\dot{p}_c - L_1, \dot{q}_c - M_1\}$ are the differences between input acceleration commands and actual plant acceleration contributions $\{L_1 (\text{rolling moment, ft-lbf}), M_1 (\text{pitching moment, ft-lbf})\}$. These plant contributions are calculated from the appropriate states for modeling the short period and roll mode dynamics.

C. Adaptive Neural Network

The purpose of the neural network system is to accommodate large errors that are not anticipated in the nominal control law design phase. In a failed flight condition or configuration (A or B matrix), errors will develop that are larger than expected. The adaptive neural network operates in conjunction with the error of the control system. By recognizing patterns in the behavior of the error, the neural networks can learn to remove the error biases through control augmentation commands (Fig. 1).

The adaptation signal attempts to remove errors in the neural flight control system to improve handling qualities in the presence of unknown failures. The neural network that provides this online adaptation scheme is known as "sigma pi." A simple single hidden layer sigma pi network is shown in Fig. 2. The name "sigma pi" is derived from the underlying equations of the network that sum (Σ) the products (π) of the inputs to the neural network with their associated weights. The weights of the neural network, W, are determined by a training algorithm, also known as an adaptation or learning rule. Learning involves adjusting these weights so that the network has a valid relationship between the inputs and outputs, and in turn, minimizes the error.

Because a neural network is designed to recognize patterns between inputs and errors, the selected inputs must provide enough coverage for the network to be able to capture the behavior of the error. If too many inputs exist, however, the neural network may not be able to adapt quickly enough for practical application. Although the adaptation gain can be used to increase the rate of adaptation, it can cause large transients in the neural network outputs when it becomes too big. These transients are caused by spikes and sometimes sign changes in the network weights, which are often encountered before the network converges to a learned state.

D. Application of the Adaptive Neural Network

The adaptive neural network implemented for the flight

test is divided into three separate networks, one each for the pitch, roll, and yaw axes. Inputs to the network consist of control commands, sensor feedback, and bias terms. The number of inputs to each individual neural network vary; the roll axis uses six, the pitch axis uses seven, and the yaw axis uses ten. The output of each neural network is an angular acceleration command that augments the control signal from the research controller for each axis. To couple the three networks, pitch information is included in the roll and yaw neural networks, and some roll axis signals are included in the pitch and yaw neural networks.

The neural network output, U_{ad}, is the control augmentation command comprised of three components: roll, pitch, and yaw (U_{pad}, U_{qad}, U_{rad}). This neural network output is computed with (9) (see Fig. 1).

$$U_{ad} = W^T B_a C \tag{9}$$

The vector of basis function, B_a, is computed from the inputs in each signal input category by means of a product. The network weights, W, are computed by an adaptation law, which includes an adaptation gain, G, and an error-modification term, L. The error-modification term helps contain the parameter growth of the weights and can be considered a damping term. The final adaptation law is

$$\dot{W} = -G\left(U_{err}B_a + LU_{err}W\right)dt \tag{10}$$

in which $\dot{W}$ is the weight increment for the current time step; W is the weight from the previous time-step; dt is the time-step, which is 0.0125 seconds (80 Hz); G is the adaptation gain, which is sometimes called the learning rate; and U_{err} is the error compensation. This weight calculation is currently implemented in the Gen 2 system.

The squashing function used to normalize inputs to the neural network is a sigmoid function and has the form (also shown in Fig. 2)

$$f(x) = (1 - e^{-x})/(1 + e^{-x}). \tag{11}$$

The adaptation law is computed as a function of the proportional and integral (PI) controller gains and tracking error (p_{err}, q_{err}, and r_{err}) equated to err in (12). The z subscript indicates pitch (p), roll (q) or yaw (r).

$$U_{err} = Kz_i \int err + Kz_p err \tag{12}$$

This computation allows the neural networks to work in conjunction with the dynamic inversion PI controllers. Furthermore, because the adaptation gain can be used to specify the overall rate of adaptation, the adaptation law gains can be viewed as specifying the relative rates of adaptation.

III. SIMULATION RESULTS

The first type of simulated failure (A matrix failure), which represents an aerodynamic failure, inserts a multiplier onto the canard surface command (change in $C_{m\alpha}$). The second type of simulated failure (B matrix failure), which represents a surface failure, inserts a jammed stabilator failure. Simulation results illustrate the experiment that was flown and highlight the benefits provided by the Gen 2 control system. The flight condition used in this research is at a Mach number of 0.7 and an altitude of 6,096 m. All of the pilot inputs to the simulation time histories are "canned" piloted stick inputs, and no attempt to correct for the aircraft attitudes is added to the piloted inputs. This "canned" pilot input method was used only for comparison purposes and is not intended for flight test at this point in time. Because the controller is a rate command system, attitudes such as bank angle, φ, are used only for comparison and disturbance rejection trade-off studies. For instance, when a failure is imparted on the aircraft and the resulting attitudes change minimally, the control system is considered to have good robustness properties.

A. A Matrix Failure (Aerodynamic Failure)

The first case is an A matrix failure imposed on both the left and right canards, with a multiplier of -0.5 on the nominal angle-of-attack schedule. This failure forces the canards to a less stable configuration. A 30-second time history with three longitudinal pilot stick inputs and a failure imposed at 11 seconds is shown in Fig. 3. Neural network is labeled NN and n_z indicates normal acceleration. In the first 10 seconds a normal response demonstrates how the pitch rate follows the commanded pitch rate (represented by the solid black line). The blue lines in Fig. 3, which represent the aircraft response when the neural network is inactive, show that after the failure is inserted the aircraft is stable but it experiences two or three overshoots. When the neural network is active (represented by the red lines in Fig. 3), the response is better damped and the commanded pitch rate is followed more closely than when the neural network is inactive. By the third pilot input, the neural network response is very close to the commanded pitch rate (represented by the solid black line). The results demonstrate that with this type of failure the neural networks help with tracking and damping.
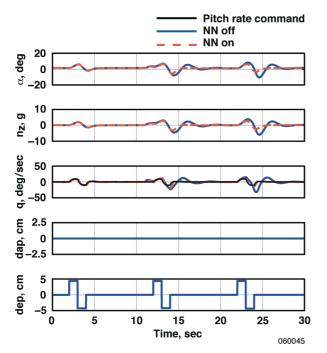
Fig. 3. Time history of longitudinal aircraft states of an *A* matrix canard failure at 11 seconds (multiplier = − 0.5).

Fig. 4. Time history of longitudinal aircraft states of a *B* matrix right stabilator lock failure (right stabilator = − 4 degrees lock from trim at 11 seconds).

B. *B Matrix Failure (Control Surface Failure)*

The second case is a *B* matrix failure, involving a − 4-degree lock from trim, which is imposed on the right stabilator 11 seconds into the simulation run. A 40-second time history in which lateral pilot stick inputs are commanded is shown in Fig. 4. During the first 10 seconds the pitch rate does not move with the lateral pilot input. This situation is preferred; roll inputs should have little or no impact on longitudinal states. When the stabilator failure is injected at 11 seconds, note that transients occur in angle of attack, normal acceleration, and pitch rate. Initially cross-coupling reduces minimally when the neural networks are active, compared to when the neural networks are inactive, as shown in Fig. 4 (see angle of attack and normal acceleration time histories). As the "canned" pilot input resumes at 16 seconds, the neural networks continue to adapt and show improvement over the nonadaptive case. The amount of normal acceleration disturbance during a roll command is approximately 40-percent less when the neural networks are active.
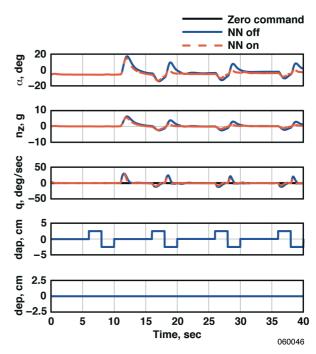
The lateral-directional states from the roll inputs are shown in Fig. 5. As previously noted, the inputs are "canned" and no other pilot commands are injected to level the aircraft after the failure. Note that the yaw rate, bank angle, and sideslip excursions are lower when the neural networks are active than when the neural networks are inactive. The roll axis is a roll rate command system (represented by the black solid line on the top plot). During the first commanded doublet no failure occurs and the tracking is good for both cases (neural networks inactive and active). After the failure is activated, the tracking is somewhat better when the neural networks are active than when they are inactive. And finally, Fig. 6 shows the neural network contribution commands to the roll, pitch, and yaw axes. An abundance of cross-coupling occurs with this failure, which in turn causes all three networks to become active. For a more detailed description of this research see [10].
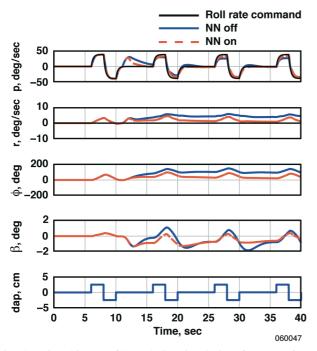
Fig. 5. Time history of lateral-directional aircraft states of a *B* matrix right stabilator lock failure (right stabilator = − 4 degrees lock from trim at 11 seconds).
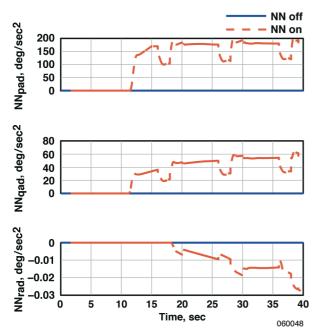

Fig. 6. Time history of neural network activity of a *B* matrix right stabilator lock failure (right stabilator = − 4 degrees lock from trim at 11 seconds).

IV. SUMMARY OF RESULTS

Simulation results have been presented of a neural network adaptive controller that compensates for errors resulting from aerodynamic and control surface failures. This hybrid controller uses simplified dynamic inversion

control for the pitch and roll axes and a classical sideslip rate controller for the yaw axis. The neural network is an on-line direct adaptive algorithm that attempts to drive the error between the reference model and commanded state to zero. An aerodynamic failure and jammed control surface failure have been demonstrated. In both failure cases the controller with neural network augmentation demonstrated improvements compared to the nonadaptive controller without neural network augmentation.

During the aerodynamic failure, the neural networks improved the damping and tracking. During the jammed control surface failure, the neural networks reduced cross-coupling, enabling the pilot to fly to the desired trajectory with fewer tracking errors. The neural networks also reduced the tracking errors.

REFERENCES

[1]   Peggy S. Williams-Hayes, *Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System*, NASA/TM-2004-212857, 2004.
[2]   Peggy S. Williams-Hayes, *Flight Test Implementation of a Second Generation Intelligent Flight Control System*, NASA/TM-2005-213669, 2005.
[3]   A. J. Calise, S. Lee, and M. Sharma, "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," AIAA-98-4108, Aug. 1998.
[4]   Rolf T. Rysdyk, and Anthony J. Calise, "Fault Tolerant Flight Control Via Adaptive Neural Network Augmentation," AIAA-98-4483, Aug. 1998.
[5]   Mario G. Perhinschi, Marcello R. Napolitano, Giampiero Campa, Brad Seanor, Srikanth Gururajan, and Gu Yu, "Design and Flight Testing of Intelligent Flight Control Laws for the WVU YF-22 Model Aircraft," AIAA-2005-6445, Aug. 2005.
[6]   John Kaneshige, John Bull, and Joseph J. Totah, "Generic Neural Flight Control and Autopilot System," AIAA-2000-4281, Aug. 2000.
[7]   *Flying Qualities of Piloted Vehicles*, U.S. Department of Defense, MIL-STD-1797, March 31, 1987.
[8]   John H. Blakelock, *Automatic Control of Aircraft and Missiles*, 2nd edition, John Wiley & Sons, Inc., New York, 1991.
[9]   Mario G. Perhinschi, Marcello Napolitano, Giampiero Campa, Heather E. Burke, Richard R. Larson, John Burken, and Mario L. Fravolini, "Design and Testing of a Safety Monitor Scheme on the NASA Gen 2 IFCS F-15 Flight Simulator," AIAA-2004-6284, Sept. 2004.
[10] John J. Burken, Peggy Williams-Hayes, John T. Kaneshige, and Susan J. Stachowiak, *Reconfigurable Control with Neural Network Augmentation for a Modified F-15 Aircraft*, NASA/TM-2006-213678, April 2006.