Flight Test Comparison of Different Adaptive Augmentations of Fault Tolerant Control Laws for a Modified F-15 Aircraft

John J. Burken[*], Curtis E. Hanson[†] and James A. Lee.[‡]
NASA Dryden Flight Research Center, Edwards, CA, 93523

and

John T. Kaneshige[§]
NASA Ames Research Center, Moffett Field, CA, 94035

This report describes the improvements and enhancements to a neural network–based approach for directly adapting to aerodynamic changes resulting from damage or failures. This research is a follow-on effort to flight tests performed on the NASA F-15 aircraft as part of the Intelligent Flight Control System research effort. Previous flight test results demonstrated the potential for performance improvement under destabilizing damage conditions. Little or no improvement was provided under simulated control surface failures, however, and the adaptive system was prone to pilot-induced oscillations. An improved controller was designed to reduce the occurrence of pilot-induced oscillations and increase robustness to failures in general. This report presents an analysis of the neural networks used in the previous flight test, the improved adaptive controller, and the baseline case with no adaptation. Flight test results demonstrate significant improvement in performance by using the new adaptive controller compared with the previous adaptive system and the baseline system for control surface failures.

Nomenclature

A	=	state derivative matrix
a_y	=	lateral acceleration, ft/s^2
B	=	control derivative matrix
B_a	=	neural network basis function
C	=	neural network input categories
CHR	=	Cooper-Harper Rating
C_{m_α}	=	coefficient of pitching moment due to angle of attack, 1/deg
deg	=	degree
Dt	=	sample time, s
G	=	neural network adaptation gain
IFCS	=	Intelligent Flight Control System
K	=	constant gain
k	=	current frame

[*] Aerospace Engineer, Flight Controls Branch, P.O. Box 273/Mail Stop 4840D, AIAA Member
[†] Aerospace Engineer, Flight Controls Branch, P.O. Box 273/Mail Stop 4840D, AIAA Member
[‡] Aerospace Engineer, Flight Controls Branch, P.O. Box 273/Mail Stop 4840D, AIAA Member
[§] Computer Engineer, Intelligent Systems Division, Mail Stop 269-1, AIAA Member

K_z	=	neural network learning gain
L	=	neural network error-modification gain
NASA	=	National Aeronautics and Space Administration
NN	=	neural network
n_y	=	lateral acceleration, g
n_z	=	normal acceleration, g
p	=	body axis roll rate, deg/s
$\dot{p}$	=	roll angle acceleration, deg/s/s
PID	=	proportional-integral-derivative
PIO	=	pilot-induced oscillation
q	=	body axis pitch rate, deg/s
$\dot{q}$	=	pitch angle acceleration , deg/s/s
R	=	diagonal scaling matrix
r	=	body axis yaw rate, deg/s
$\dot{r}$	=	yaw angle acceleration, deg/s/s
s	=	Laplace operator
T	=	transpose
U	=	controller command(s)
u	=	flight control surface deflection(s), deg
V_t	=	total velocity, ft/s
W	=	neural network weights
$\dot{W}$	=	rate of change of neural network weights
x	=	aircraft state(s)
$\dot{x}$	=	rate of change of the state x
Z	=	neural network learning signal(s)
α	=	angle of attack, deg
$\dot{\alpha}$	=	rate of change of angle of attack, deg/s
β	=	angle of sideslip, deg
$\dot{\beta}$	=	rate of change of angle of sideslip, deg/s
δ_{lat}	=	pilot lateral stick position, in.
δ_{lon}	=	pilot longitudinal stick position, in.
δ_{dir}	=	pilot rudder pedal position, in.
θ	=	pitch angle, deg
ϕ	=	bank angle, deg

Subscripts

ad	=	adaptation
c	=	command
d	=	derivative
dd	=	direct feedback
e	=	error
i	=	integral
P	=	proportional
ref	=	reference
v	=	virtual

I. Introduction and Statement of the Problem

The objective of the NASA Intelligent Flight Control System (IFCS) research effort was to develop and flight-test schemes that enhance control during primary control surface failures and other aerodynamic changes resulting from damage. The first flight phase, referred to as “generation 1” (“Gen1”), focused on the development and evaluation of an aerodynamic parameter identification method.[1] The second flight phase, referred to as "generation 2" ("Gen2"), focused on the development and evaluation of a direct adaptive control method that uses neural networks to generate command augmentation signals to compensate for errors caused by unmodeled dynamics.[2] In principle, this approach does not require information on the nature of the failure or the extent of the damage. This report describes a follow-on effort to the second flight phase, referred to as "generation 2a" ("Gen2a"), to develop and evaluate enhancements to the neural adaptive flight control system.

A simulated stabilator failure mode was developed as a test for the IFCS. The stabilator surfaces provide the majority of the control power for both pitch and roll control. Immobilizing one of these control surfaces (simulating a frozen or jammed surface) causes a significant change in the dynamic behavior of the vehicle. For example, with this failure, a pitch command results in half of the expected pitching moment and generates a significant rolling moment. This failure is referred to as a B-matrix failure, because it is associated with a control derivative. This failure is then tested with open-loop flying tasks such as roll or pitch captures or with a closed-loop task. The closed-loop task involves 1-g formation flight using a chase airplane for tight station keeping.

The purpose of the Gen2a research task was to improve the command tracking and pilot-handling qualities with control surface failures when compared to Gen2. During the Gen2 flight phase, the neural network adaptation worsened the Cooper-Harper Rating[3] (CHR) for the B-matrix failures.

With a stabilator failure at trim without adaptation, the pilots gave an average CHR of 4 for the fine-tracking task during formation flight. The same task with Gen2 adaptation received an average CHR of 5. One pilot gave the task a CHR of 7 with adaptation compared with a CHR of 4 without adaptation.[4] Note that more CHR data would be needed to provide a definitive pilot rating. The overall results indicate, however, that with a surface failure, the aircraft had pilot-induced oscillation (PIO) tendencies caused by the neural network adaptation.[5] If an aircraft is PIO prone, then the handling quality ratings will be comparatively poorer.

The goals for the Gen2a phase were to remedy the problems of Gen2 and improve the performance of the adaptation control laws. This report presents flight results showing the improvements of Gen2a compared with Gen2 and compared with results without adaptation. This report does not discuss the flying qualities in detail and only briefly discusses A-matrix failures. For more information on A-matrix failures, see references 5 and 6. Handling qualities resulting from A- and B-matrix failures will be covered in future reports.

II. Background

Flight tests of the IFCS Gen2 controller began in early 2006 on the NASA F-15 aircraft (Fig. 1), originally built by McDonnell Douglas Aircraft Company (now Boeing Phantom Works, St. Louis, Missouri). This aircraft has been highly modified to support various experimental capabilities, including the simulation of damage through the use of additional canards. These canards are scheduled with angle of attack and Mach number and are not symmetrically controlled by the IFCS controller. When the aircraft changes angle of attack and/or Mach number, the canards move symmetrically to reduce the destabilizing effect of the canard surfaces. As a result, they can be used to emulate symmetric aerodynamic (or A-matrix) changes in the aircraft by scaling the nominal angle-of-attack schedule with reduced gain and negative value multipliers. These "canard failures" result in changes in the coefficient of pitching moment due to angle-of-attack (C_{m_α}), which decreases the aircraft stability.[6] Control surface (or B-matrix) failures are emulated by commanding one of the stabilators to a fixed position offset from trim. Because pitch control is provided by symmetric stabilators, and roll control is provided by differential ailerons and differential stabilators, these "stabilator failures" result in out-of-trim conditions, losses of control power, and control-induced cross-coupling effects in both the longitudinal and lateral axes (see section III, "Flight Results," for more details).

Figure 1. The modified NASA F-15 aircraft with canards.

The Gen2 flight tests consisted of performance evaluation, with and without neural network augmentation, in the presence of *A*-matrix (canard) failures and *B*-matrix (left stabilator) failures. The evaluation was based on performance measurements and pilot ratings.[4] In summary, for *A*-matrix (canard) failures, neural network augmentation was generally found to provide significant improvements in overall pitch performance. The *B*-matrix failures had mixed results. In some cases the neural network augmentation provided only slight improvements in pitch rate response; in other cases a tendency for PIOs occurred.

A. Generation 2 Neural Adaptive Flight Control System Description

Figure 2 shows the generic neural adaptive flight control system, which is based upon the augmented model inversion controller, developed by Calise, et al.,[7] and Rysdyk.[8] This method was used for the pitch and roll axes in the Gen2 and Gen2a flight phases. An explicit model-following scheme is used to achieve desired handling qualities. This direct adaptive approach integrates feedback linearization theory with on-line learning sigma-pi neural networks. These networks generate command augmentation signals to compensate for errors in the model inversion. A Lyapunov stability proof guarantees boundedness of the tracking error and network weights.[8]

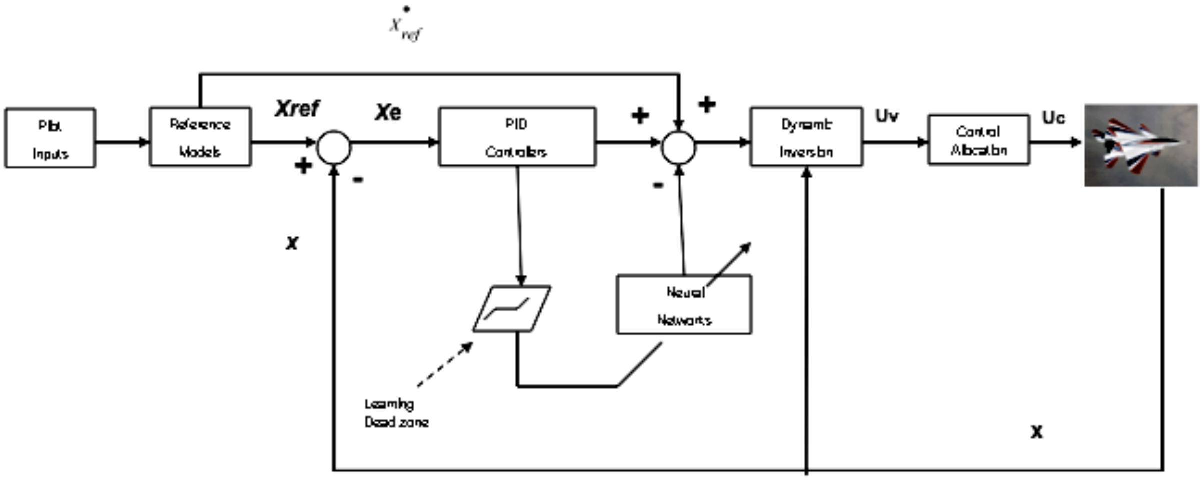


Figure 2. Neural adaptive flight control system block diagram for the pitch and roll axes in generation 2 (Gen2) and generation 2a (Gen2a).

The pilot generates flight commands through longitudinal and lateral stick deflections and rudder pedal inputs. These commands are then filtered through reference models, which produce body axis rotational rate (x_{ref}) and acceleration ($\dot{x}_{ref}$) signals with the desired frequency and damping characteristics.[9] Dynamic inversion is then used to compute the necessary virtual moment commands (u_v) to achieve the desired accelerations. If the dynamic inversion and aircraft dynamics behave together as a perfect integrator, then the response of the aircraft would match the reference models. Because errors are introduced by inaccuracies in the inversion, however, proportional-integral-derivative (PID) controllers are necessary to generate error compensation commands (U_{dd}), as shown in Eq. (1).

$$U_{dd} = K_i \int x_e + K_p x_e + K_d \dot{x}_e \tag{1}$$

The neural networks work in conjunction with the PID controllers, by recognizing error patterns and learning how to compensate for them, through the generation of augmentation commands (U_{ad}). The selected neural network inputs consist of sensor feedback (for *A*-matrix failures), control commands (for *B*-matrix failures), and bias terms (for out-of-trim conditions). These inputs are normally separated into different categories (C), which are then combined together to form a basis function (B_a) using nested Kronecker products, as shown in Eq. (2).

$$B_a = kron\left(kron\left(C_1, C_2\right), C_3\right)... \tag{2}$$

Note that the nested Kronecker products within the sigma-pi neural networks were not used in the Gen2 or Gen2a flight phases to limit the number of network weights. This method helps to simplify the sensitivity analysis of each input and tends to increase learning rates while reducing learning transients. In an ideal situation, the neural networks would be provided with a wealth of information so that they could adapt to the widest variety of unknown damage or failure scenarios. A large number of neural network inputs, however, typically result in larger learning transients and longer weight convergence times. This basis function can then be used to update the neural network weights (W) with an adaptation law, as shown in Eq. (3), and to compute the augmentation command (U_{ad}), as shown in Eq. (4).

$$\dot{W} = -G\left(ZB_a + L|Z|W\right) \tag{3}$$

$$U_{ad} = W^T B_a \tag{4}$$

The adaptation gain (G) specifies how fast the weights will adapt. The error-modification (L) term helps to contain the growth of the weights. The learning signal (Z) is computed as a function of error, as shown in Eq. (5), with learning gains (K_z) that are conditionally based upon the dynamic inversion PID controller gains.[7, 8]

$$Z = K_{z_i} \int x_e + K_{z_p} x_e + K_{z_d} \dot{x}_e \tag{5}$$

The IFCS Gen2 (and Gen2a) pitch and roll axes architecture uses a pseudo inverting Versatile Control Augmentation System (VCAS), developed by Boeing Phantom Works. Directional control for Gen2 is provided by a neural adaptive classical yaw-damping controller (Fig. 3) in place of PID control with dynamic inversion. Gen2 was designed with Eq. (5) to have "tuning knobs" on the Lyapunov gains, K_{z_i} , K_{z_p} , and K_{z_d} , which made the terms an array. This design was later discovered to be very complicated and not worthwhile. The tuning knobs required the learning dead zone on each axis (*p*, *q*, and *r*) to be an array (Fig. 2). (The dimensions of the dead zones, "DZ," for the *p*, *q*, and *r* axes were 6, 7, and 10, respectively; $\mathrm{DZ}p \in \Re^6, \mathrm{DZ}q \in \Re^7 \;\&\; \mathrm{DZ}r \in \Re^{10}$.) A dead zone is very important, because it forces the adaptation to stop changing when the error signal is small. The size of the dead zone depends on the requirements and can be derived by analytical methods or in the simulator. For the Gen2 flight phase, the sizes of the dead zones were determined in the simulator by testing a healthy aircraft and observing the size of the error signals (*Z*). The dead zones were set at 120 percent of the healthy aircraft signal (*Z*) to keep the neural networks from adapting to a healthy aircraft. Adaptation to small errors can potentially lead to gain saturation. Limits were also placed on the magnitudes of the network weights to provide a limited authority system for initial flight test purposes.

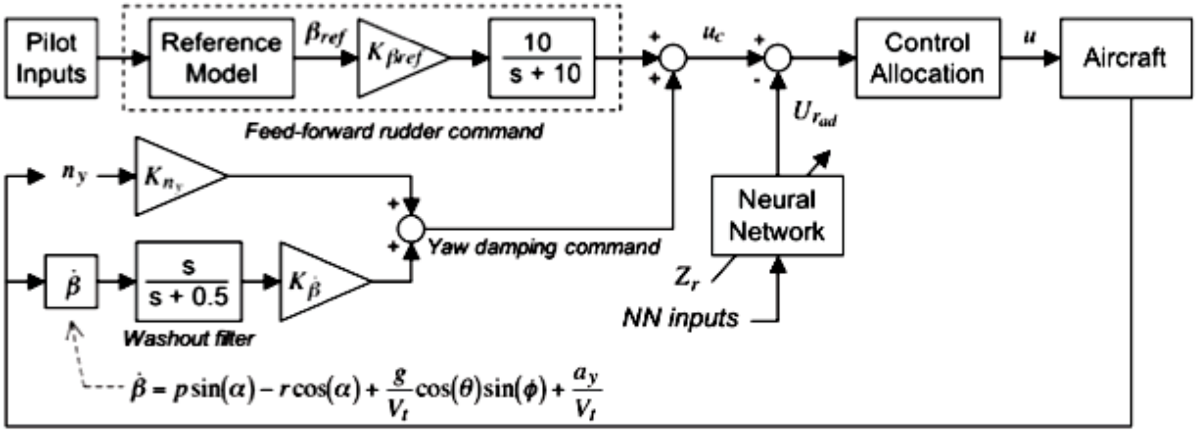


Figure 3. Generation 2 (Gen2) neural adaptive classical yaw controller.

B. Generation 2a Design Enhancements

Performance objectives of the neural adaptive flight control tests include restoring model-following tracking performance, reducing cross-coupling effects, and reducing initial transients caused by failure insertion. For application to piloted aircraft, however, the neural networks must also adapt quickly enough to assist pilots in controlling a damaged aircraft, yet avoid learning transients that could interfere with the pilot's ability to control the aircraft during adaptation. The Gen2a enhancements have five parts:

1) Neural network input changes
2) Array-to-scalar dead zones
3) Beta yaw axis controller
4) Reduced learning rate modification
5) Neural network weight decay term.

1. Neural Network Input Changes

Some of the Gen2 neural network inputs have resulted in stability and robustness problems that can have a negative effect on the performance of the neural adaptive flight control system.[10] Table 1 shows the inputs selected for the Gen2 and Gen2a neural networks.

Table 1. Neural network inputs.

Axis	Gen2	Gen2a
Lateral	$bias, (\dot{p}_{ref} + Up_{dd}), p_e, \int p_e, (\dot{q}_{ref} + Uq_{dd}), (\dot{r}_{ref} + Ur_{dd})$	$bias, p, \beta, \dot{p}_{ref}, p_{ref}, \dot{q}_{ref}, \dot{\beta}_{ref}, \beta_{ref}$
Longitudinal	$bias, \alpha, (\dot{q}_{ref} + Uq_{dd}), q_e, \int q_e, (\dot{p}_{ref} + Up_{dd}), (\dot{r}_{ref} + Ur_{dd})$	$bias, q, \alpha, \dot{q}_{ref}, \dot{p}_{ref}, p_{ref}, \dot{\beta}_{ref}, \beta_{ref}$
Directional	$bias, \delta_{lat}, \delta_{lon}, \delta_{dir}, \dot{\beta}, \varphi, u_c, (\dot{p}_{ref} + Up_{dd}), (\dot{q}_{ref} + Uq_{dd}), Ur_{ad}$	$bias, p, \beta, r, \dot{\beta}_{ref}, \beta_{ref}$

These inputs consist of bias terms, sensor feedback, and control commands. The bias terms are used to compensate for out-of-trim conditions. Sensor feedback provides state variable information to compensate for *A*-matrix failures. Control commands are used to compensate for *B*-matrix failures. The virtual surface commands (u_v) might seem to be ideal inputs for *B*-matrix failure compensation. These commands contain the adaptive signal (U_{ad}), however, as shown in Fig. 4, which can result in stability problems when fed back into the neural networks. As a result, the Gen2 neural network inputs used a combination of the feedforward acceleration signal ($\dot{x}_{ref}$) and output of the PID controller (U_{dd}). As also shown in Fig. 4, however, this combination can result in equivalently increasing the gains of the PID controller. Furthermore, it can allow the neural networks to compensate for all sources of error (including *A*-matrix failures) by applying more gain, which can result in a high-gain controller. Gen2a did not use the U_{dd} input term and therefore did not have the high-gain controller problems that Gen2 had.[10]

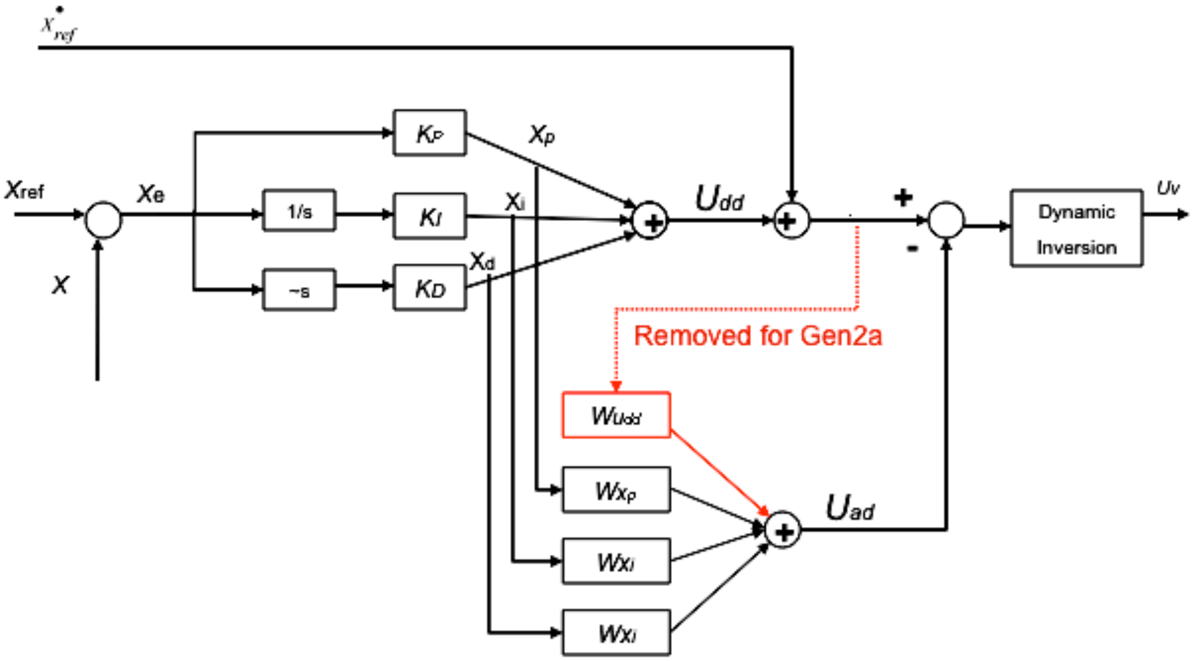


Figure 4. Simplified block diagram of proportional-integral-derivative dynamic inversion adaptive controller for generation 2 (Gen2) and generation 2a (Gen2a).

Perhaps the most detrimental effect of using the output of the PID controller (U_{dd}) as a neural network input is that it is highly correlated with the learning signal (Z). They are both computed as a function of the same error, as shown in Eq. (1) and Eq. (5). As a result, the corresponding neural network weight ($W_{U_{dd}}$) tends to adapt very quickly, even in the presence of very small errors. Furthermore, because the sign of the input (U_{dd}) and the learning signal (Z) are usually the same, the weight will tend to grow in only one direction. As a result, the Gen2 controller incorporated weight limits and large learning signal (Z) dead bands to prevent the weights from becoming too large.

The errors are especially problematic when factors, such as lags caused by time delays, result in persistent errors that cannot be compensated for by the adaptive system. The time delay between the neural networks and the flight computer is four to five frames (in which one frame is 0.0125 s or 80 Hz). Delays become very problematic for any controller but even more problematic for high-gain adaptive systems.[11, 12] The performance of any system is better without large time delays. Both Gen2 and Gen2a algorithms are located on the single-string processor, and therefore both have the same delays. When the pilot commands an input, time delays in the system result in tracking errors that cause the weight ($W_{U_{dd}}$) to grow. If unaccounted for, this weight growth eventually leads to a high-gain effect that causes oscillations. The additional tracking error resulting from the oscillations, coupled with the high correlation between the command input (U_{dd}) and the learning signal (Z), cause the weight ($W_{U_{dd}}$) to grow even faster. This accelerated weight growth eventually leads to larger oscillations that persist even when the pilot is no longer commanding an input. Conversely, the Gen2a controller incorporates a command input ($\dot{x}_{ref}$) that does not include the output of the PID controller. As a result, it does not encounter the same high-correlation problems. Hypothetically, if the weight ($W_{\dot{x}_{ref}}$) of the Gen2a controller were to be passively computed, the error resulting from the oscillations would cause the weight to decrease.[10]

2. *Array-to-Scalar Dead Zones*

Some of the main problems with Gen2 were the high correlation between the pilot inputs and the tracking errors compounded with time delays, as discussed previously. As a result of these problems, the performance was poor during the initial design phase, and "tuning knobs" were incorporated into the Lyapunov adaptation law for each neural network input. This complication led to array dead zones for each axis, as discussed previously. These Lyapunov tuning array gains were specifically designed for A-matrix or B-matrix type failures and were tested in the simulation. The "tuning knob" method does allow for more flexibility during the design phase but was difficult for implementation. The dead zone levels had to be set and tested for a given failure in the simulation and regression-tested for a wide range of failure scenarios. The array dead zones caused some weights to grow while other inputs were inside the dead zone threshold, causing uncoordinated neural network tuning.

Figure 5 illustrates the problem of the array dead zones used in Gen2. The figure shows a typical time history response for a formation flight tracking task, with a jammed left stabilator at -2° from trim, occurring at 7 s, at Mach 0.75 and an altitude of 20,000 ft. In a formation flight task, the pilot starts with moderately aggressive tracking or gross acquisition followed by a fine-tracking task that usually requires high frequency stick inputs. The figure shows the pitch stick and power lever angle required to keep the aircraft in tight formation with the target aircraft. Gross acquisition starts at 37 s and the fine-tracking task starts at 55 s. Figure 6 shows a blowup of the pilot longitudinal stick input (Dep - blue line) and the pitch rate of the airplane. The figure illustrates the PIO trends that occur when the pilot command is out of phase with the aircraft response. The PIO is a major nuisance in control performance and results in comparatively worse pilot CHRs.

During the formation flight task, the dead zone for the roll axis was too large to help the weight converge to a useful size. Figure **7** shows that most of the information going into the first roll axis neural network node is ignored, and most of the adaptation occurs in the first few seconds after the failure (7 to 9 s). Later in the maneuver when the pilot is getting into the PIO, the first roll node sees very little of the input and does very little to change the value of the output. The bottom plot in Fig. 7 shows the node output for the formation flight test of Gen2. Thirteen other dead zones were considered in the implementation of the Gen2 adaptation control laws.

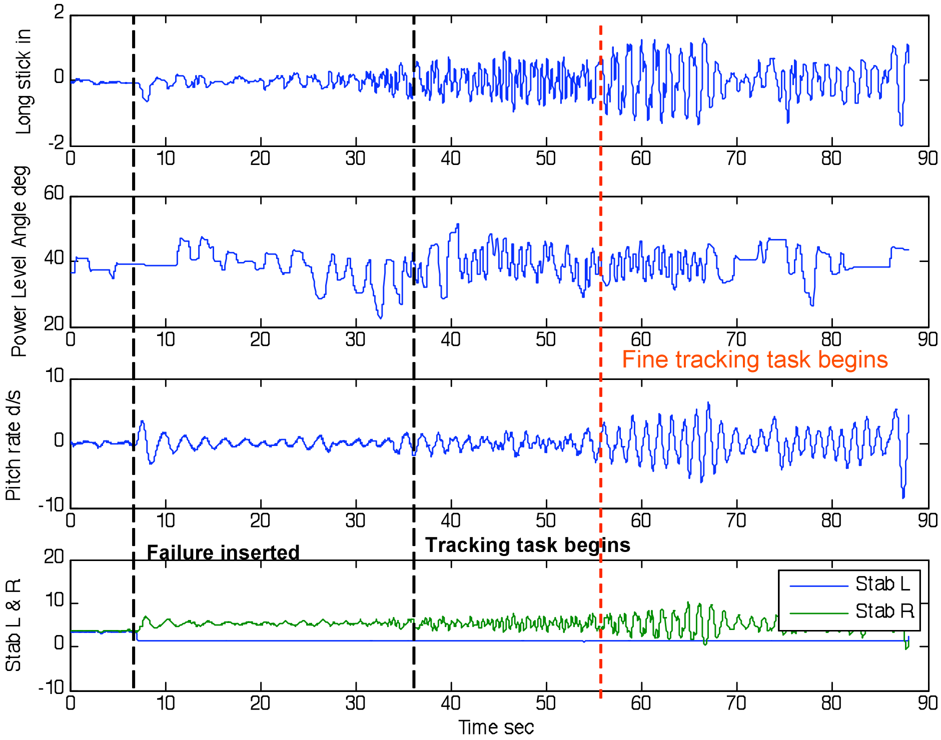


Figure 5. Generation 2 (Gen2) time history for a 1-*g* formation flight with the left stabilator jammed at 7 s at a position of -2° from trim with neural networks active.

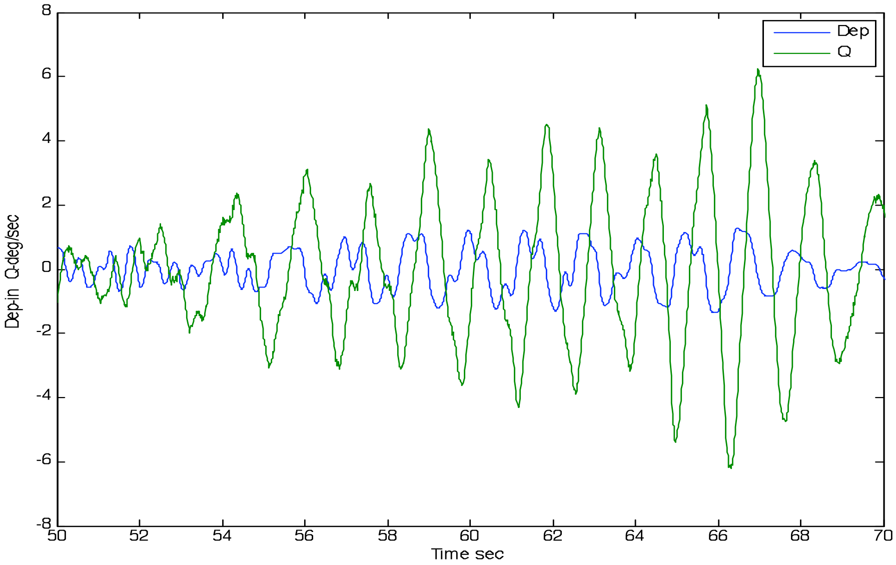


Figure 6. Generation 2 (Gen2) pilot-induced oscillation time history trends of pitch stick (Dep) and pitch rate (*q*).

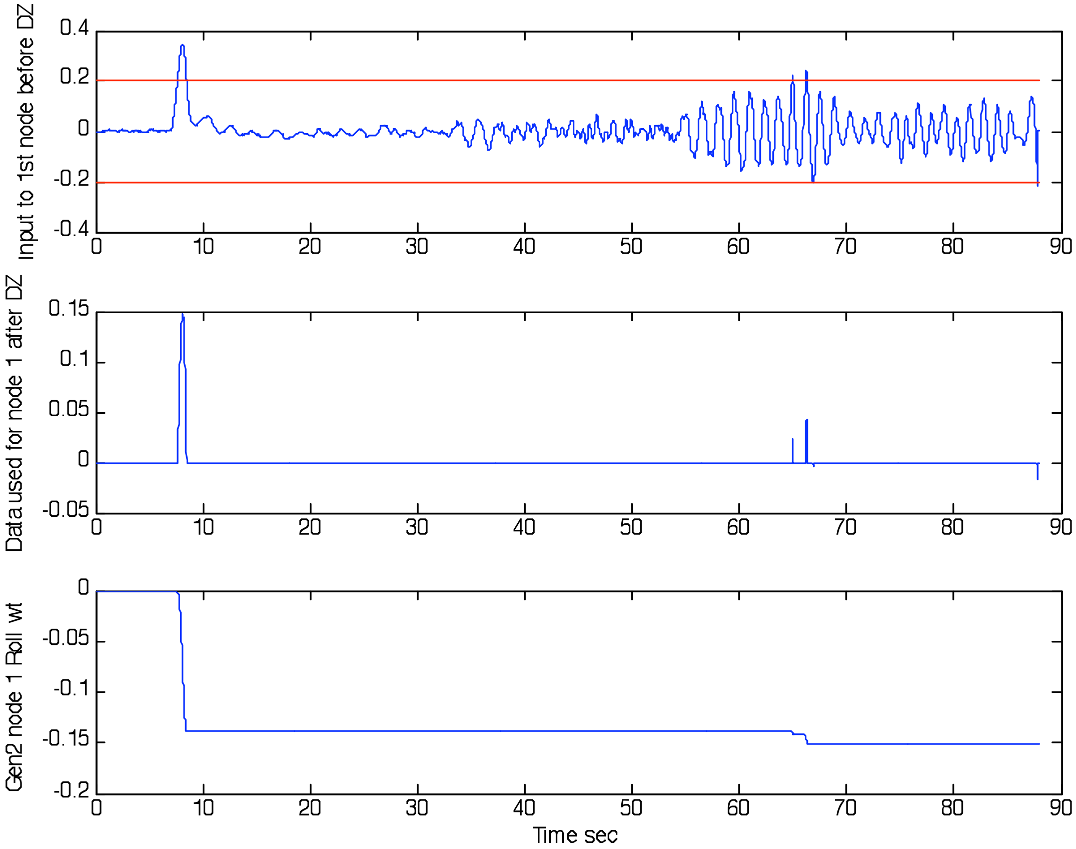


Figure 7 Generation 2 (Gen2) time history of the roll axis neural network node 1 before and after dead zones (red line shows the level of node 1 dead zone).

The dead zone levels were designed to use 120 percent of the healthy aircraft response data. The problem is that some node dead zones sized for a healthy aircraft are not appropriate for a failed airplane. When the axis has mismatched learning, the adaptive system encounters difficulty in driving the errors to zero or a lower minimum. Figure 8 illustrates that the neural network dead zones were not synchronized in the same axis. The figure plots all the roll axis data before and after the dead zones for Gen2. Looking at all the roll axis input nodes, one can see that the first node dead zone cuts out almost all the information, whereas the node 6 dead zone passes almost all the data to the neural network. This failure could have been adjusted with more flight test and simulation studies; however, a simulation should not be designed for a specific failure.

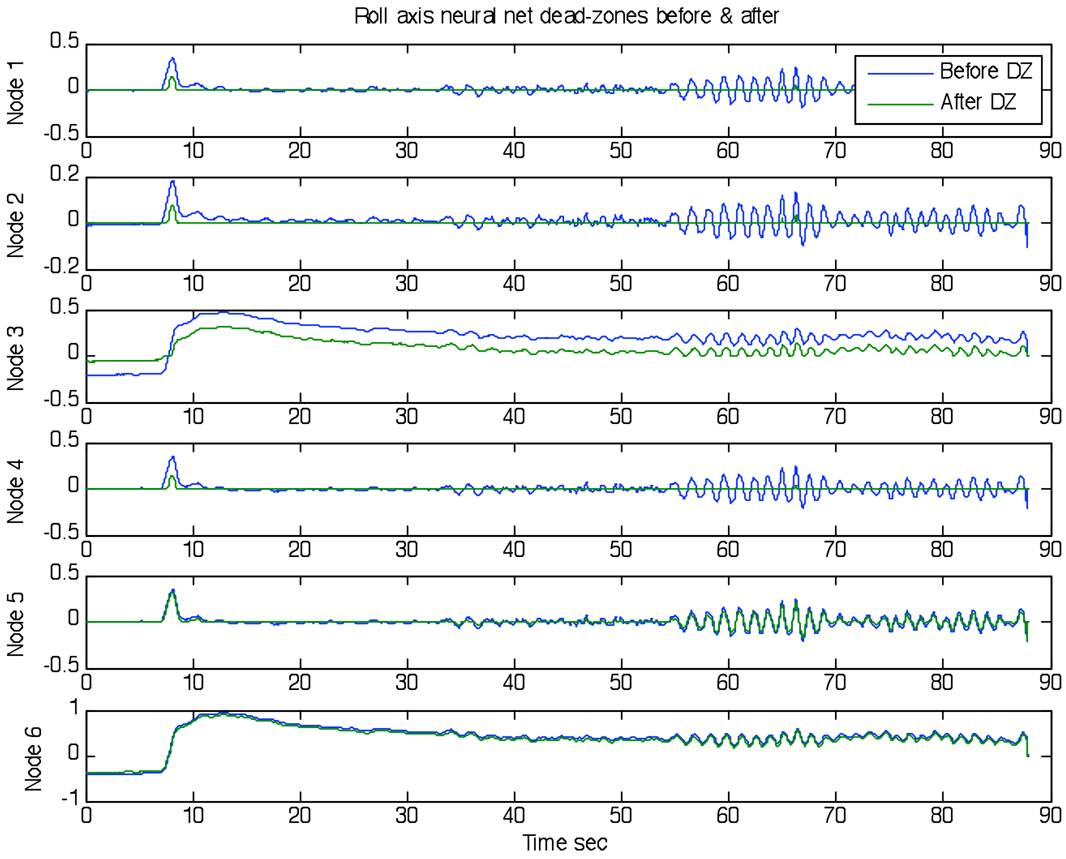


Figure 8. Generation 2 (Gen2) time history of the roll axis neural network nodes 1–6 before and after dead zones.

The inputs from Gen2 were back driven into the Gen2a control system in an open-loop manner as shown in Fig. 9. The figure demonstrates how the scalar dead zone would have had more information to adapt to the pilot and aircraft inputs. The Gen2a roll axis scalar dead zone was determined by simply setting the level of the roll rate error (to which the neural networks did not need to adapt), which was 2 deg/s. The dead zone level was then tested in the simulator before the flight. Figure 9 shows that the dead zones are smaller for Gen2a, and the neural network therefore has to form information to adapt to pilot inputs. A comparison of the last plot in Figs. 7 and 9 (Gen2 and Gen2a, respectively) shows how differently the neural network node changes with Gen2a as opposed to Gen2. Notice how the function is not stair stepped with the Gen2a node compared with the Gen2 node. The Gen2a dead zones are scalar values, one for each neural network axis. As a result of all the changes, the smaller Gen2a dead zones help improve performance.

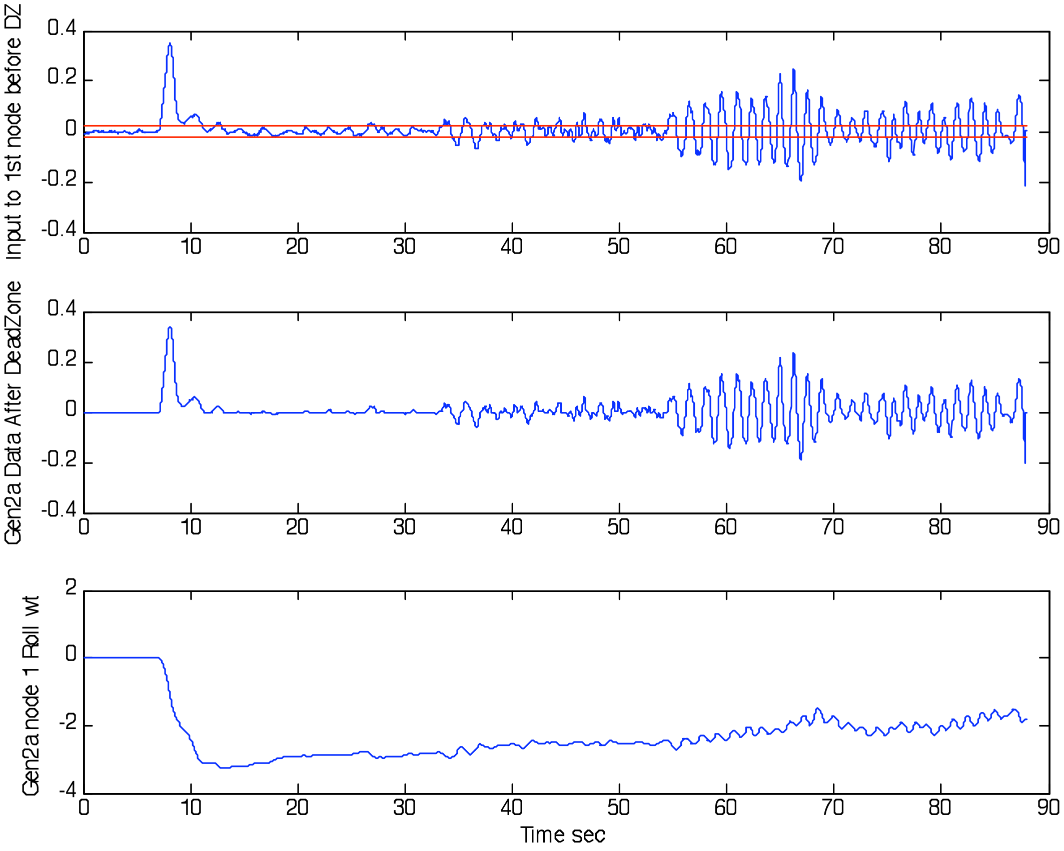


Figure 9. Generation 2a (Gen2a) time history of a scalar dead zone used for the first roll neural network node back-driven with inputs from generation 2 (Gen2).

3. Beta Yaw Axis Controller

As shown in Fig. 3, the Gen2 yaw-damping controller is augmented by computing an adaptive signal (U_{ad}) in a similar manner to the pitch and roll axes. Unlike the pitch and roll axes, however, there is no PID controller to generate the learning signal (Z), as shown in Eq. (5). Instead, the Gen2 controller incorporates a virtual proportional-integral (PI) controller to compute the learning signal from the rudder command, in which

$$Z_r = K_{z_i}\int u_c + K_{z_p} u_c \tag{6}$$

Although this implementation provided some improvement in terms of yaw damping, a fair amount of sideslip angle (β) and lateral acceleration (n_y) were still experienced during B-matrix (left stabilator) failures.

The Gen2a controller uses a different approach, as shown in Fig. 10. In this case, the portion of the feedforward rudder command that computes a reference sideslip angle (β_{ref}) is used as a directional attitude command. A virtual proportional-derivative (PD) controller is then used to compute the learning signal from the error in the sideslip angle (β_e), in which

$$Z_r = K_{z_p}\beta_e + K_{z_d}\dot{\beta}_e \tag{7}$$

Meanwhile, the yaw-damping command is treated as part of the well-damped aircraft, rather than used as an input to the neural network.

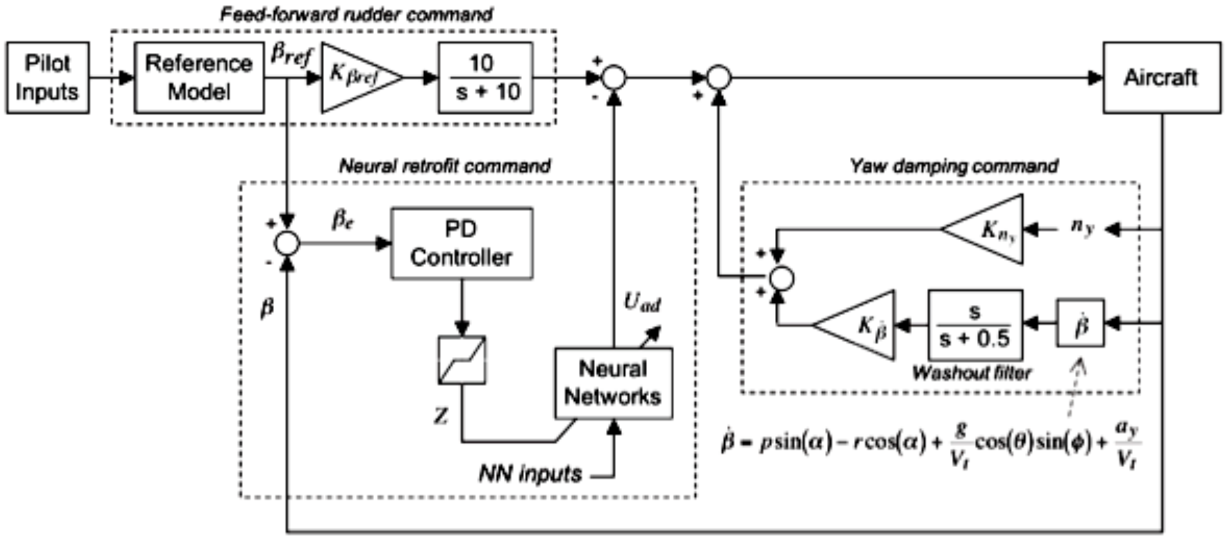


Figure 10. Neural yaw axis modification, which includes angle of sideslip for generation 2a (Gen2a).

4. Reduced Learning Rate Modification

Although the neural network inputs of the Gen2a controller provide significantly more robustness, they also require higher adaptation gains (G) to achieve the same learning rates. The reason for this requirement is that less correlation exists between the neural network inputs (removed U_{dd}) and the learning signal (Z), especially during fine-tracking maneuvers, which normally result in smaller errors. Adaptation gains (G) that are too high, however, can cause learning transients that can lead to controller instability. This effect is especially true during gross-acquisition maneuvers, which normally result in larger errors. As such, the Gen2a controller incorporates a form of normalized learning rates, which reduces the overall rate of adaptation during aggressive maneuvers. This form of normalized learning rates was accomplished by reducing the overall adaptation gain ($\hat{G}$) during large acceleration commands ($\dot{x}_{ref}$), large rate commands (x_{ref}), and periods of low neural network confidence defined by $U_{ad}Z$ (see reference 10 for details). Equation (8) provides a simple representation of the reduced adaptation gain incorporated in Gen2a (not in Gen2). Equation (8) can be tuned by specifying a diagonal matrix (R).[13] Large pilot inputs with high frequency contents along with time delays caused very large tracking errors. The adaptation gain needed to be "tuned down" or "slowed down" during such inputs. Equation (8) reduces *G* when the inputs (*Z*) are large.

$$\hat{G} = \frac{G}{1 + B_a^T R B_a} \tag{8}$$

5. Neural Network Weight Decay Term

A way to simplify a network and avoid overfitting a function is to impose a heuristic method in which the weights should be small. There is no principled reason that such a method of "weight decay" should always lead to improved performance, but it is found to help in most cases.[14] Small weights favor models that are nearly linear. Weight decay adds a penalty term (or a gain 0< K_{decay} < 1) to the error function associated with the weight update law from Eq. (3), $\dot{W} = -G\left(ZB_a + L|Z|W\right)$. The decay is applied to the past weight, as shown in Eq. (9):

$$W(k) = W(k-1) * K\_decay + \overset{\bullet}{W}\, Dt \tag{9}$$

The weight decay term penalizes the weights and causes them to converge to absolute values that are smaller than they otherwise would have been. After each weight update, every weight is "reduced" or "decayed." Weights that are not needed for reducing the error function become smaller, possibly to such a small value that they can be eliminated altogether, thus avoiding the problem of overfitting the data.

III. Flight Results

Flight tests were performed to compare the performance results of no adaptation to the Gen2 and Gen2a controllers under the same failure conditions. The *B*-matrix failure consisted of commanding the left stabilator to the trim point, which simulated a jammed surface. In most flight tests, the aircraft was flying straight and level at Mach 0.7 and an altitude of 20,000 ft. For purposes of brevity, the maneuvers presented in this report consist of roll axis bank-to-bank captures. During the flight test, however, several different types of maneuvers were performed, such as pitch captures, formation flight, and a 3-*g* tracking task. The data presented in this report are representative of the other maneuvers discussed previously.

Figure 11 shows the time history of a bank-to-bank capture maneuver without adaptation and the left stabilator jammed at trim. This maneuver is the reference response from which to judge the other controllers. The goal of the adaptive system (Gen2 or Gen2a) is to improve the flight dynamics compared to that of the nonadaptive system. The figure plots bank angle, roll rate, roll reference (p_c), roll error (p_e), and the neural network roll axis output signal, Sigpi-P. As shown, the maximum roll axis error is approximately +/-14 deg/s with no neural network signal. Figure 12 shows the Gen2 time history of a bank-to-bank capture maneuver. The Gen2 maximum roll axis error is approximately the same or worse than that of the nonadaptive controller shown in Fig. 11. (Not all of the Gen2 results are worse than the no–neural network results; see reference 4 for more information.)

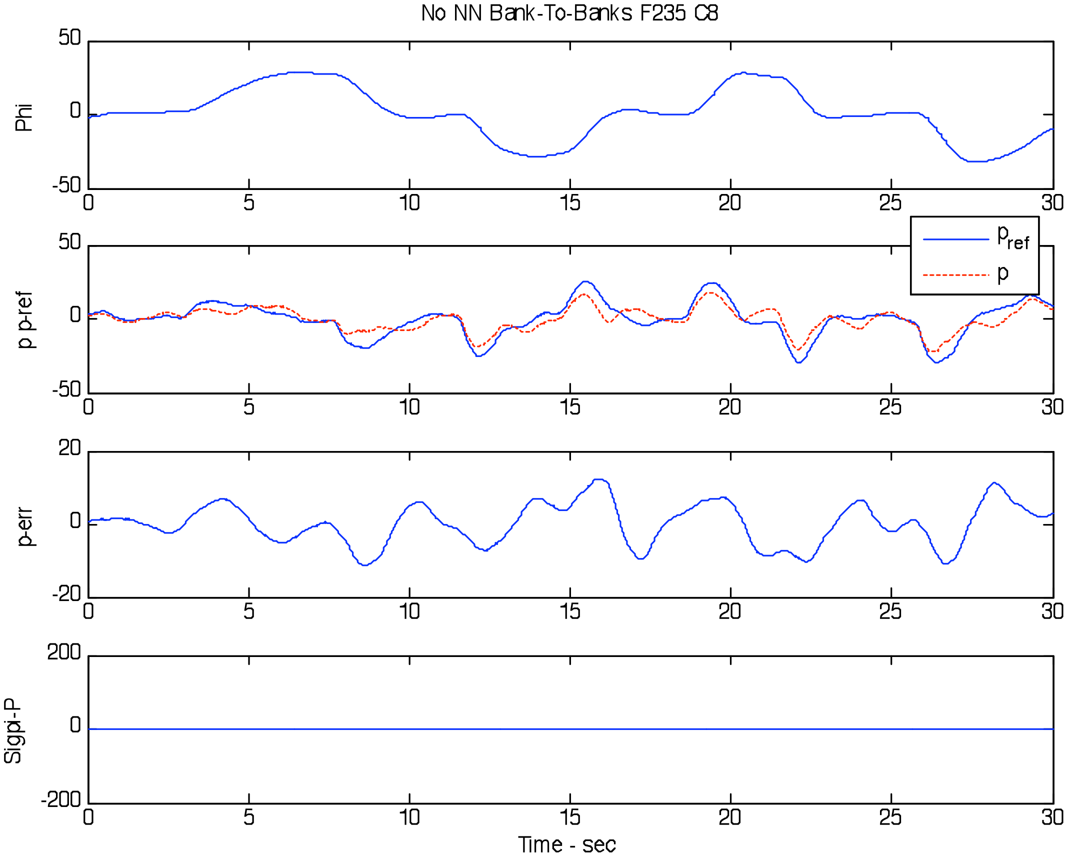


Figure 11. Time history of a bank-to-bank capture maneuver without adaptation and the left stabilator jammed at trim.

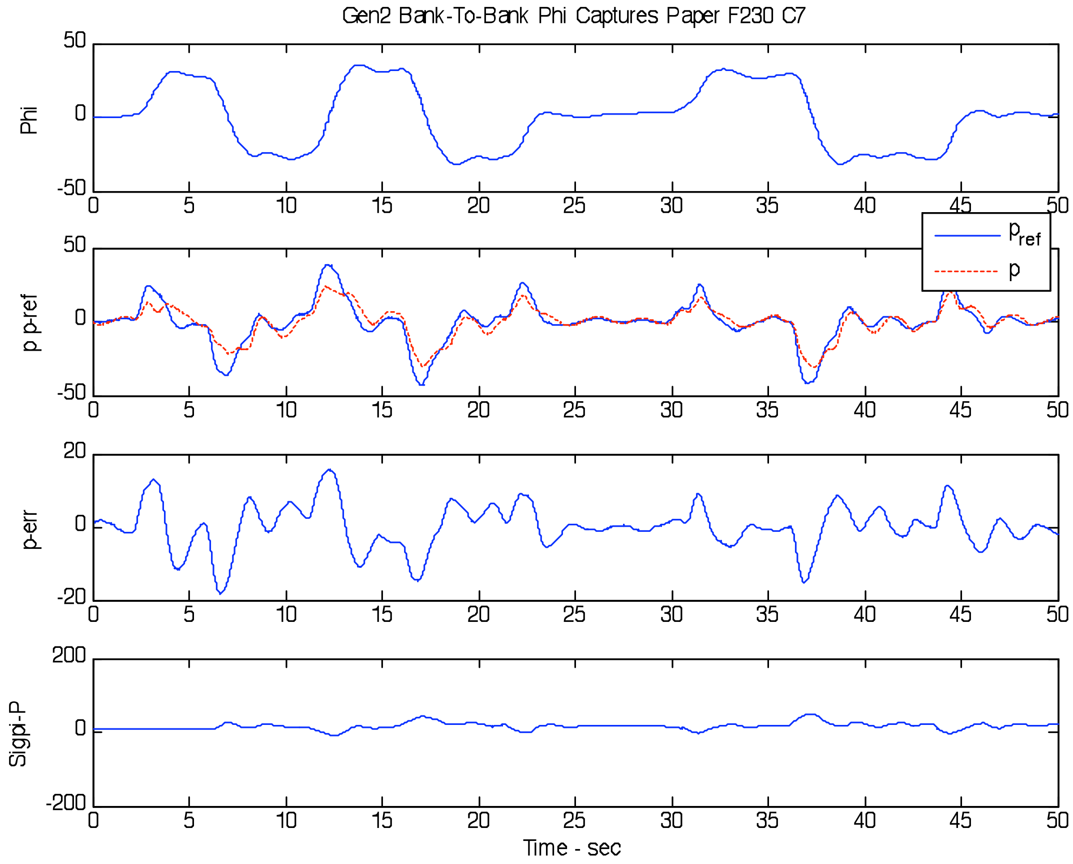


Figure 12. Generation 2 (Gen2) time history of a bank-to-bank capture maneuver with the left stabilator jammed at trim.

Figure 13 shows the time history of the Gen2a response. The roll error is much smaller compared to those of the Gen2 or nonadaptation test cases, with a maximum roll rate error of 7 deg/s. The neural network had to contribute more control power to the system (the Sigpi-P) as shown in the last plot in Fig. 13. This signal is much greater than the Sigpi-P plot for Gen2. The closer or tighter tracking of commands of Gen2a compared with Gen2 can be easily seen in the p, p_{ref} plots in Figs. 12 and 13. The CHRs improved, because the tracking error was reduced. In the case of a failed stabilator at trim without adaptation, the pilots gave the fine-tracking formation flight task an average CHR of 4. The same task with Gen2 adaptation resulted in an average CHR of 5. The pilots gave the fine-tracking task for Gen2a an average CHR of 3. See table 2 for a summary of the CHRs.

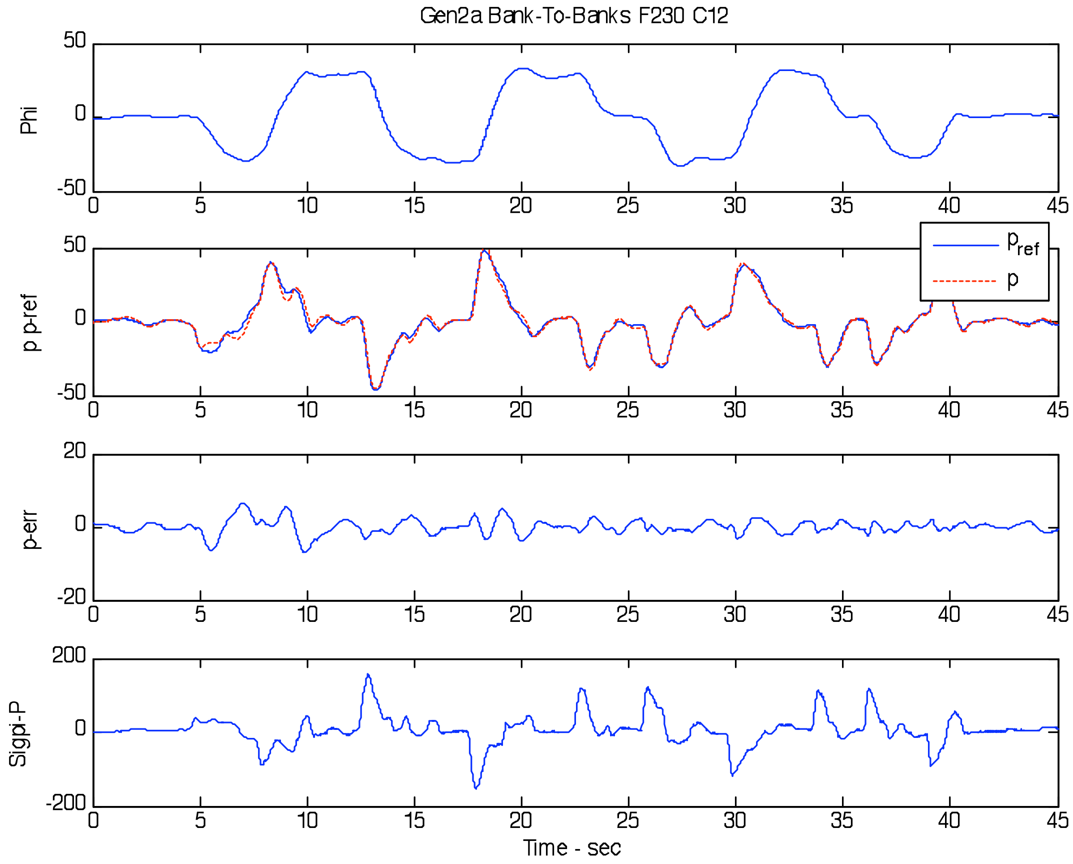


Figure 13. Generation 2a (Gen2a) time history of a bank-to-bank capture maneuver with the left stabilator jammed at trim.

Table 2. Average Cooper-Harper Ratings for the fine-tracking task with the left stabilator jammed at the trim position.

No adaptation	Gen2	Gen2a
4	5	3

Because a pilot is not able to consistently apply the same inputs, performance comparison becomes difficult. When a statistical-based evaluation approach is used, determination of a better (lower error) tracking controller is possible. The performance of the controllers (with neural networks inactive and active, Gen2 or Gen2a) has been evaluated by means of trajectory tracking parameters, such as mean and standard deviation of roll rate error and pitch rate error. Other evaluation parameters can include control surface activity such as deflections of canards, stabilators, rudders, and ailerons, and pilot workload such as longitudinal stick, lateral stick, pedal, and throttles.[2] Figures 14 and 15 present the roll rate and pitch rate statistical error comparison, respectively, of no neural networks, Gen2, and Gen2a. As shown in Figs. 14 and 15, Gen2a outperformed both Gen2 and the case of no adaptation.

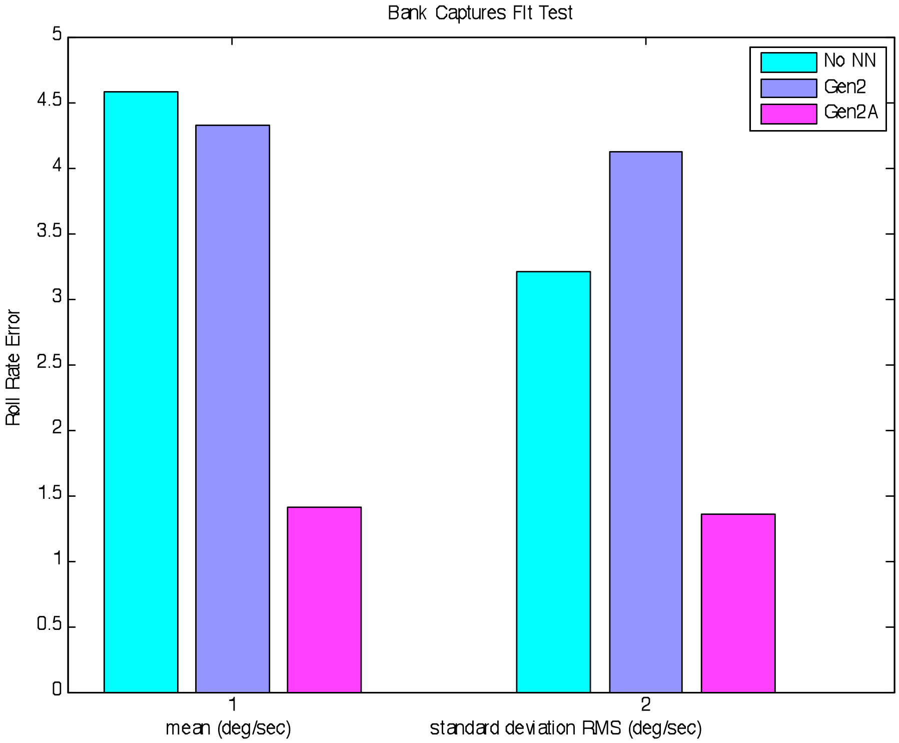


Figure 14. Roll rate error mean and root mean square (RMS) standard deviation for no neural networks, generation 2 (Gen2), and generation 2a (Gen2a), for the bank-to-bank capture manuevers.

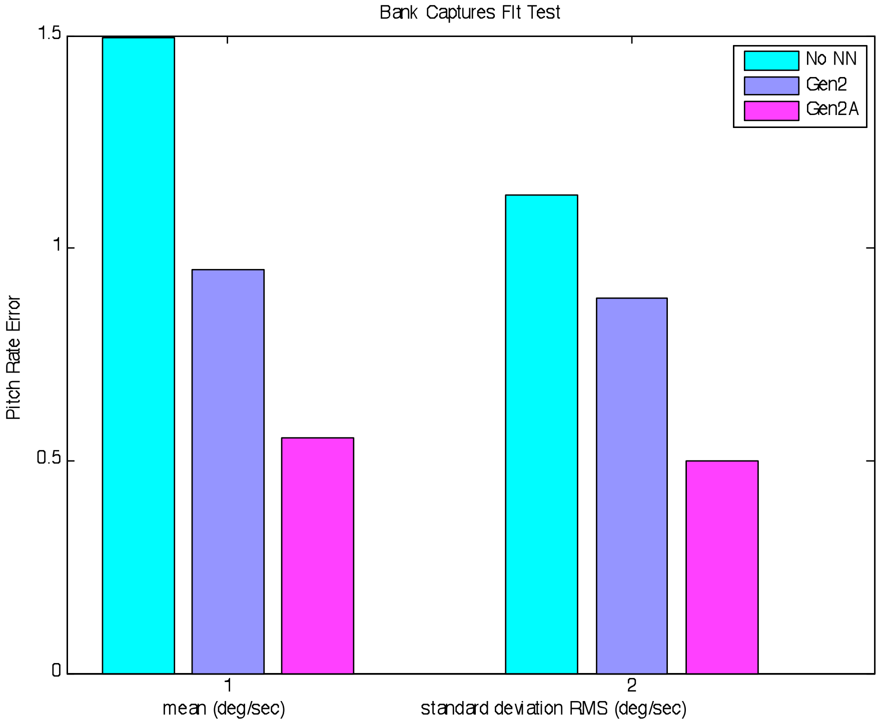


Figure 15. Pitch rate error mean and root mean square (RMS) standard deviation for no neural network, generation 2 (Gen2), and generation 2a (Gen2a) for the bank-to-bank capture manuevers.

IV. Conclusion

A neural network–based approach for directly adapting to aerodynamic changes resulting from damage or failures was recently flight tested on a modified F-15 aircraft. Initial flight test results demonstrated that the adaptation provided improvements in the presence of damage but resulted in a tendency of cross-coupled pilot-induced oscillations in the presence of control surface failures. As a result, potential redesign topics were investigated, which led to the development and testing of several enhancements.

These enhancements included the selection of alternate neural network inputs, which bypassed high-correlation and high-gain deficiencies that were identified in the previous implementation, referred to as "generation 2." This bypass allowed the neural networks to focus on addressing inversion error, without simply applying more gain. These enhancements required the implementation of a technique for normalizing learning rates to achieve the same rate of adaptation. To improve directional performance, a method of implementing a neural yaw-damping retrofit was also investigated using sideslip angle. Weight decay was added for weight protection, and scalar dead zones were added for simplicity. The flight results showed that generation 2a outperformed both generation 2 and the nonadaptive controller for *B*-matrix failures.

Acknowledgments

This research was performed under the Integrated Resilient Aircraft Control (IRAC) project, as part of the Aviation Safety Program (AvSP), within the NASA Aeronautics Research Mission Directorate (ARMD).

References

[1] Hageman, J.J., Smith, M.S., and Stachowiak, S., "Integration of Online Parameter Identification and Neural Network for In-Flight Adaptive Control," NASA/TM-2003-212028, 2003.

[2] Burken, J.J., Williams-Hayes, P., Kaneshige, J.T., and Stachowiak, S.J., "Reconfigurable Control with Neural Network Augmentation for a Modified F-15 Aircraft," NASA/TM-2006-213678, 2006.

[3] Cooper, G.E., and Harper, R.P., Jr., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities," NASA TN D-5153, 1969.

[4] Bosworth, J.T., and Williams-Hayes, P.S., "Flight Test Results from the NF-15B Intelligent Flight Control System (IFCS) Project with Adaptation to a Simulated Stabilator Failure," AIAA 2007-2818, May 2007.

[5] Bomben, C.R., Smolka, J.W., Bosworth, J.T., Williams-Hayes, P.S., Burken, J.J., Larson, R.R., Buschbacher, M.J., and Maliska, H.A., "Development and Flight Testing of a Neural Network Based Flight Control System on the NF-15B Aircraft," *2006 Report to the Aerospace Profession, Fiftieth SETP Symposium Proceedings*, The Society of Experimental Test Pilots, Sept. 2006, pp. 214–240.

[6] Bosworth, J.T., "Flight Results of the NF-15B Intelligent Flight Control System (IFCS) Aircraft with Adaptation to a Longitudinally Destabilized Plant," AIAA 2008-6985, Aug. 2008.

[7] Calise, A.J., Lee, S., and Sharma, M., "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft," AIAA-98-4108, Aug. 1998.

[8] Rysdyk, R.T., and Calise, A.J., "Fault Tolerant Flight Control Via Adaptive Neural Network Augmentation," AIAA-98-4483, Aug. 1998.

[9] *Flying Qualities of Piloted Aircraft*, U.S. Department of Defense, MIL-STD-1797A, June 1995.

[10] Kaneshige, J., and Burken, J., "Enhancements to a Neural Adaptive Flight Control System for a Modified F-15 Aircraft," AIAA 2008-6986, Aug. 2008.

[11] Sharma, M., and Lavretsky, E., "Application and Flight Testing of an Adaptive Autopilot on Precision Guided Munitions," AIAA 2006-6568, Aug. 2006.

[12] Cao, C., Patel, V.V., Reddy, C.K., Hovakimyan, N., Lavretsky, E., and Wise, K., "Are Phase and Time-delay Margins Always Adversely Affected by High-Gain?" AIAA 2006-6347, Aug. 2006.

[13] Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft," AIAA 2006-6049, Aug. 2006.

[14] Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification*, 2nd ed., John Wiley & Sons, Inc., New York, 2001, p. 314.

Flight Test Comparison of Different Adaptive Augmentations for Fault Tolerant Control Laws for a Modified F-15

John Burken
Curt Hanson &
James A. Lee
NASA, Dryden Flight Research Center, Edwards CA, 93523

John Kaneshige
NASA Ames Research Center, Moffett Field, CA, 94035

AIAA Infotech@Aerospace Conference and Exhibit
6 - 9 Apr 2009
Sheraton Seattle Hotel
Seattle, Washington

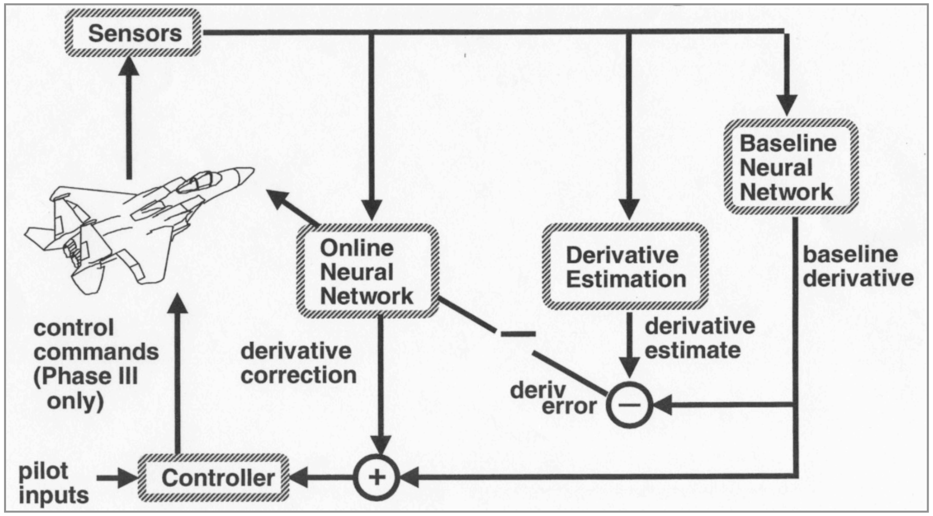
Outline

- Introduction
  - Background
  - Challenges
  - Simulated Damage and Failures
- System Overview
  - Neural Flight Controller
  - Neural Yaw Damping Controller
- Gen 2 Flight Tests
  - Canard Failure Test
  - Stabilator Failure Tests
  - Lessons Learned
- Gen 2a Enhancements
  - Alternate NN Inputs
  - Normalized Learning Rates
  - Neural Yaw Damping Retrofit
  - Weight Decay Term
- Flight Tests
  - Canard Failure Tests
  - Stabilator Failure Tests
  - Robustness Tests
- Conclusions

Background

The Intelligent Flight Control (IFC) research effort was established to develop next generation adaptive flight control technologies that can accommodate changes in aircraft dynamics due to damage or failures.

- **Gen 1 - Indirect Adaptation**
  - **Parameter Identification method**
  - **Dynamic Cell Structure (DCS) neural network**
  - **Stochastic Optimal Feedforward and Feedback Technology (SOFFT) controller**
- **Gen 2 - Direct Adaptation**
  - **Dynamic Inversion based controller**
  - **Sigma-Pi neural networks**
  - **Lyapunov stability proof guarantees boundedness of the tracking error and network weights**
- **Gen 2a - Follow-on to Gen 2**
  - **Design and evaluate enhancements to address lessons learned**



F-15 IFCS Project Goals

- Demonstrate Control Approaches that can Efficiently Optimize Aircraft Performance in both Normal and Failure Conditions [A] & [B] failures.

- Advance Neural Network-Based Flight Control Technology for New Aerospace Systems Designs with a Pilot in the Loop

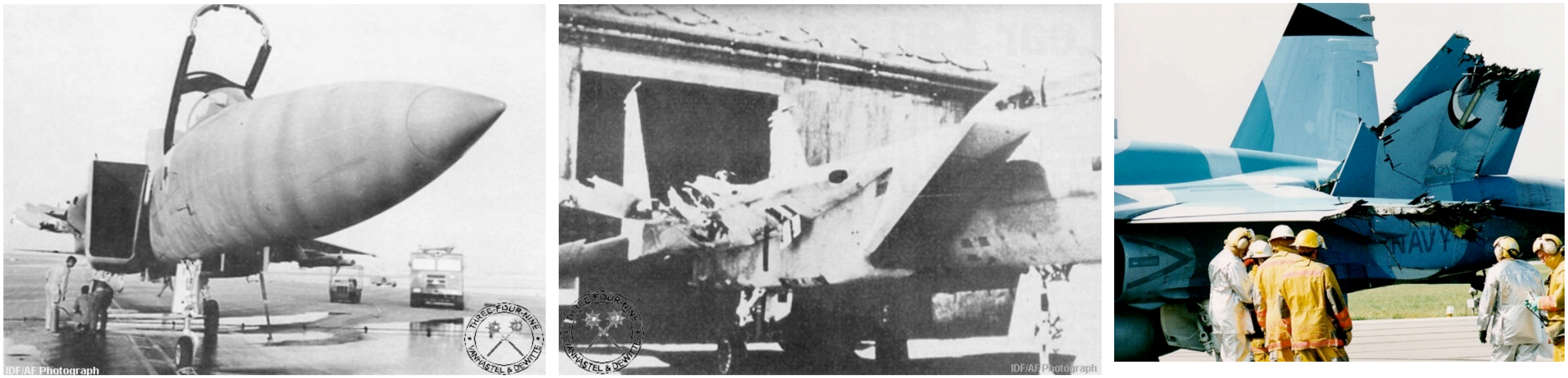
Flight Experiment

- Assess Handling Qualities of Gen II Controller without Adaptation
- Activate Adaptation and Assess Changes in Handling Qualities
- Introduce Simulated Failures
  - Control Surface Locked (“B Matrix Failure”)
  - Angle of Attack to Canard Feedback Gain Change (“A Matrix Failure”)
- Report on Experience with a Neural Network Based Flight Control System

F-15 Intelligent Flight Control Systems

- Motivation / Problem Statement {The Big Picture}
  - Land a damaged airplane or, return to a safe ejection site.

- General Goals & Objectives
  - Flight evaluation of neural net software.
  - Increased survivability in the presence of failures or aircraft damage.
    - Increase your boundary of a flyable airplane.
    - Increase your chances to see another day.
    - Increase your chances to continue the mission.


IDF/AF Photograph

IDF/AF Photograph

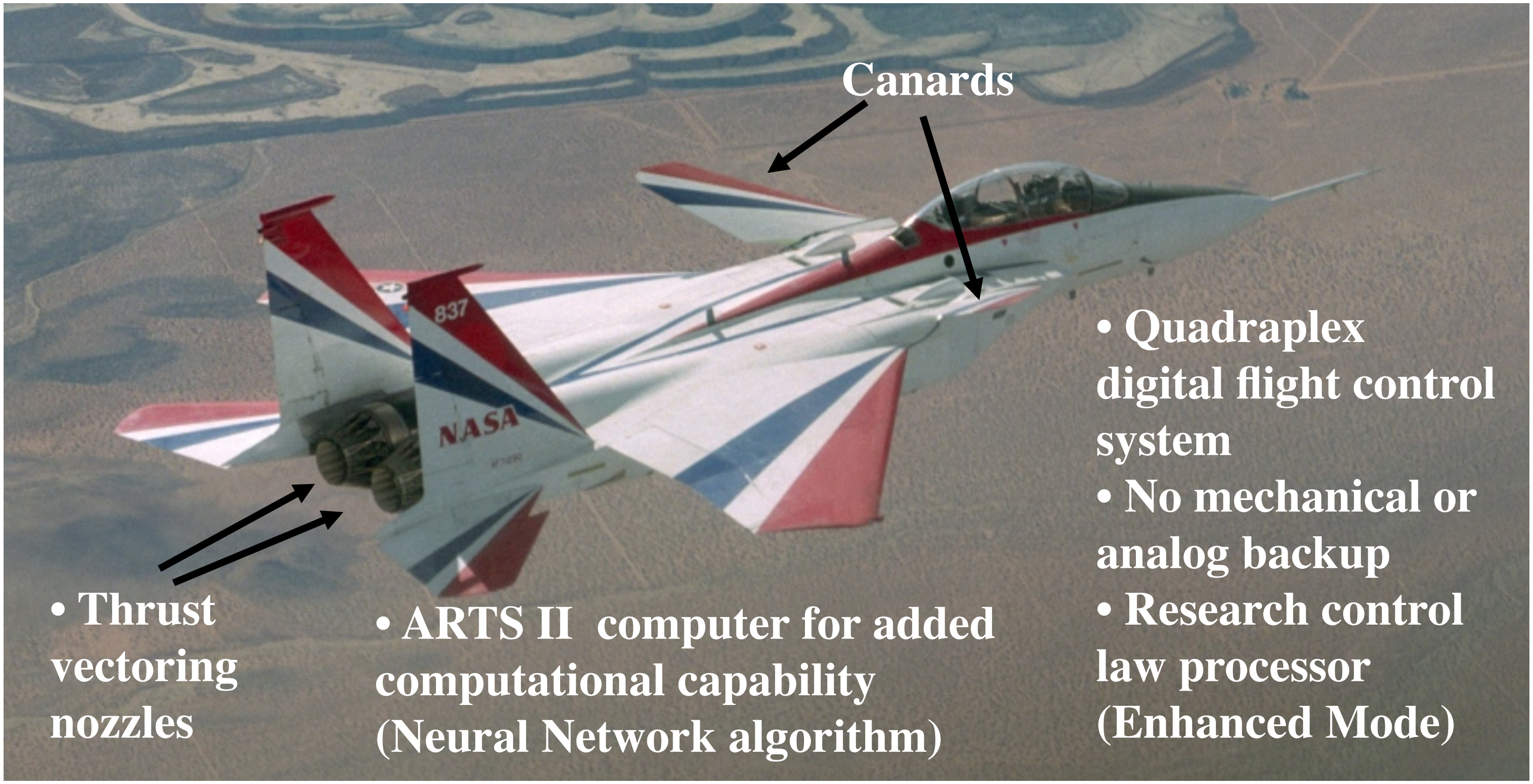
These are survivable accidents

Outline

- Introduction
  - Background
  - ✓ Simulated Damage and Failures
- System Overview
  - Neural Flight Control System
  - Neural Yaw Damping Controller
- Gen 2 Analysis
  - Flight Test Results
  - NN Input Analysis
- Gen 2a Enhancements
  - Alternate NN Inputs
  - Normalized Learning Rates
  - Neural Yaw Damping Retrofit
- Test Results
  - Performance Tests
  - Robustness Tests
- Conclusions

NASA NF-15B

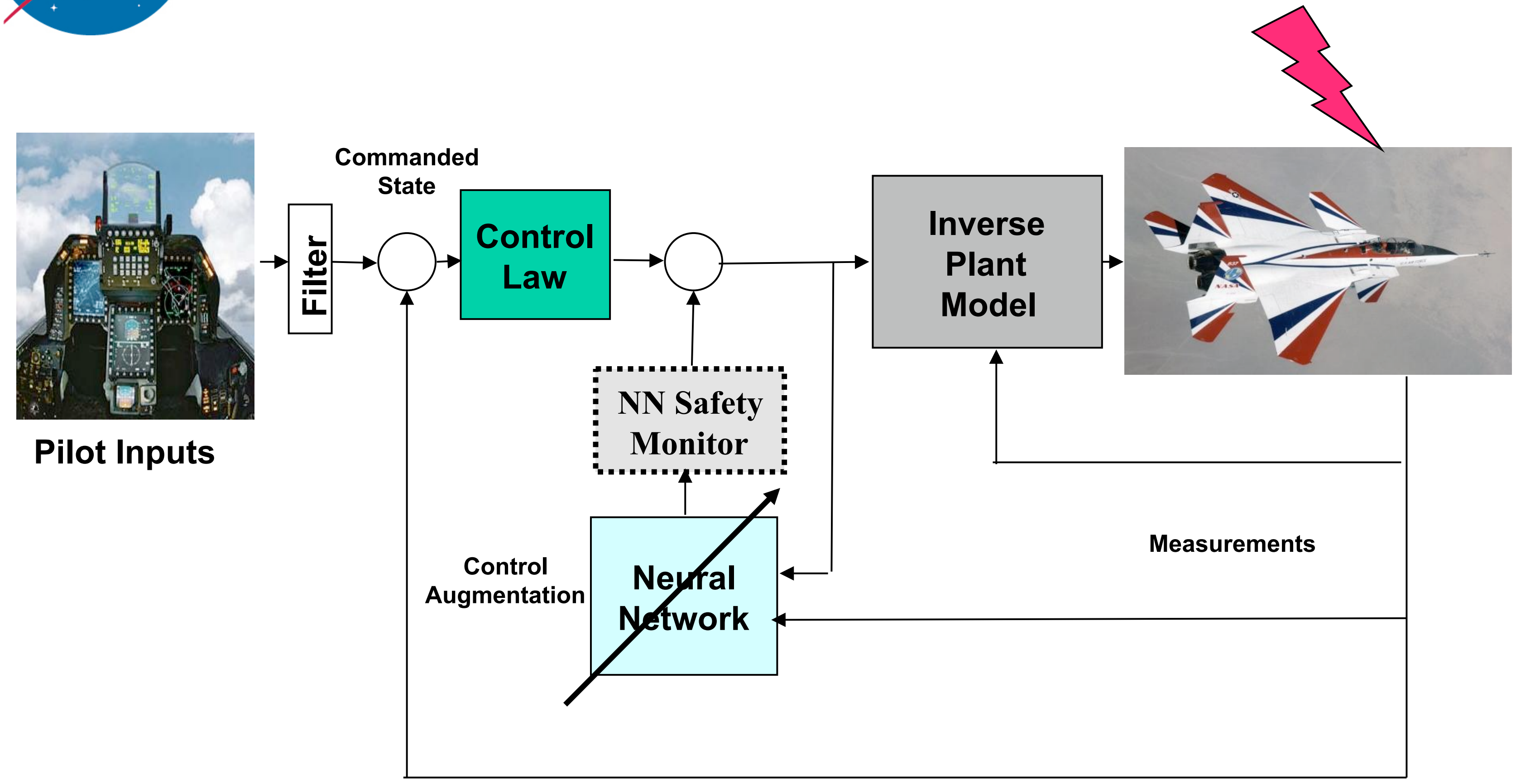
Extensively modified F-15 airframe



Canards

- Thrust vectoring nozzles
- ARTS II computer for added computational capability (Neural Network algorithm)
- Quadraplex digital flight control system
- No mechanical or analog backup
- Research control law processor (Enhanced Mode)

IFCS Overview



Pilot Inputs

Filter

Commanded State

Control Law

NN Safety Monitor

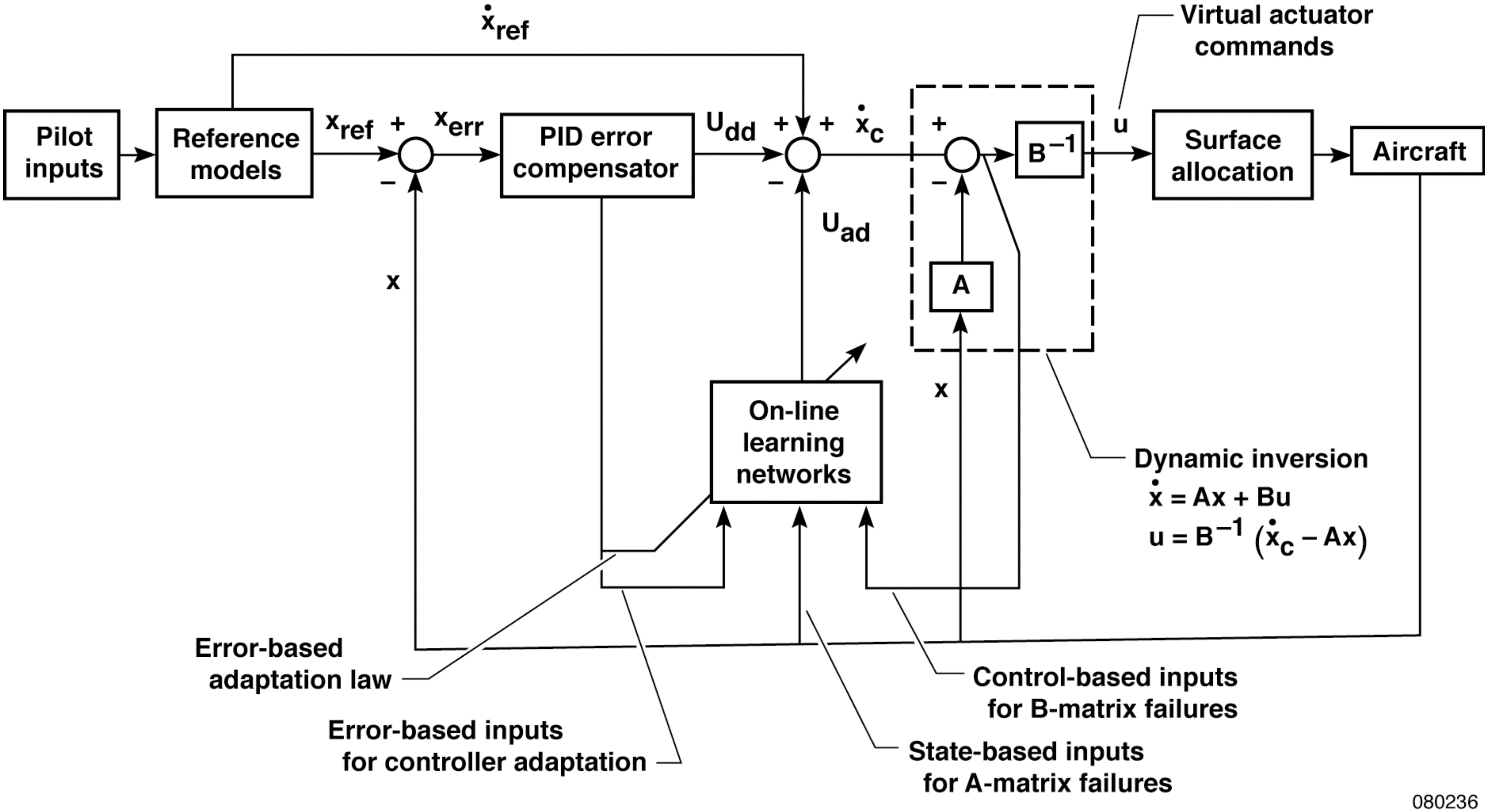
Inverse Plant Model

Control Augmentation

Neural Network

Measurements

Direct Adaptive Control Architecture



$\dot{x}_{ref}$
Pilot inputs
Reference models
x_{ref}
+
−
x_{err}
PID error compensator
U_{dd}
+
+
−
$\dot{x}_c$
+
−
B^{-1}
u
Virtual actuator commands
Surface allocation
Aircraft
x
U_{ad}
A
x
On-line learning networks
Dynamic inversion
$\dot{x} = Ax + Bu$
$u = B^{-1}(\dot{x}_c - Ax)$
Error-based adaptation law
Error-based inputs for controller adaptation
State-based inputs for A-matrix failures
Control-based inputs for B-matrix failures
080236

Direct Adaptive Neural Network

Neural Network Design and Implementation

Background : Historical Note

- Neural Networks are a subset of Adaptive Control.
- Adaptive Control Research Started in the early 1950's.
  - Auto-Pilot work (non-Neural Network).
- Research Diminished due to the crash of X-15.
  - Reference: Eugene Lavretsky, "Adaptive Control: Introduction, Overview, and Applications."

General Neural Network Problem Statements Plus Others

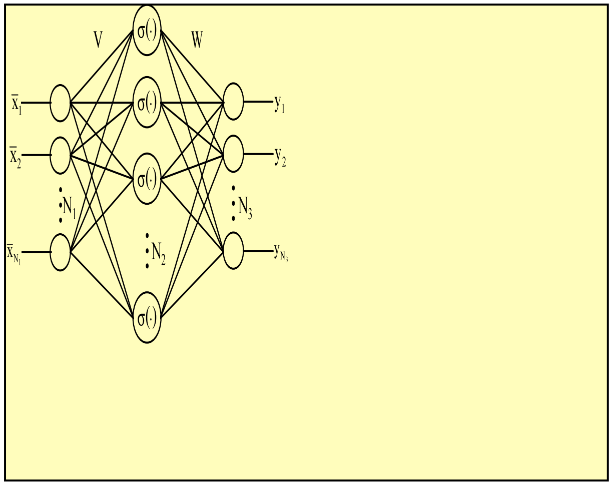
- Why Use a Neural Network?
- How much do Neural Networks help a controller?
- Why Use Dynamic Inverse Control?
- How much do Neural Networks cost w.r.t. compute power?
- How can we certify a Neural Network?
- Some of these questions are NOT answered in this presentation

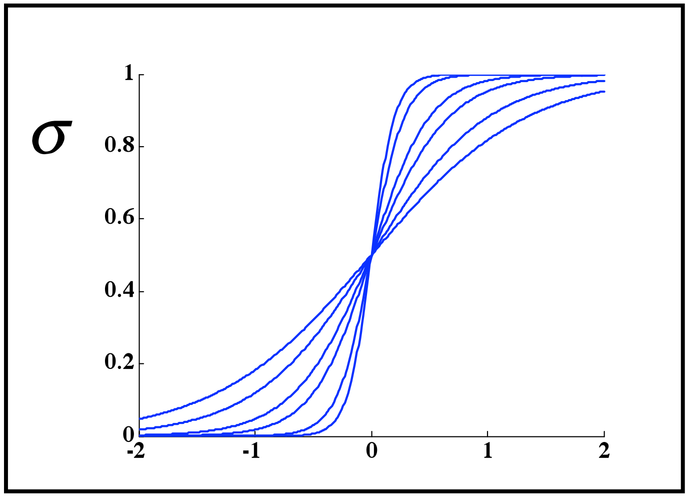
Why Neural Networks?

Neural Networks are Universal Approximators

Minimizes a H^2 norm

They permit a nonlinear parameterization of uncertainty





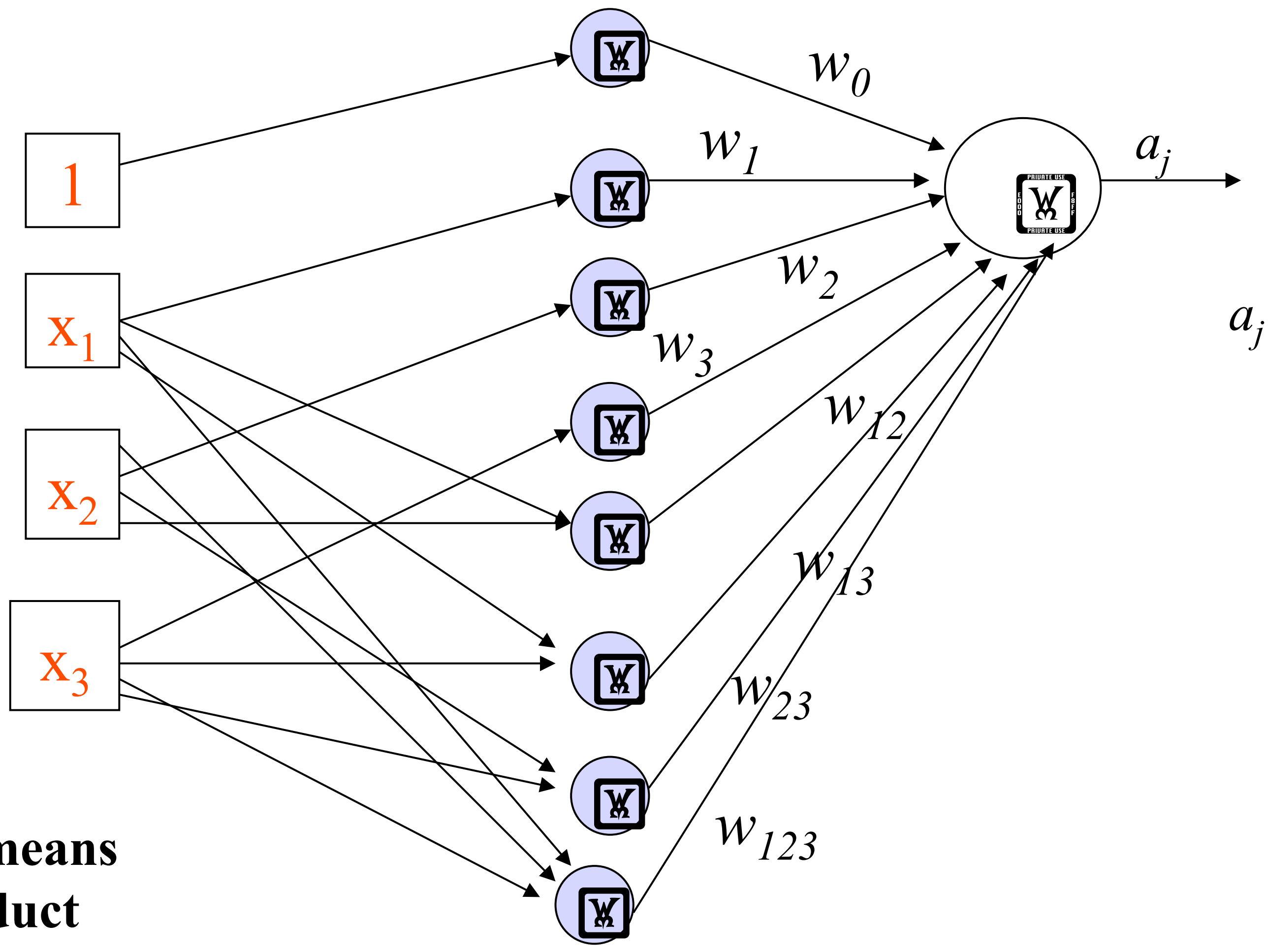
$$y = f(x) = W\sigma(Vx) + \varepsilon(x)$$

$$|\varepsilon(x)| < \varepsilon^* \;\; \forall \;\; x \varepsilon \Omega$$

$$\dot{W} = -\left[\left(\sigma - \sigma' V^T \bar{x}\right)\eta + \kappa \|e\| W\right]\Gamma_W$$

Multiple neurons

For 1 neuron with 3 inputs:



$$
\begin{aligned}
a_j = \; & w_0 1 && + \\
& w_1 x_1 && + \\
& w_2 x_2 && + \\
& w_3\, x_3 && + \\
& w_{12}\, x_1 x_2 x && + \\
& w_{13}\, {}_1 x_3 x_2 && + \\
& w_{23}\, x_3 && + \\
& w_{123} x_1 x_2 x_3 &&
\end{aligned}
$$

▣ means product

Failures Investigated

2 groups of failures are “common” among aircraft mishaps/crashes.

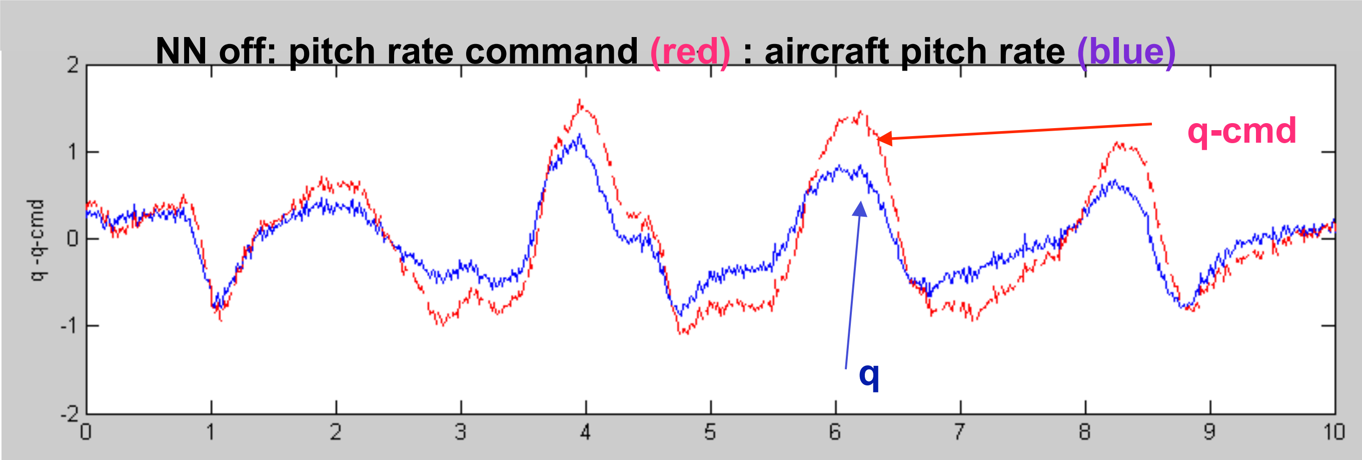
- Aerodynamic Failures (A Matrix problems / lost aero surfaces, bent wings)
  - Canard Failure (0.8 to -1.75 multiplier)

- Control Failures (B Matrix problems / jammed control surfaces)

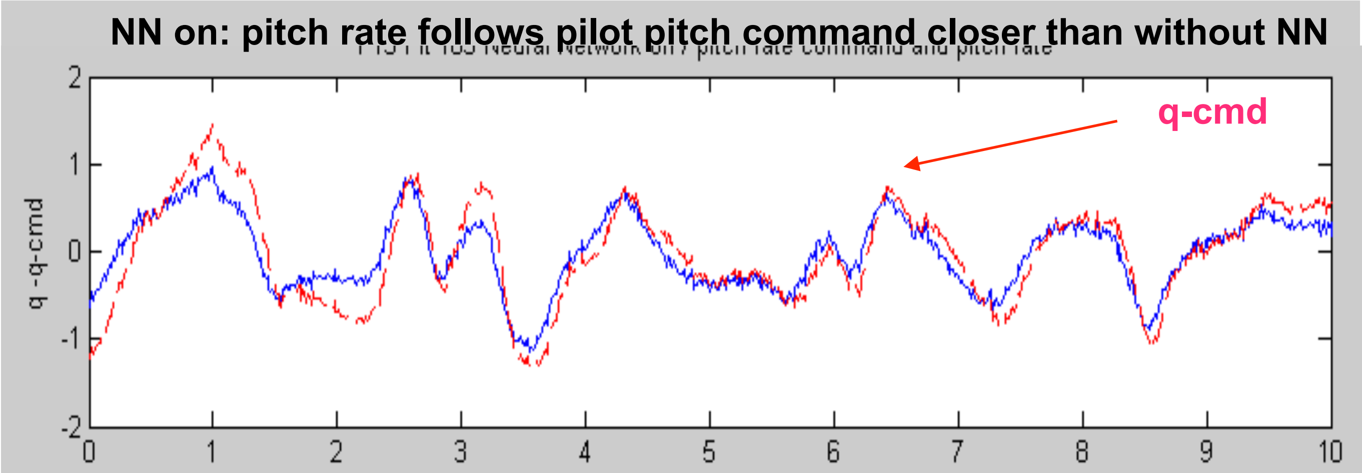
Outline

- ✓ Introduction
  - Background
  - Simulated Damage and Failures
- ✓ System Overview
  - Neural Flight Control System
  - Neural Yaw Damping Controller
- ✓ Gen 2 Analysis
  - Flight Test Results
  - NN Input Analysis
- ✓ Gen 2a Enhancements
  - Alternate NN Inputs
  - Normalized Learning Rates
  - Neural Yaw Damping Retrofit
- Test Results
  - Performance Tests
  - Robustness Tests
- Conclusions

Flight Test Results: Canard Failure



NN off: pitch rate command (red) : aircraft pitch rate (blue)



NN on: pitch rate follows pilot pitch command closer than without NN

Neural networks provided a significant improvement in tracking performance during formation flight maneuvers with a 0.2 canard multiplier.

Canard Failure (continued)

(1) Formation flight with no neural networks. (2) Formation flight with neural networks. (3) 3g tracking with neural networks. "It seems to be much more predictable with the neural networks on."

QuickTime™ and a
YUV420 codec decompressor
are needed to see this picture.

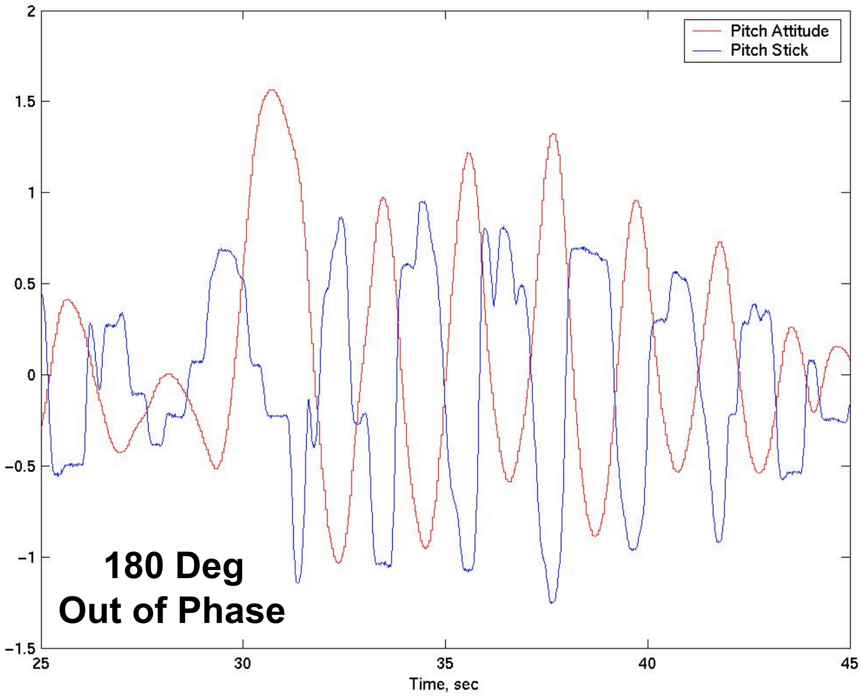
Neural Network Flight Test Video

[A] matrix failure with adaptation on and off during a 1 g formation flight

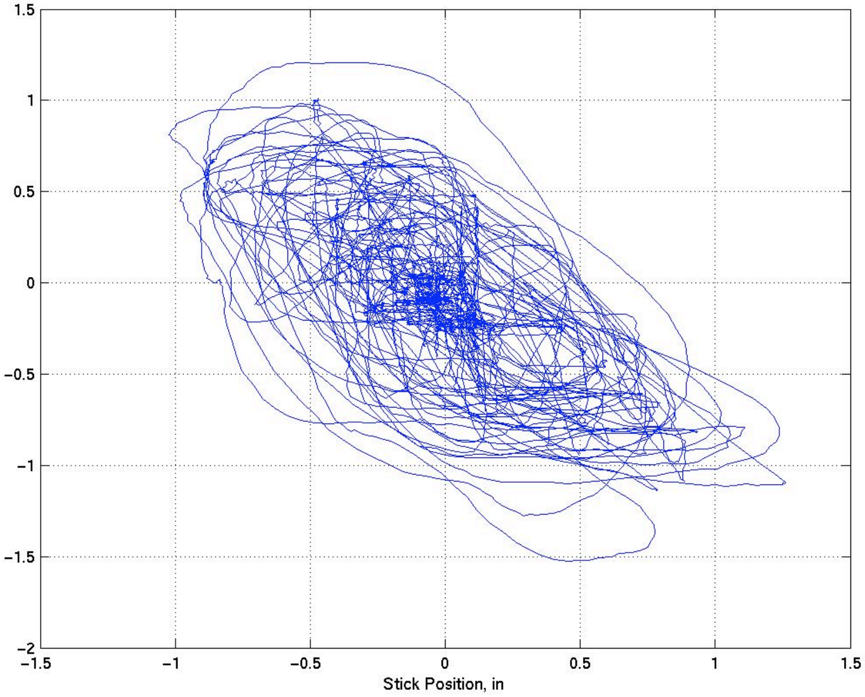
(1) Formation flight with no neural networks. (2) Formation flight with neural networks. (3) 3g tracking with neural networks. "It seems to be much more predictable with the neural networks on."

Flight Test Results: Stabilator Failure

Neural networks provided no significant improvement in pitch rate response, when failing the left stabilator at +4 degrees from trim, and resulted in a tendency for pilot-induced oscillations (PIOs). Cross-coupling effects also remained a problem.



180 Deg Out of Phase



Stabilator Failure (continued)

(1) Formation flight with neural networks.

QuickTime™ and a
H.264 decompressor
are needed to see this picture.

Outline

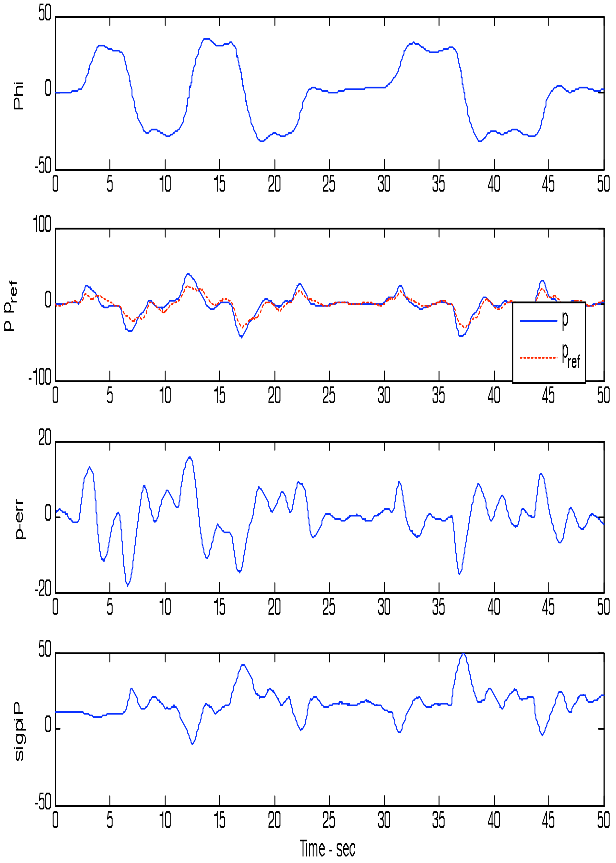
- ✓ Introduction
  - Background
  - Simulated Damage and Failures
- ✓ System Overview
  - Neural Flight Control System
  - Neural Yaw Damping Controller
- ✓ Gen 2 Analysis
  - Flight Test Results
  - NN Input Analysis
- Gen 2a Enhancements
  - Alternate NN Inputs
  - Normalized Learning Rates
  - Neural Yaw Damping Retrofit
- Test Results
  - Performance Tests
  - Robustness Tests
- Conclusions

Gen2a Enhancements

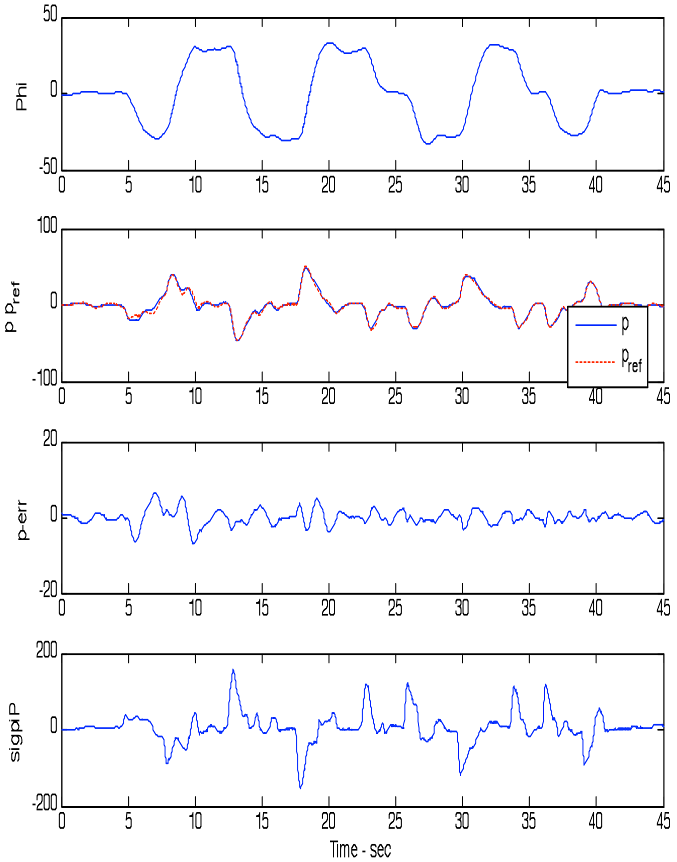
- Changed inputs to adaptation.
- Normalized Learning Rates.
- Went to scalar learning rates per axes.
- Add a beta feedback term.
- Add weight decay.

Gen2 & Gen2a Sigma Pi Flight Results

Gen2 Results



Gen2a Results



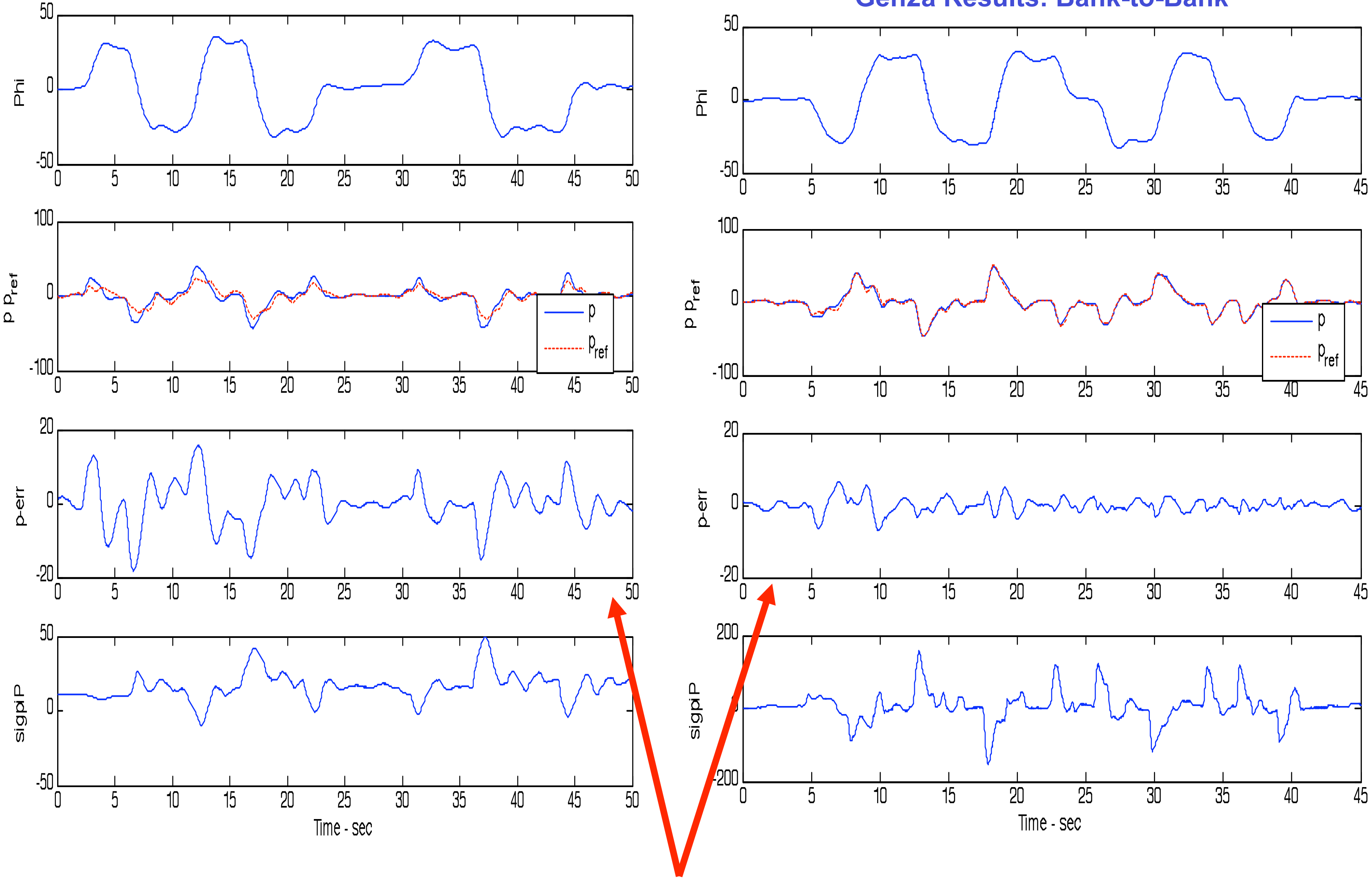
Handling Qualities ratings improvement	
Gen2	Gen2a
Gross Acq HQR: 4	Gross Acq HQR: 3
Fine Tracking HQR: 3	Fine Tracking HQR: 2

- •Flight Condition: Flt 230, Mach 0.75 / H = 20,000 ft.
- •Task: Bank-to-Bank Captures
- •Failure: Left Stabilator Jammed at +3.5 deg (trim)

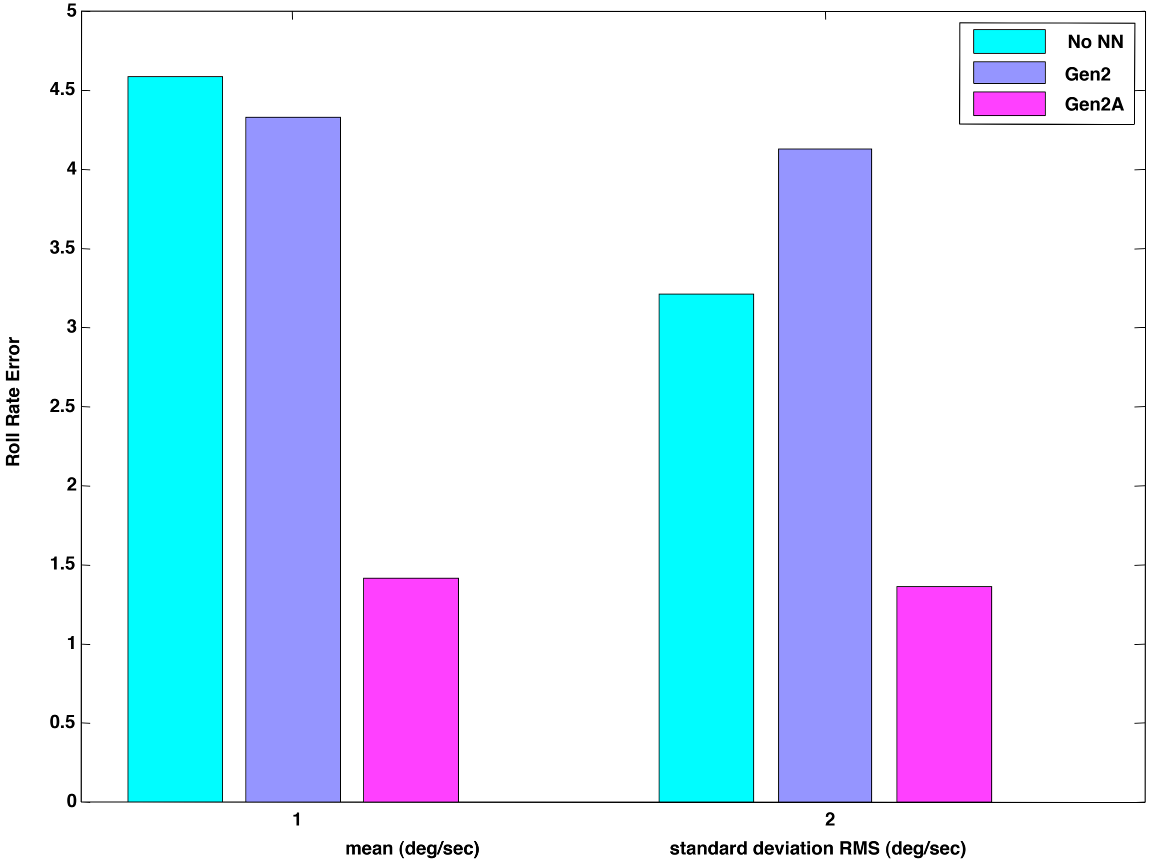
Gen2 & Gen2a Sigma Pi Flight Results

Gen2 Results: Bank-to-Bank

Gen2a Results: Bank-to-Bank



Note: Reduced Tracking errors for similar Pilot Inputs



Roll rate error mean and root mean square deviation standard (RMS) for:

•no neural networks,

•Gen2

•Gen2a

Conclusions

- Gen2 controller had deficiencies with [B] failures
- Gen2a controller out performed Gen2
- Handling Qualities ratings improvement
  - CHR went down (improved) one level with Gen2a

Questions?