Modeling-Error-Driven Performance-Seeking Direct Adaptive Control

Nilesh V. Kulkarni[1]
Perot Systems, Inc., NASA Ames Research Center, Moffett Field, CA – 94035

John Kaneshige[2] and Kalmanje Krishnakumar[3]
NASA Ames Research Center, Moffett Field, CA – 94035

John Burken[4]
NASA Dryden Flight Research Center, Edwards, CA - 93523

This paper presents a stable discrete-time adaptive law that targets modeling errors in a direct adaptive control framework. The update law was developed in our previous work for the adaptive disturbance rejection application. The approach is based on the philosophy that without modeling errors, the original control design has been tuned to achieve the desired performance. The adaptive control should, therefore, work towards getting this performance even in the face of modeling uncertainties/errors. In this work, the baseline controller uses dynamic inversion with proportional-integral augmentation. Dynamic inversion is carried out using the assumed system model. On-line adaptation of this control law is achieved by providing a parameterized augmentation signal to the dynamic inversion block. The parameters of this augmentation signal are updated to achieve the nominal desired error dynamics. Contrary to the typical Lyapunov-based adaptive approaches that guarantee only stability, the current approach investigates conditions for stability as well as performance. A high-fidelity F-15 model is used to illustrate the overall approach.

I.Introduction

Stable direct adaptive flight control using neural networks received significant attention after the work of Rysdyk and Calise[1]. The Rysdyk and Calise baseline control design consists of dynamic inversion with command augmentation. The augmented command signal is given by an adaptive neural network, whose weights are updated online. The update law guarantees that a Lyapunov function, which includes the system error and the neural network weight error, is non-increasing with time. Prior to this work, various researchers, including Narendra[2-4], have used adaptive augmentation of the command signal in a dynamic-inversion or feedback-linearization-based control design. The subsequent work has looked at different aspects of the problem including stability with different kinds of networks, actuator saturation, and output feedback. These developments have also been implemented in various flight applications.

Complementing this effort, researchers at NASA Ames Research Center have been looking at different adaptive control approaches within the Intelligent Flight Control program[5-8]. Several approaches have been considered in the various guidance and control loops. These include the work of Rysdyk and Calise for inner-loop control and adaptive critics[8] for outer-loop guidance. While guaranteeing stability, achieving good performance has always been one of the major concerns in the inner-loop architecture. The key issue in all the adaptive control literature is the definition of the Lyapunov function. Once the Lyapunov function is prescribed, the update law guarantees negative semi-definite time derivative of this Lyapunov function. Typically the Lyapunov function includes the tracking error and the parameter error terms. In this work, we define the Lyapunov function that captures the modeling error between the assumed plant and the actual plant. Guarantees of a monotonical decrease of this Lyapunov function

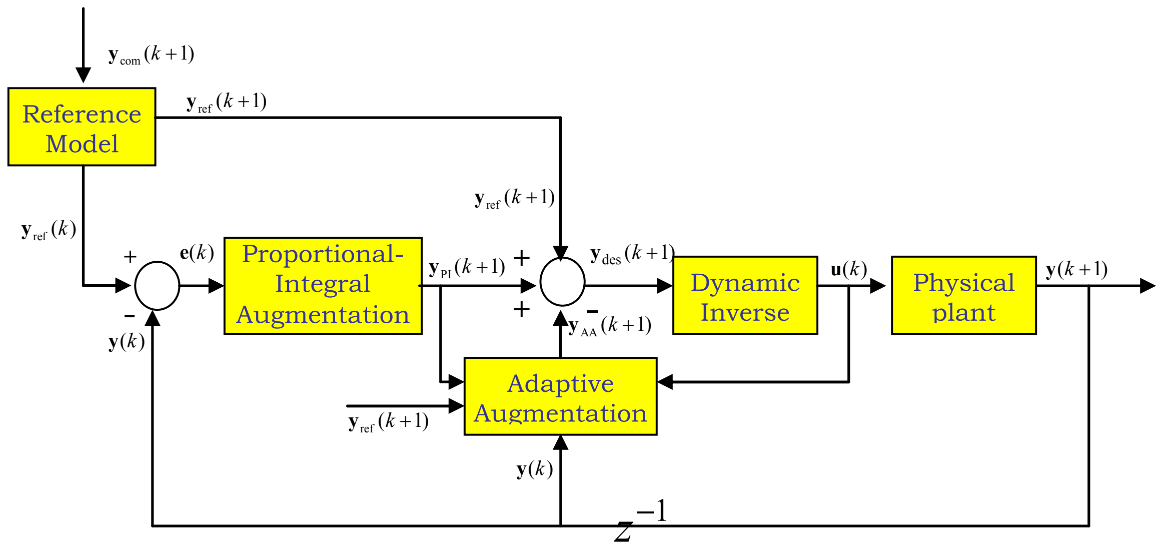
[1] Scientist, AIAA Member, Nilesh.V.Kulkarni@nasa.gov
[2] Flight Control Engineer, AIAA Member, John.T.Kaneshige@nasa.gov
[3] Scientist, Group Lead (Adaptive Control Technologies Group), AIAA Associate Fellow, kkumar@mail.arc.nasa.gov
[4] Flight Control Engineer, AIAA Member, John.Burken@dfrc.nasa.gov

then reduce the modeling error, and thereby provide performance that would have been achieved from the original control system with exact knowledge of the plant. The critical point is that this is achieved in a direct adaptive framework rather than the typical indirect adaptive control framework. In our previous work, we investigated this approach for the problem of adaptive disturbance rejection[9].

With this introduction, the remaining portion of the paper is organized as follows. Section II outlines the overall control architecture, which includes the baseline control design and motivates the adaptive control design with a SISO example. Section III looks at a general state space problem, parameterization of the adaptive controller, and presents a normalized gradient update law. Section IV presents the results of this adaptive control implementation for the rate-command attitude-hold (RCAH) of an F-15 model. Finally, section V concludes the paper, and outlines directions for future work.

II.Control Architecture

Figure 1 presents the overall adaptive control architecture. We first outline the baseline controller (without adaptation) to motivate the underlying control principle. This is followed by discussion and equations with the adaptive element.



$\mathbf{y}_{\mathrm{com}}(k+1)$
Reference Model
$\mathbf{y}_{\mathrm{ref}}(k+1)$
$\mathbf{y}_{\mathrm{ref}}(k+1)$
$\mathbf{y}_{\mathrm{ref}}(k)$
+
$\mathbf{e}(k)$
Proportional-Integral Augmentation
$\mathbf{y}_{\mathrm{PI}}(k+1)$
+
$\mathbf{y}_{\mathrm{des}}(k+1)$
Dynamic Inverse
$\mathbf{u}(k)$
Physical plant
$\mathbf{y}(k+1)$
-
$\mathbf{y}(k)$
+
-
$\mathbf{y}_{\mathrm{AA}}(k+1)$
Adaptive Augmentation
$\mathbf{y}_{\mathrm{ref}}(k+1)$
$\mathbf{y}(k)$
z^{-1}

Figure 1: Adaptive Control Architecture

The control system is given a command $y_{\mathrm{com}}(k+1)$ (eg. Pitch rate command from the pilot's stick). The time index $(k+1)$ refers to the desired value at the next time index. Given the knowledge of how fast or slow the plant can handle such a command, it is typically taken through a second order reference model with appropriate damping and natural frequency to get the corresponding *achievable* reference command $y_{\mathrm{ref}}(k+1)$. The controller is design to achieve a prescribed second order error dynamics with respect to this reference command. Let this error dynamics, in the discrete form, be given as:

$$\mathbf{e}(k+1)+\mathbf{K}_{\mathrm{Pe}}\mathbf{e}(k)+\mathbf{K}_{\mathrm{Ie}}\mathbf{e}_{\mathrm{I}}(k)=0\,, \tag{1}$$

where $\mathbf{e}(k)=y(k)-y_{_{\mathrm{ref}}}(k)$, and $\mathbf{e}_{\mathrm{I}}(k)$ represents the integrated error until time index *k*. $\mathbf{K}_{\mathrm{Pe}}$ and $\mathbf{K}_{\mathrm{Ie}}$ are chosen appropriately to have the desired transient response characteristics. Equation (1), with the definition of the error, is used to compute the control input to achieve this desired error dynamics as follows. Equation (1) corresponds to:

$$\begin{gathered}\left[y(k+1)-y_{\text{ref}}(k+1)\right]+\mathbf{K}_{\text{Pe}}\left[y(k)-y_{\text{ref}}(k)\right]+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)=0\\ \Downarrow\\ y(k+1)=y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y_{\text{ref}}(k)-y(k)\right]-\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)\end{gathered}\tag{2}$$

The plant output $y(k+1)$ needs to satisfy Eq. (2) to achieve the prescribed second order error dynamics. The right hand side of Eq. (2) can thus be labeled as $y_{des}(k+1)$, the desired plant output. Let the plant dynamics be given as:

$$\begin{aligned}y(k+1)=&f\left[y(k),y(k-1),\ldots,y(k-p_y),u(k-1),u(k-2),\ldots,u(k-p_u)\right]\\&+g\left[y(k),y(k-1),\ldots,y(k-p_y),u(k-1),u(k-2),\ldots,u(k-p_u)\right]u(k)\end{aligned}\tag{3}$$

We can thus invert the dynamics represented by Eq. (3) to compute the control $u(k)$ to achieve the desired error dynamics [Eq.(1)]as:

$$u(k)=\frac{1}{g\left[y(k),\ldots,y(k-p_y),u(k-1),\ldots,u(k-p_u)\right]}\left\{\begin{array}{l}y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y_{\text{ref}}(k)-y(k)\right]-\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)\\-f\left[y(k),y(k-1),\ldots,y(k-p_y),u(k-1),u(k-2),\ldots,u(k-p_u)\right]\end{array}\right\}\tag{4}$$

This control input, with exact knowledge of the plant (*f* and *g*), will help achieve the desired second order error dynamics. With modeling uncertainties/errors, we will not know *f* and *g* exactly, but only their estimates given by the model, $\hat{f}$ and $\hat{g}$. The adaptive augmentation is now designed to offset these modeling errors, so that we can get the same error dynamics or the desired performance. With the adaptive augmentation, as shown in Figure 1, the desired output, $y_{des}(k+1)$, is given as:

$$y_{\text{des}}(k+1)=y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y_{\text{ref}}(k)-y(k)\right]-\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-y_{\text{ad}}(k+1)\tag{5}$$

The control input is given as:

$$u(k)=\frac{1}{\hat{g}\left[y(k),\ldots,y(k-p_y),u(k-1),\ldots,u(k-p_u)\right]}\left\{\begin{array}{l}y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y_{\text{ref}}(k)-y(k)\right]-\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-y_{\text{ad}}(k+1)\\-\hat{f}\left[y(k),y(k-1),\ldots,y(k-p_y),u(k-1),u(k-2),\ldots,u(k-p_u)\right]\end{array}\right\}\tag{6}$$

To analyze the effect of this control input, we look at the modeling error, which is defined as the difference between the actual plant output and that predicted by the model:

$$\begin{aligned}\varepsilon(k+1)=\mathbf{y}(k+1)&-\hat{f}\left[y(k),y(k-1),\ldots,y(k-p_y),u(k-1),u(k-2),\ldots,u(k-p_u)\right]\\&-\hat{g}\left[y(k),\ldots,y(k-p_y),u(k-1),\ldots,u(k-p_u)\right]u(k)\end{aligned}\tag{7}$$

Substituting the expression for the control input, given by Eq. (6), in Eq. (7) gives:

$$\begin{aligned}\varepsilon(k+1)&=y(k+1)-\left\{y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y_{\text{ref}}(k)-y(k)\right]-\mathbf{K}_{\text{Ie}}e_{\text{I}}(k)-y_{\text{ad}}(k+1)\right\}\\ \varepsilon(k+1)-y_{\text{ad}}(k+1)&=y(k+1)-y_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\left[y(k)-y_{\text{ref}}(k)\right]+\mathbf{K}_{\text{Ie}}e_{\text{I}}(k)\end{aligned}\tag{8}$$

In terms of the definition of the tracking error, Eq. (8) can written as:

$$e(k+1)+\mathbf{K}_{\text{Pe}}e(k)+\mathbf{K}_{\text{Ie}}e_{\text{I}}(k)=\varepsilon(k+1)-y_{\text{ad}}(k+1) \tag{9}$$

Equation (9) represents the critical equation of this approach. The left hand side of the equation is the desired second order error dynamics. The right hand side of the equation is the difference between the modeling error and adaptive augmentation signal input. The equation, thus, says that if the adaptive augmentation signal can learn the modeling error and cancel it, the error dynamics of this control loop will be restored to its desired nature. In other words, we will recapture the performance desired from this control loop. We, therefore, define the left hand side of Eq. (9) as the performance error, *E*.

$$E(k+1)=e(k+1)+\mathbf{K}_{\text{Pe}}e(k)+\mathbf{K}_{\text{Ie}}e_{\text{I}}(k) \tag{10}$$

We can now form a Lyapunov function that is a function of the performance error as:

$$L(k)=\gamma E(k)^{2} \tag{11}$$

An update law now can now be devised for the adaptive augmentation input, y_{ad}, that forces a decreasing/non-increasing nature to the this Lyapunov function.

III.Parameterization and Update Laws for the Adaptive Augmentation

Section II provided a SISO example for motivating the proposed adaptive control approach. In this section, we investigate two main questions. The first relates to the parameterization of the modeling error, and the second relates to the choices for designing stable update laws.

1. Parameterization for a linear system:

Consider a linear system of the form:

$$\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}\mathbf{u}(k) \tag{12}$$

In a manner similar to that illustrated by Eqs. (4-6), the control input is computed as:

$$\mathbf{u}(k)=\hat{\mathbf{B}}^{-1}\left[\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-\mathbf{x}_{\text{ad}}(k+1)-\hat{\mathbf{A}}\mathbf{x}(k)\right], \tag{13}$$

where $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are estimates of the system **A** and **B** matrices. If the system matrices (**A**, **B**) are known, there is no need for the adaptive augmentation, and the control input is computed as:

$$\mathbf{u}(k)=\mathbf{B}^{-1}\left[\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-\mathbf{A}\mathbf{x}(k)\right] \tag{14}$$

If these control inputs need to provide the same desired error dynamics, they need to be equated, which gives the form of the idealized value of the augmentation signal.

$$\hat{\mathbf{B}}^{-1}\left[\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-\mathbf{x}^{*}_{\text{ad}}(k+1)-\hat{\mathbf{A}}\mathbf{x}(k)\right]=\mathbf{B}^{-1}\left[\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)-\mathbf{A}\mathbf{x}(k)\right] \tag{15}$$

$$\begin{aligned}\mathbf{x}^{*}_{\text{ad}}(k+1)&=\left(\mathbf{I}-\hat{\mathbf{B}}\mathbf{B}^{-1}\right)\left[\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)\right]+\left(\hat{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}-\hat{\mathbf{A}}\right)\mathbf{x}(k)\\&=\left[\left(\mathbf{I}-\hat{\mathbf{B}}\mathbf{B}^{-1}\right)\quad\left(\hat{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}-\hat{\mathbf{A}}\right)\right]\begin{bmatrix}\mathbf{x}_{\text{ref}}(k+1)+\mathbf{K}_{\text{Pe}}\mathbf{e}(k)+\mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k)\\ \mathbf{x}(k)\end{bmatrix}\end{aligned} \tag{16}$$

2. Parameterization for a non-linear system affine in control:

Consider a non-linear system that is affine in control, and whose dynamics can be written as linear in parameters.

$$\mathbf{x}(k+1) = \mathbf{W}_f \boldsymbol{\beta}_f(k) + \mathbf{B}\mathbf{u}(k)\,, \tag{17}$$

where $\mathbf{W}_f$ is the linear dynamic weight matrix, and $\boldsymbol{\beta}_f$ correspond to the nonlinear functions of the system state. The control input is computed in a similar manner as:

$$\mathbf{u}(k) = \hat{\mathbf{B}}^{-1}\left[\mathbf{x}_{\text{ref}}\left(k+1\right) + \mathbf{K}_{\text{Pe}}\mathbf{e}(k) + \mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k) - \mathbf{x}_{\text{ad}}\left(k+1\right) - \hat{\mathbf{W}}_f \boldsymbol{\beta}_f(k)\right], \tag{18}$$

where $\hat{\mathbf{W}}_f$ and $\hat{\mathbf{B}}$ are the corresponding estimates of the system matrices. By carrying out the analysis similar to the linear system case, the ideal augmentation signal can be computed to be:

$$\mathbf{x}^*_{\text{ad}}\left(k+1\right) = \left[\left(\mathbf{I} - \hat{\mathbf{B}}\mathbf{B}^{-1}\right) \quad \left(\hat{\mathbf{B}}\mathbf{B}^{-1}\mathbf{W}_f - \hat{\mathbf{W}}_f\right)\right]\begin{bmatrix}\mathbf{x}_{\text{ref}}\left(k+1\right) + \mathbf{K}_{\text{Pe}}\mathbf{e}(k) + \mathbf{K}_{\text{Ie}}\mathbf{e}_{\text{I}}(k) \\ \boldsymbol{\beta}_f(k)\end{bmatrix}, \tag{19}$$

Equations (16) and (19) imply that the ideal augmentation signal can be written as:

$$\mathbf{x}^*_{\text{ad}}\left(k+1\right) = \mathbf{W}^{*\,T}_{\text{ad}}\boldsymbol{\beta}(k)\,, \tag{20}$$

with the ideal weights, $\mathbf{W}^*_{\text{ad}}$, and the basis functions, $\boldsymbol{\beta}$, as given in Eqs. (16) and (19). It is interesting to note that these are the same basis functions used in Rysdyk and Calise[1]. Thus, we can parameterize a neural network in this form, and compute the ideal weights iteratively using an appropriate update algorithm.

A.Update Laws for the Adaptive Augmentation:

Having looked at the question of parameterization, we now consider the question of a stable update law for the parameters, $\mathbf{W}_{\text{ad}}$. Parameterizing, the adaptive augmentation signal in the form given by Eq. (20), and using the definition of the performance error as given in Eq. (10), Eq. (9) can be rewritten as:

$$\mathbf{E}\left(k\right) = \boldsymbol{\varepsilon} - \mathbf{y}_{\text{ad}}(k) \tag{21}$$

Compared to Eq. (9), this is written for a vector, **E**, corresponding to the general case of multiple control loops. Written in this form, the equation says that we are trying to estimate the vector modeling error, $\boldsymbol{\varepsilon}$ (for all loops) with the adaptive augmentation signal, $\mathbf{y}_{\text{ad}}(k)$. $\mathbf{E}\left(k\right)$ is the corresponding error in the estimate. This error dynamics for the performance error, **E**, corresponds to a system identification like problem. This opens up a host of approaches for doing this online system identification. In this work, we consider the normalized gradient update approach.

3. Normalized Gradient Update[11]:

Let $E_i(k+1)$ correspond to the ith element of the vector performance error $\mathbf{E}(k+1)$. Let $\mathbf{W}^*_{\text{ad}_i}$ represent the ith column vector of the weight matrix $\mathbf{W}^*_{\text{ad}}$, which corresponds to the ideal weights that minimize the performance error $\mathbf{E}$ to Δ^*. Similarly, let $\mathbf{W}_{\text{ad}_\text{i}}$ represent the ith column vector of the current estimate of the ideal weight matrix. The update law for each of these column vectors of the weight matrix is given as:

$$\mathbf{W}_{\mathrm{ad}_i}(k+1) = \mathbf{W}_{\mathrm{ad}_i}(k) + \frac{a(k) * E_i(k) * \boldsymbol{\beta}(k)}{\left[1 + \boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} \tag{22}$$

$a(k)$ corresponds to the learning rate that needs to satisfy the condition:

$$0 < a(k) \leq 2 \tag{23}$$

If the system experiences sufficient persistent excitation, reference [11] proves that this weight update law [Eqs.(22-23)] guarantees $E_i(k) \rightarrow \delta^*$ and $\mathbf{W}_{\mathrm{ad}_i}(k) \rightarrow \mathbf{W}^*_{\mathrm{ad}_i}$ $\forall i = 1 \ldots l$ as $k \rightarrow \infty$. Appendix A provides the stability proof for this update law.

B.What happened to tracking error?

The final part of this analysis corresponds to investigating the behavior of the system error $\mathbf{e}(k)$. This work is motivated towards providing an update only when we see modeling error, and not just if there is tracking error. However, having noting that, tracking error is what is ultimately of importance. It is therefore important to analyze the asymptotic behavior of the tracking error given the behavior of the performance error. For simplicity, in this analysis we consider the case where the desired error dynamics is first order given as:

$$E_i(k) = e_i(k+1) - K_{pe_i} e_i(k) = 0 \tag{30}$$

Let $|E_i(k)| < \delta$ after time *k*, where δ is some small positive scalar. This implies

$$\left| e_i(k+1) - K_{pe_i} e_i(k) \right| < \boldsymbol{\delta} \tag{31}$$

From Cauchy-Schwarz inequality,

$$\left| e_i(k+1) - K_{pe_i} e_i(k) \right| \geq \left| e_i(k+1) \right| - \left| K_{pe_i} \right| \left| e_i(k) \right| \tag{32}$$

Equations (31) and (32) imply:

$$\begin{aligned}
&\left| e_i(k+1) \right| - \left| K_{pe_i} \right| \left| e_i(k) \right| < \boldsymbol{\delta} \\
&\left| e_i(k+1) \right| < \left| K_{pe_i} \right| \left| e_i(k) \right| + \boldsymbol{\delta} \\
&\left| e_i(k+2) \right| < \left| K_{pe_i} \right| \left| e_i(k+1) \right| + \boldsymbol{\delta} \\
&\left| e_i(k+2) \right| < \left| K_{pe_i} \right| \left(\left| K_{pe_i} \right| \left| e_i(k) \right| + \boldsymbol{\delta} \right) + \boldsymbol{\delta} \\
&\qquad\qquad < \left| K_{pe_i} \right|^2 \left| e_i(k) \right| + \left| K_{pe_i} \right| \boldsymbol{\delta} + \boldsymbol{\delta} \\
&\left| e_i(k+3) \right| < \left| K_{pe_i} \right|^3 \left| e_i(k) \right| + \left| K_{pe_i} \right|^2 \boldsymbol{\delta} + \left| K_{pe_i} \right| \boldsymbol{\delta} + \boldsymbol{\delta} \\
&\vdots \\
&\left| e_i(k+n) \right| < \left| K_{pe_i} \right|^n \left| e_i(k) \right| + \boldsymbol{\delta} \left(1 + \left| K_{pe_i} \right| + \left| K_{pe_i} \right|^2 + \ldots + \left| K_{pe_i} \right|^{n-1} \right)
\end{aligned} \tag{33}$$

Since $\left| K_{pe_i} \right| < 1$ for stable error dynamics, as $k \rightarrow \infty$, $\left| e_i(k) \right|$ is bounded above as:

$$|e_i(k)| < \frac{\delta}{\left(1-\left|K_{pe_i}\right|\right)} \tag{34}$$

Thus, if the performance error is bounded, Eq. (34) establishes bounds on the tracking errors. A similar analysis can be carried out for second order error dynamics. The result summarizes that as long as the desired error dynamics (first or second order) is stable, the tracking error will be bounded above given bounds on the performance error.

IV.Application to Aircraft Control

The modeling error-driven performance-seeking adaptive control design was implemented for aircraft roll, pitch, and yaw rate control. The NASA Intelligent Flight Controller (IFC) design has been tested, and is currently undergoing various modifications for being flight-tested on the research F-15 aircraft. The IFC design has been implementing the adaptive control design as outlined by Calise and Rysdyk. For implementing the performance-seeking adaptive augmentation, the requirement was that it needed to fit within the existing architecture. The main issue in the implementation is that the baseline controller in the IFC architecture uses continuous-time aircraft dynamic inversion, whereas the proposed design has been outlined in the discrete-time. The equations outlined in sections II and III have been formulated for a discrete-time model inversion. We realized, however, that after reducing the problem to the core error dynamics, the problems became identical.

The error equation for the continuous-time implementation for a desired second-order error dynamics is given as:

$$\dot{e} + K_p e + K_I \int e dt = \varepsilon - U_{\text{ad}} \tag{35}$$

The error is defined in the same manner as the discrete case (eg. $q - q_{ref}$). The modeling error, ε , however corresponds to the difference in the acceleration as predicted by the model and that actually observed. Similarly, U_{ad} represents the augmentation acceleration command given by the adaptive block. If the left hand side of Eq. (35) is discretized while maintaining the continuous-time constants, the resulting discrete-time equation is given as:

$$\frac{1}{\Delta t}\left\{e(k) + (K_p \Delta t - 1)e(k-1) + K_I \Delta t e_I(k-1)\right\} = \left[\varepsilon(k) - U_{\text{ad}}(k)\right] \tag{36}$$

Defining the left hand side of Eq. (36) as the modified performance error, $\hat{E}(k)$, we get

$$\hat{E}(k) = \varepsilon(k) - U_{\text{ad}}(k) \tag{37}$$

This modified performance error equation is identical to the discrete-time version given by Eq. (21). The adaptive augmentation acceleration signal, $U_{\text{ad}}(k)$, can be parameterized in a similar manner, and the same update laws remain valid for the parameters of this augmentation signal for reducing $\hat{E}(k)$. A zero value of this modified performance error restores the second order error dynamics (LHS of Eq. (35)) to zero, and thereby regains the desired performance from the control loops.

Formulated in this manner, this adaptive approach fits within the existing IFC framework, and is considered as an alternate approach for flight testing. In the following discussion, we present the results of this implementation on the high fidelity model of the modified F-15 aircraft used at the Dryden Flight Research Center. In this study, we look at 2 cases. In the first case, the right stabilator is locked at 4 degrees at 10 seconds into the flight experiment. In the second case, the canard multiplier is set at -1, again at 10 seconds into the flight experiment. The behavior of the aircraft and update algorithm is examined for the the longitudinal and lateral pilot stick inputs given by Figure 2.

Figures 3-7 present the behavior of the aircraft and the neural net signal for the right stabilator failure case. In figure 3, we can observe the learning in the pitch channel, when the aircraft pitch rate starts following the desired pitch rate. Figures 4 and 5 examine the learning in the roll and yaw channels respectively. The learning in the yaw channel is not as good as in the pitch and roll channels. Figure 6 examines the behavior of the performance errors in each of the three axes as a function of learning. These performance errors drive the updates in each of the three axes.

We can note that these get smaller with time. Figure 7 presents the aircraft surface commands for this maneuver.

In a similar manner, figures 8-11 present the results for the incorrect canard multiplier case. Again we can note the good learning achieved in the pitch and roll channels. The insufficient learning in the yaw channel is being examined based on its baseline control system.

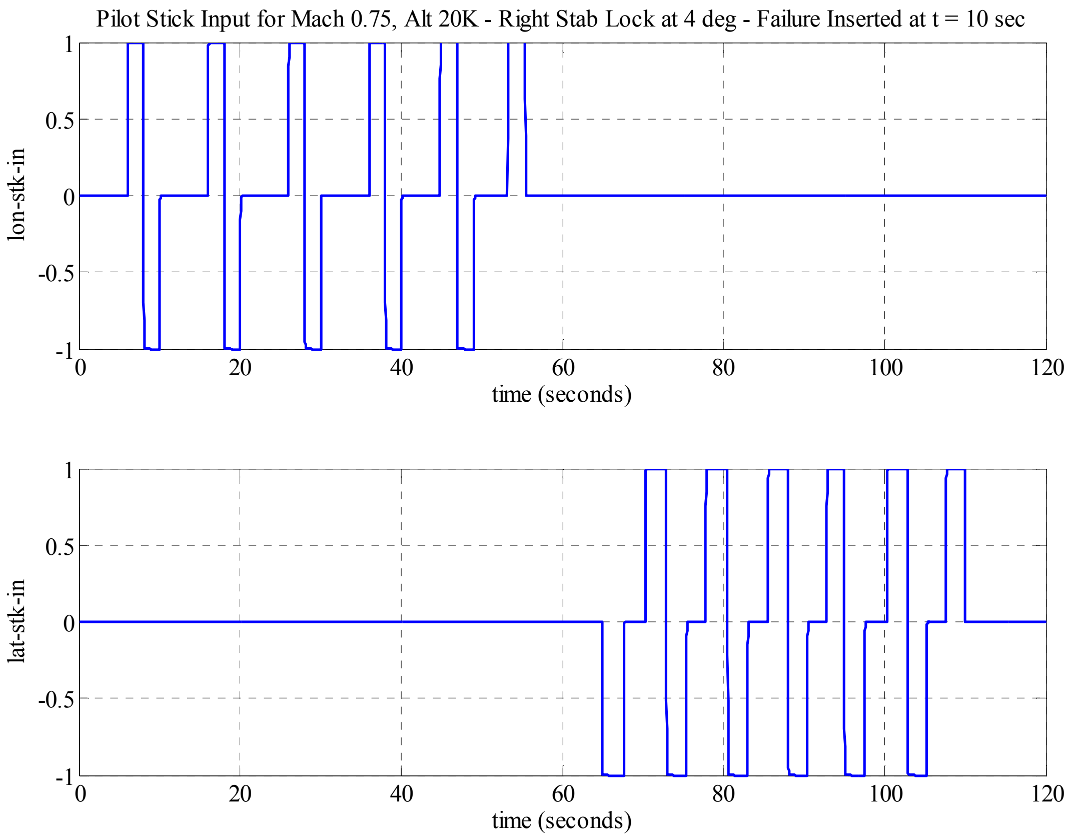


Figure 2: Pilot Longitudinal and Lateral Stick Inputs

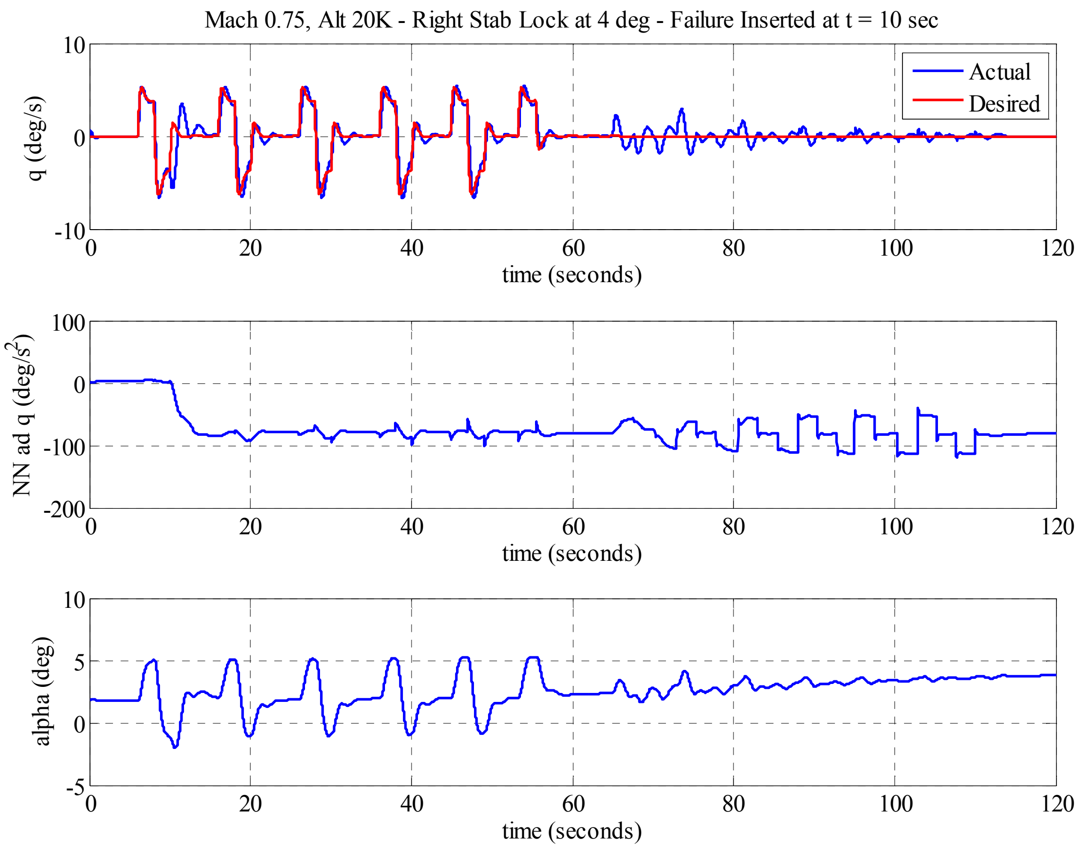


Figure 3: Aircraft pitch behavior along with the adaptive augmentation pitch acceleration signal for the right stab locked at 4 degrees at 10 seconds

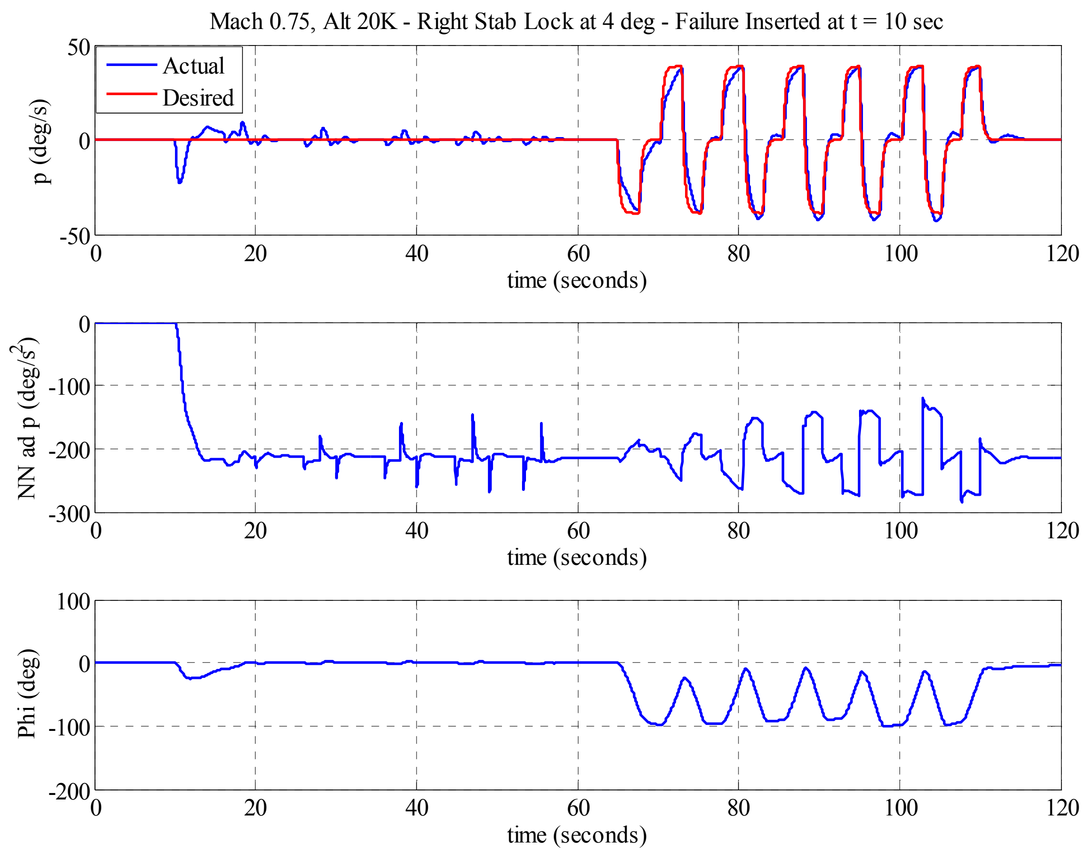


Figure 4: Aircraft roll behavior along with the adaptive augmentation roll acceleration signal for the right stab locked at 4 degrees at 10 seconds

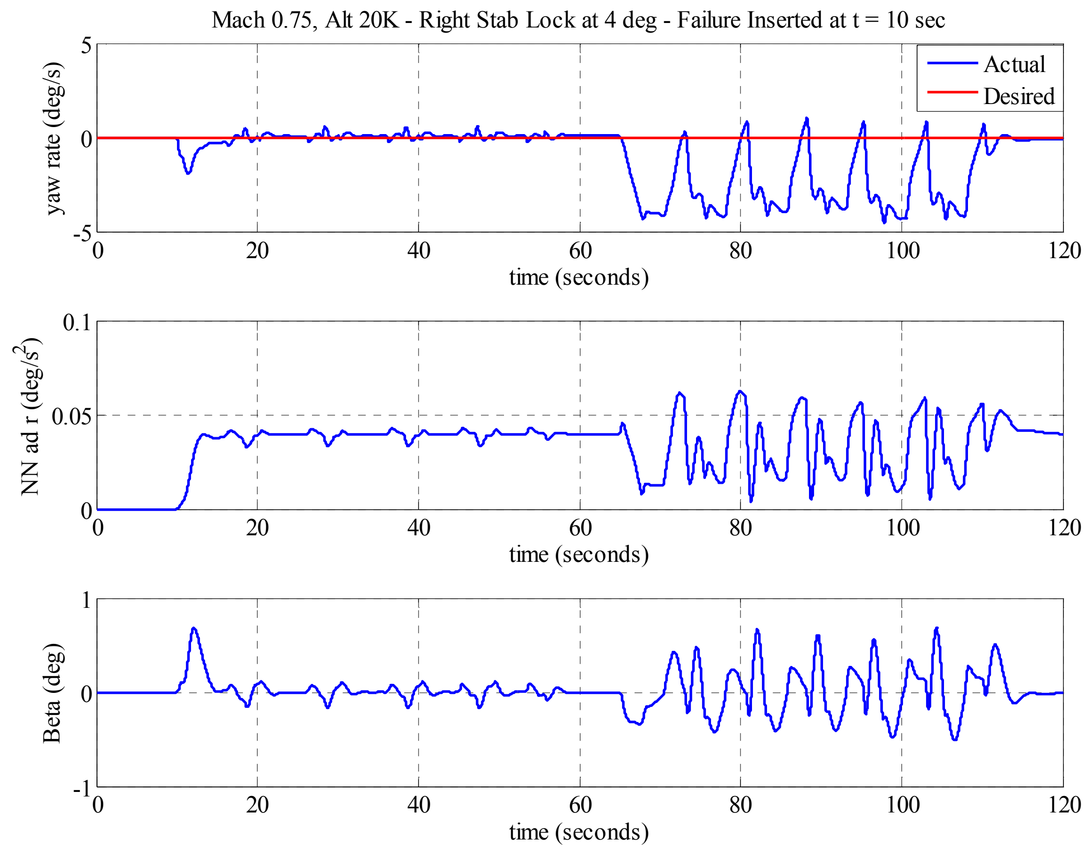


Figure 5: Aircraft yaw behavior along with the adaptive augmentation roll acceleration signal for the right stab locked at 4 degrees at 10 seconds

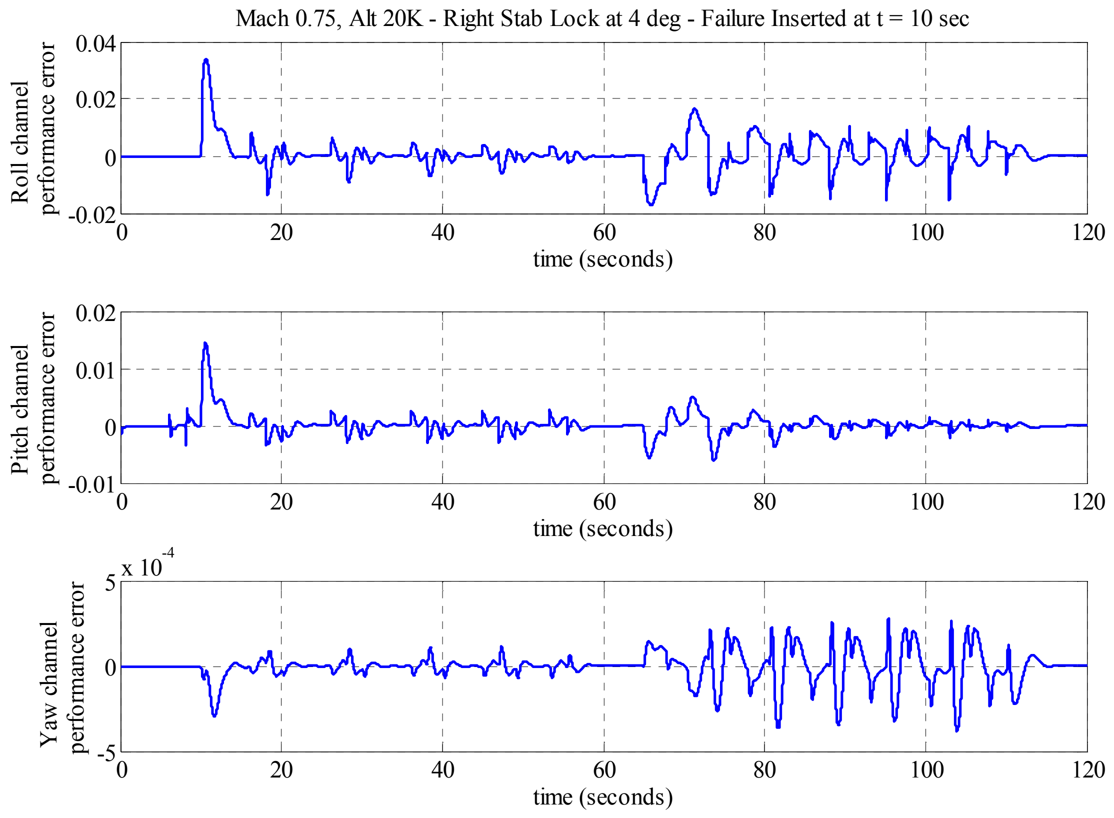


Figure 6: Behavior of the performance errors (that drive the adaptation) along the three axes for the right stab locked at 4 degrees at 10 seconds

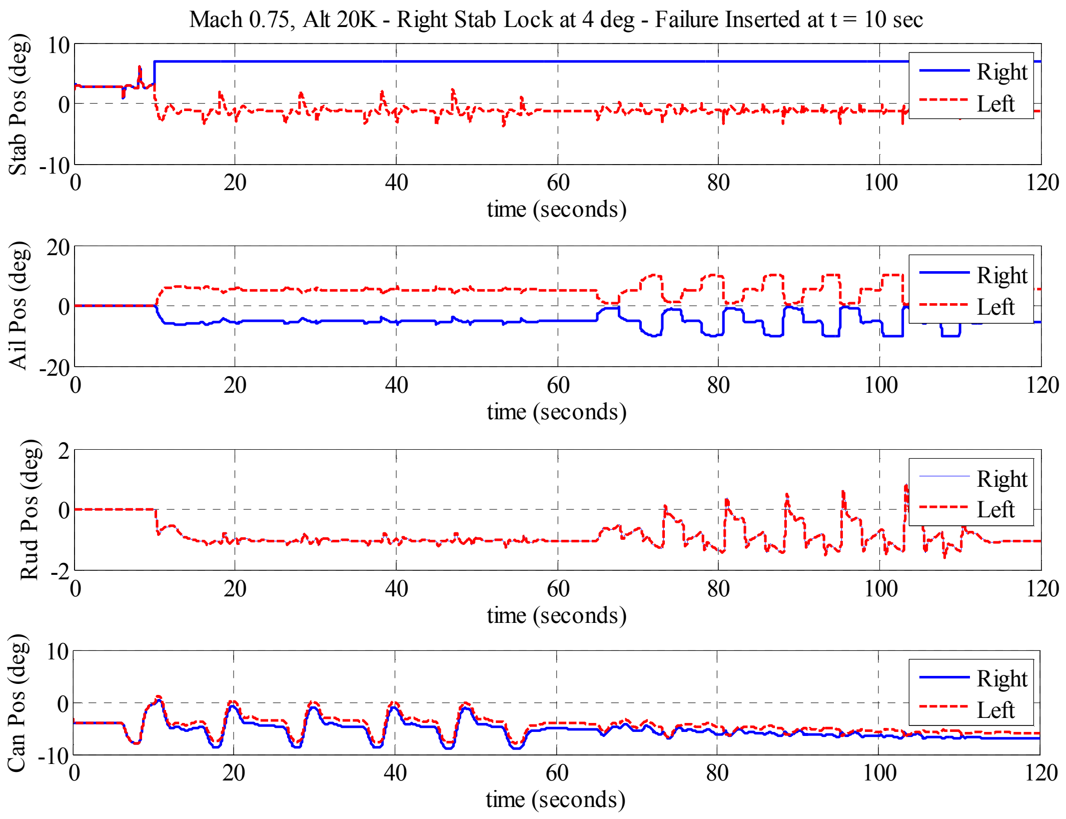


Figure 7: Aircraft surface commands axes for the right stab locked at 4 degrees at 10 seconds

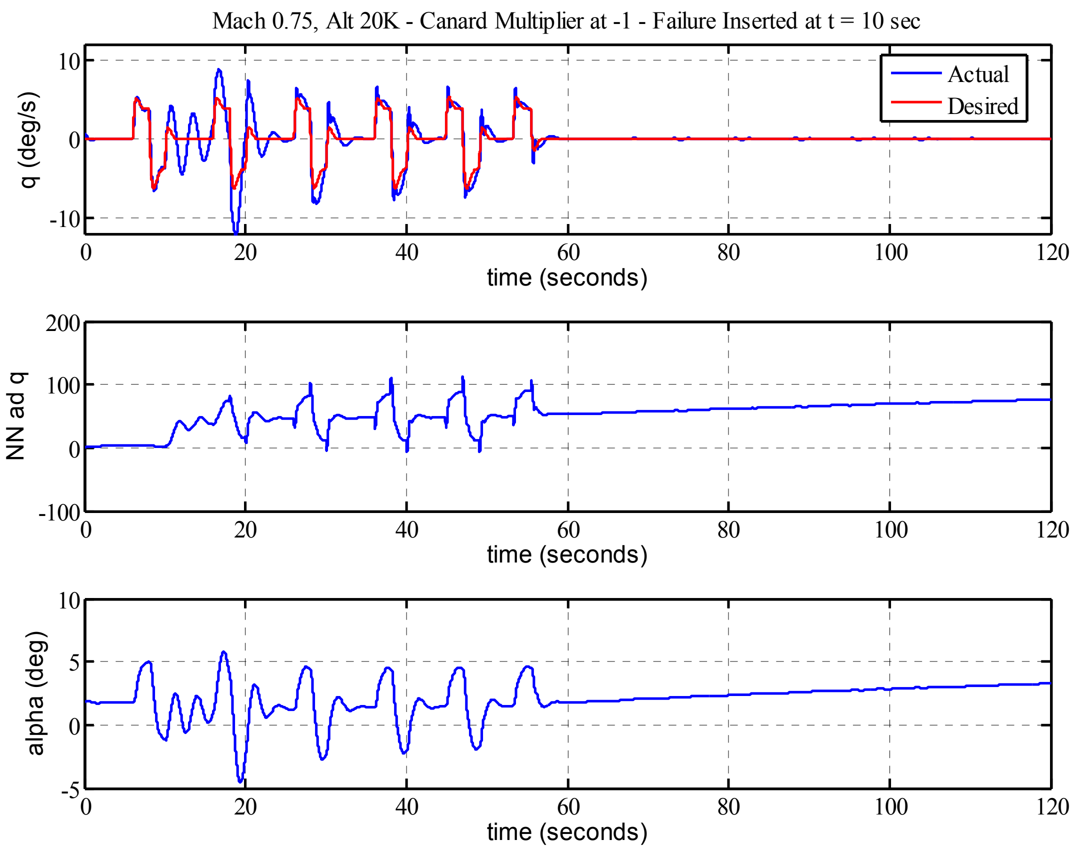


Figure 8: Aircraft pitch behavior along with the adaptive augmentation pitch acceleration signal for the canard multiplier set at -1 at 10 seconds

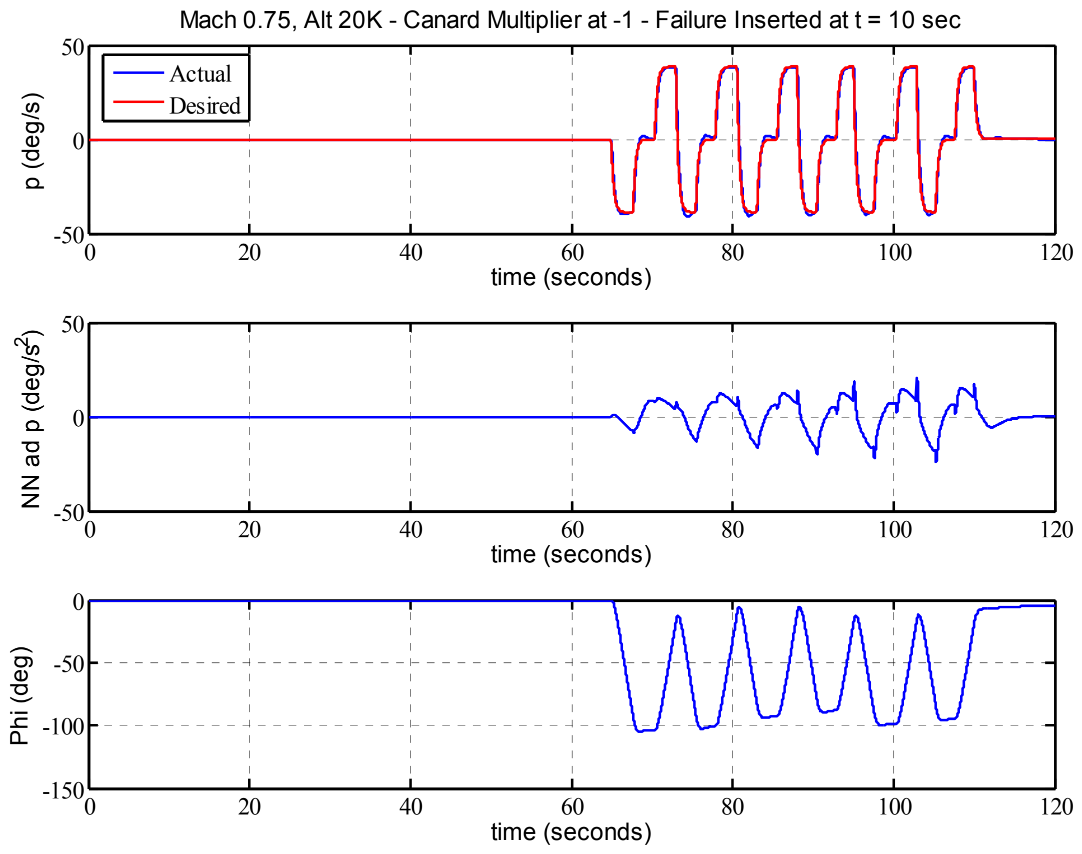


Figure 9: Aircraft roll behavior along with the adaptive augmentation roll acceleration signal for the canard multiplier set at -1 at 10 seconds

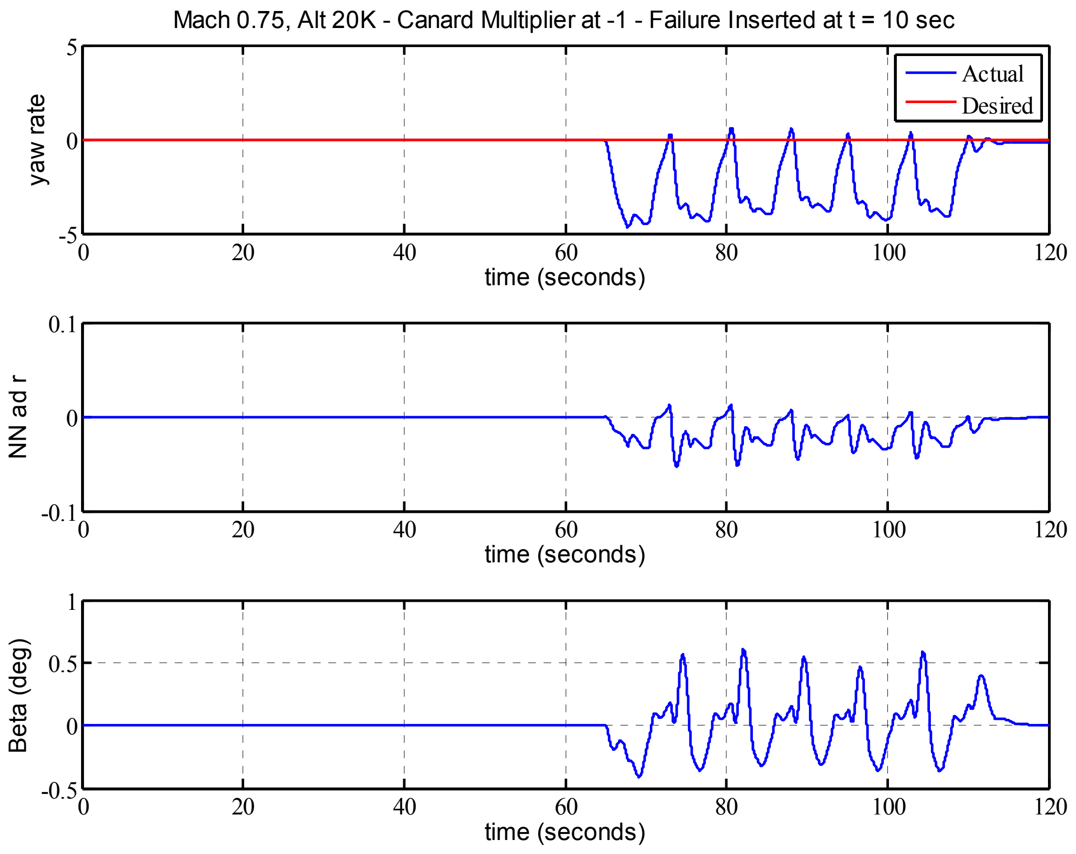


Figure 10: Aircraft yaw behavior along with the adaptive augmentation roll acceleration signal for the canard multiplier set at -1 at 10 seconds

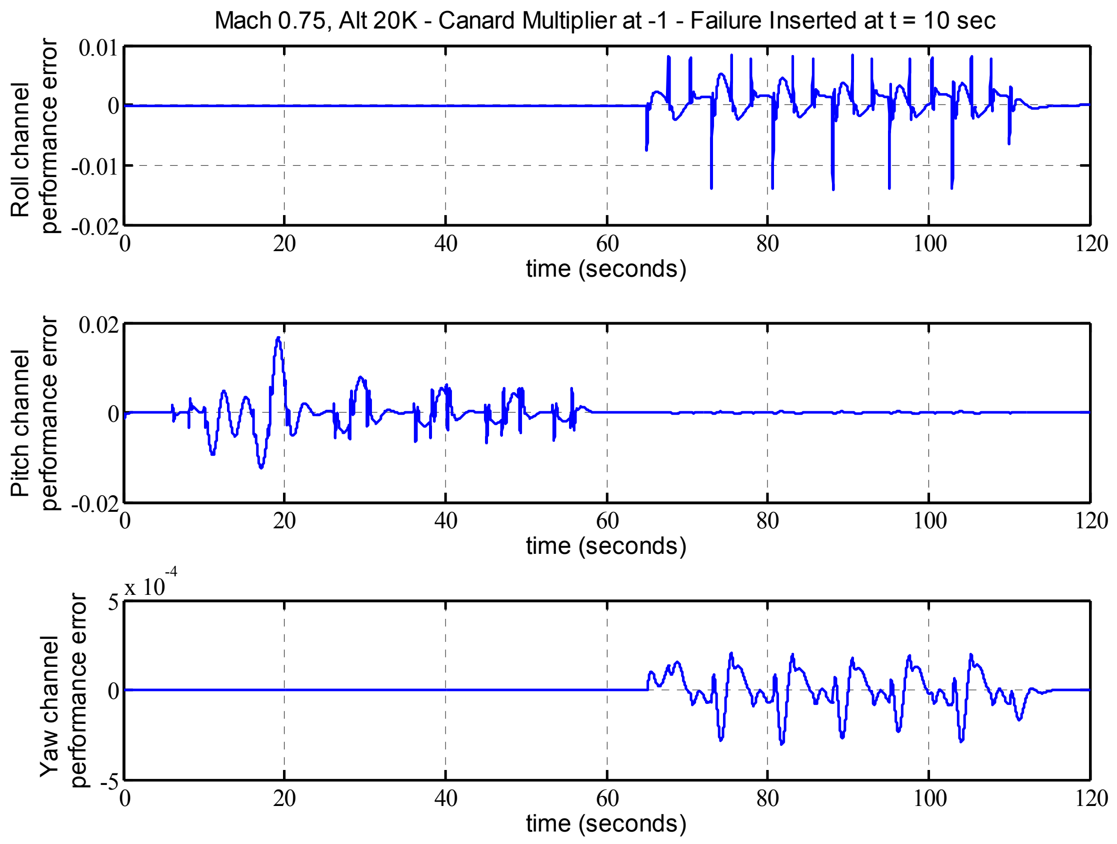


Figure 11: Behavior of the performance errors (that drive the adaptation) along the three axes for the canard multiplier set at -1 at 10 seconds

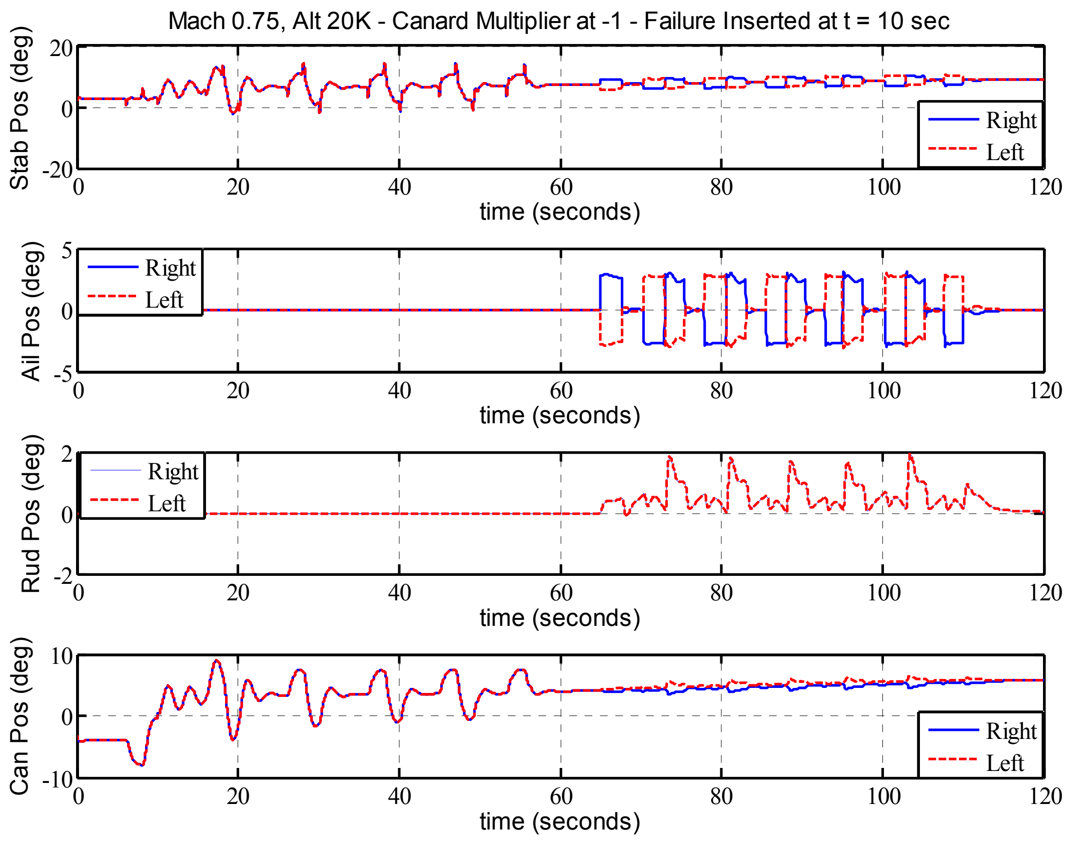


Figure 12: Aircraft surface commands axes for the canard multiplier set at -1 at 10 seconds

V.Conclusions

A stable discrete-time update law is investigated that provides augmentation to the existing controller only if the tracking error does not conform to its desired behavior. It is shown that not conforming to the desired first or second order behavior corresponds to the modeling error between the assumed and the actual plant. Thus the update law is modeling error driven, but in a direct adaptive framework. This update philosophy tries to regain the tuned performance that the control loop would have provided had there been no modeling error. This approach is being investigated for implementation as an alternate adaptive controller in the Intelligent Flight Control (IFC) architecture using the NASA Dryden high fidelity F-15 model. The results illustrate that we can achieve significant improvements in the aircraft behavior with the proposed approach. Future work will investigate a more comprehensive study of the approach including comparison of this approach for pilot handling qualities and frozen weight stability margin studies. We will also look at second order update methods for investigating the speed of convergence.

References

[1]Rysdyk, R. T., and Calise, A. J., “Fault Tolerant Flight Control via Adaptive Neural Network Augmentation,” AIAA 98-4483, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, MA, August 1998.
[2]Narendra, K. S., Lin, Y-H., and Valavani, L. S., “Stable Adaptive Controller Design, Part II: Proof of Stability,” IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, pp. 440-448, June 1980.
[3]Narendra, K. S., and Lin, Y-H.., “Stable Discrete Adaptive Control,” IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, pp. 456-461, June 1980.
[4]Lin, Y-H., and Narendra, K. S., “A New Error Model for Adaptive Systems,” IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, pp. 585-587, June 1980.
[5]Kaneshige, John, Bull, John, and Totah, Joseph J., “Generic Neural Flight Control and Autopilot System,” AIAA Guidance, Navigation, and Control Conference, AIAA-2000-4281, August 2000.
[6]Kaneshige, John, and Gundy-Burlet Karen, “Integrated neural flight and propulsion control system,” AIAA Guidance, Navigation, and Control Conference, AIAA-2001-4386, August 2001.
[7]Nguyen, N., Krishnakumar K., and Kaneshige J., “Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft,” AIAA Guidance, Navigation, and Control Conference, AIAA-2006-6049, August 2006.
[8]Krishnakumar, K., Limes, G., Gundy-Burlet, K., and Bryant, D., “An Adaptive Critic Approach to Reference Model Adaptation,” AIAA Guidance, Navigation, and Control Conference, AIAA-2003-5790, August 2003.
[9]Kulkarni, N., “Adaptive Disturbance Rejection Control Using System Input-Output Data,” AIAA Guidance, Navigation, and Control Conference, AIAA-2006-6414, August 2006.
[10]Ioannou, P. A., and Sun, J., *Robust Adaptive Control*, Englewood Cliffs, NJ, Prentice Hall, 1996.
[11]Goodman, G. C., Ramadge, P. J., and Caines, P. E., “Discrete-time Multivariable Adaptive Control,” *IEEE Transactions on Automatic Control*, Vol. 25, No. 3, June 1980, pp. 449-456.

VI.Appendix

Proof of stability for the update law given in Eq. (22) [11]:

The weight error is defined as:

$$\hat{\mathbf{W}}_{\mathrm{ad}_i} = \mathbf{W}_{\mathrm{ad}_i} - \mathbf{W}^*_{\mathrm{ad}_i} \tag{A1}$$

Substituting in Eq. (25) gives:

$$\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1) = \hat{\mathbf{W}}_{\mathrm{ad}_i}(k) + \frac{a(k) * E_i(k+1) * \boldsymbol{\beta}(k)}{\left[1 + \boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} \tag{A2}$$

The performance error is given by Eq. (24) as:

$$E_i(k+1) = \mathbf{W}^{*}_{\mathrm{ad}_i}{}^{T}\boldsymbol{\beta}(k) - \mathbf{W}_{\mathrm{ad}_i}{}^{T}(k)\boldsymbol{\beta}(k) \tag{A3}$$

The weight error update can thus be given as:

$$\begin{aligned}
\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1) &= \hat{\mathbf{W}}_{\mathrm{ad}_i}(k) + \frac{a(k) * \left[\mathbf{W}^{*}_{\mathrm{ad}_i}{}^{T}\boldsymbol{\beta}(k) - \mathbf{W}_{\mathrm{ad}_i}{}^{T}(k)\boldsymbol{\beta}(k)\right] * \boldsymbol{\beta}(k)}{\left[1 + \boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} \\
&= \hat{\mathbf{W}}_{\mathrm{ad}_i}(k) - \frac{a(k) * \boldsymbol{\beta}(k)^T\left[\mathbf{W}_{\mathrm{ad}_i}(k) - \mathbf{W}^*_{\mathrm{ad}_i}\right] * \boldsymbol{\beta}(k)}{\left[1 + \boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} \\
&= \hat{\mathbf{W}}_{\mathrm{ad}_i}(k) - \frac{a(k) * \boldsymbol{\beta}(k)^T\hat{\mathbf{W}}_{\mathrm{ad}_i}(k) * \boldsymbol{\beta}(k)}{\left[1 + \boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}
\end{aligned} \tag{A4}$$

The norm of the weight error is given as:

$$\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)\right\|^2 = \hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1) \tag{A5}$$

Substituting Eq. (35) gives:

$$\begin{aligned}
\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)\right\|^2 &= \left\{\hat{\mathbf{W}}_{\mathrm{ad}_i}(k) - \frac{a(k)\boldsymbol{\beta}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\}^T \left\{\hat{\mathbf{W}}_{\mathrm{ad}_i}(k) - \frac{a(k)\boldsymbol{\beta}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\} \\
&= \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k) \\
&\quad - \frac{2a(k)\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\boldsymbol{\beta}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} + \frac{a(k)^2 \boldsymbol{\beta}(k)^T \boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\left\{\frac{\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\boldsymbol{\beta}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\} \\
&= \left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\right\|^2 + a(k)\left\{-2 + \frac{a(k)\boldsymbol{\beta}(k)^T \boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\}\left\{\frac{\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\boldsymbol{\beta}(k)^T \hat{\mathbf{W}}_{\mathrm{ad}_i}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\}
\end{aligned} \tag{A6}$$

$$\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)\right\|^2 - \left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\right\|^2 = a(k)\left\{-2 + \frac{a(k)\boldsymbol{\beta}(k)^T \boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\}\frac{\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\right\|^2}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} \tag{A7}$$

For the condition given by Eq. (26),

$$\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)\right\|^2 - \left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\right\|^2 < 0 \tag{A8}$$

The weight vector error norm, being bounded and decreasing, converges to zero as $k \to \infty$. This implies that $\mathbf{W}_{\mathrm{ad}_i}(k) \to \mathbf{W}^*_{\mathrm{ad}_i}$. Equation (38) also gives:

$$\begin{aligned}
&\lim_{k\to\infty}\left(\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k+1)\right\|^2 - \left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)\right\|^2\right) = \lim_{k\to\infty} a(k)\left\{-2 + \frac{a(k)\boldsymbol{\beta}(k)^T \boldsymbol{\beta}(k)}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]}\right\}\frac{\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\right\|^2}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} = 0 \\
&\Rightarrow \lim_{k\to\infty} \frac{\left\|\hat{\mathbf{W}}_{\mathrm{ad}_i}(k)^T \boldsymbol{\beta}(k)\right\|^2}{\left[1+\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)\right]} = 0 \\
&\Rightarrow \lim_{k\to\infty} E_i(k)^2 = 0
\end{aligned} \tag{A9}$$

Thus $E_i(k) \to 0$ as $k \to \infty$.