

A Nonlinear Dynamic Inversion Predictor-Based Model Reference Adaptive Controller for a Generic Transport Model

Stefan F. Campbell and John T. Kaneshige

Abstract—Presented here is a *Predictor-Based Model Reference Adaptive Control* (PMRAC) architecture for a generic transport aircraft. At its core, this architecture features a three-axis, non-linear, dynamic-inversion controller. Command inputs for this baseline controller are provided by pilot roll-rate, pitch-rate, and sideslip commands. This paper will first thoroughly present the baseline controller followed by a description of the PMRAC adaptive augmentation to this control system. Results are presented via a full-scale, non-linear simulation of NASA’s Generic Transport Model (GTM).

I. INTRODUCTION

The Integrated Resilient Aircraft Control project (IRAC) is a part of the NASA Aviation Safety program. A key focus of this project is to investigate adaptive control systems as a risk-mitigating technology for off-nominal conditions. This paper, based upon the research presented in [5], proposes a PMRAC adaptive control architecture as just such a candidate technology.

To begin, section II presents the baseline control architecture and justifies the design selection. Section III then presents the PMRAC adaptive augmentation to this system. Here some of the fundamental stability proofs for the architecture are shown, but the interested reader is referred to specific publications for further detail. Section IV presents results from a full non-linear simulation of the control system, both with, and without the PMRAC adaptive augmentation. Finally, section V presents a brief summation of this paper’s contribution to the community.

The primary intentions of this paper are two fold: to present the PMRAC architecture for a complete nonlinear dynamic inversion baseline controller (a novel contribution to the community in as far as the authors are aware) and to demonstrate the PMRAC architecture on a non-linear flight simulation. Future work will be aimed at more thoroughly assessing the strengths and weaknesses of this control strategy.

II. BASELINE CONTROLLER: NONLINEAR DYNAMIC INVERSION

A. Control Selection Justification

The baseline controller for this work is a full non-linear dynamic inversion controller based in large part on [1], [2], and [3]. This control architecture is chosen for several key reasons. First, this control approach can be effectively implemented as evinced by the recent selection of a dynamic inversion controller for the F-35 aircraft. Second, and more importantly, a dynamic inversion controller offers a cost and time effective way to develop a control system; with appropriate modeling, a full-flight control system can be quickly and efficiently developed for research studies, as contrasted with a more time intensive traditional gain-scheduled controller. This ultimately facilitates the rapid evaluation and testing of multiple adaptive control systems over a large range of flight conditions.

B. Control Architecture

The general (and well known) rigid body dynamics for an aircraft are presented below:

$$\tau = I\dot{\omega} + \omega \times I\omega. \quad (1)$$

Here, ω is $\in \mathbb{R}^{3 \times 1}$ and represents a vector of the roll (p), pitch (q), and yaw (r) rates. By modeling the torques on the aircraft using traditional aerodynamic stability derivatives (including p , q , and r derivatives), (1) may be decomposed into the following convenient form:

$$\dot{\omega} = A(t)\omega + G(t)z + B(t)u. \quad (2)$$

For this decomposition, the matrices A ($\in \mathbb{R}^{3 \times 3}$), G ($\in \mathbb{R}^{3 \times 7}$), and B ($\in \mathbb{R}^{3 \times 3}$) are time-varying (from this point forward the time dependence of these variables will not be expressly shown). Moreover, the aircraft’s control allocation tables are incorporated into (1) such that the control vector u is $\in \mathbb{R}^{3 \times 1}$ and represents the three, non-dimensional lateral, longitudinal, and directional control signals (limited from -1 to 1). The vector z represents a non-linear combination of ω , specifically pq , qr , pr , $(p^2 - q^2)$, $(r^2 - p^2)$, and $(q^2 - r^2)$, as well as a bias term (which accounts for the dependence of τ on slowly changing variables, such as angle of attack and sideslip).

Manuscript received September 16, 2009.

S. F. Campbell is with SGT Inc., Greenbelt, MD 20770 USA (e-mail: stefan.f.campbell@nasa.gov).

J. T. Kaneshige is with NASA, Moffett Field, CA 94035 USA. (e-mail: john.t.kaneshige@nasa.gov).

A non-linear dynamic inversion control law is then:

$$u = \hat{B}(t)^{-1}(\dot{\omega}_d - \hat{A}(t)\omega - \hat{G}(t)z). \quad (3)$$

For clarity, over-hats have been used to denote that parameters in the dynamic inversion control law are estimated; look-up tables provide these matrices in real-time operation. The reader should be careful not to confuse the dynamic inversion parameters with real-time, adaptively estimated parameters that will be introduced in later sections. Finally, the subscript d denotes a *desired* value.

To generate the desired values in (3), each of the three directional axes is treated independently. For the longitudinal and lateral axes, the pilot's stick commands are interpreted as pitch and roll *rate* commands. These commands are then filtered to produce a tracking signal for a linear PI control law. The control signal is further augmented by a feedforward term from the filter states. The resulting control signals are the desired pitch and roll rates for the dynamic inversion in (3).

For the directional axis, the pilot's pedal inputs are not interpreted directly as yaw rate commands, but instead as sideslip commands. As a result, an additional dynamic inversion is employed to generate the necessary yaw rate command; this is similar to the work done in [4] in which a two stage *slow* and *fast* dynamic inversion architecture was proposed (in this work the fast dynamic inversion is similar to the dynamic inversion shown in (3)). More precisely, the pilot's sideslip command is filtered to produce a tracking command for a proportional controller. This proportional control signal is augmented with a feedforward term from the filter states to generate a *desired sideslip* command. From this, a dynamic inversion is performed to generate a yaw rate command for the *fast* dynamic inversion in (3). The *desired* yaw rate command is then generated identically to the lateral and longitudinal axes, i.e. the *desired* yaw rate is generated from a linear PI controller augmented with a feedforward term.

For the slow dynamic inversion, the following well-known relationship is used as a starting point for derivation:

$$\begin{aligned} \dot{\beta} = \frac{1}{mV} [& D \sin(\beta) + Y \cos(\beta) - X_r \cos(\alpha) \sin(\beta) + \\ & mg(\cos\alpha \sin\beta \sin\theta + \cos\beta \sin\phi \cos\theta - \\ & \sin\alpha \sin\beta \cos\phi \cos\theta + p \sin\alpha - r \cos\alpha)] \end{aligned} \quad (4)$$

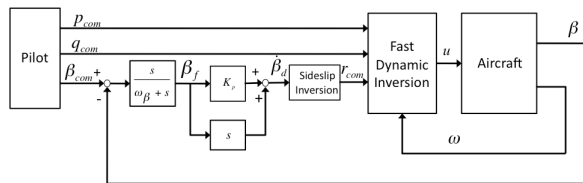


Fig. 1. High-level base-line control architecture. Here the dependency of the sideslip inversion on additional states (α , θ , ϕ , p , and V_t) is not illustrated.

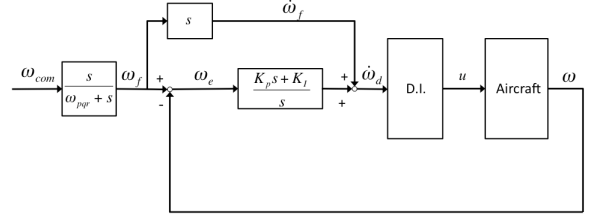


Fig. 2. Presented here is the inner-loop fast dynamics controller. The *D.I.* block contains the equation presented in (3).

The above expression is simplified to the more tractable relationship shown in (5).

$$\dot{\beta} \approx p \sin(\alpha) - r \cos(\alpha) + (g/V_t) \cos(\theta) \sin(\phi). \quad (5)$$

The yaw rate command is then determined from (5), as shown in (6).

$$r_{com} = -(\dot{\beta}_d - p \sin(\alpha) - (g/V_t) \cos(\theta) \sin(\phi)) / \cos(\alpha) \quad (6)$$

The complete architecture is presented in both Fig. 1 and Fig. 2. Specifically, Fig. 1 illustrates the outer loop control architecture while Fig. 2 shows the fast dynamic inversion using all three of the pitch, roll, and yaw rate commands. It is worth noting that the desired angular accelerations for this architecture (as diagramed in Fig. 2) are given below in (7).

$$\dot{\omega}_d = K_p(\omega_f - \omega) + K_I \int_0^t (\omega_f - \omega) d\tau + \dot{\omega}_f \quad (7)$$

Here, the output of the command filter is expressly represented with a subscript f . Moreover, it should be noted that both K_p and K_I are *constant* diagonal, 3x3 matrices.

As a final note on the general architecture, the pilot's commanded pitch rate is augmented by the level turn-compensation term $g \sin^2 \phi / V \cos \phi$. Qualitatively, this helps keep the nose of the aircraft up during bank and roll maneuvers.

III. PMRAC AUGMENTATION

The PMRAC architecture adjusts a typical MRAC architecture by adding an additional, predictive training error. The following section will develop the traditional MRAC reference error as well as the PMRAC prediction error. It should be noted that the PMRAC augmentation is only applied to the inner most loop of this controller, i.e. it is applied to the *fast* dynamic inversion illustrated in Fig. 2.

A. Preliminaries

To begin defining the *PMRAC* adaptive augmentation, we manipulate the general system model of (2) by incorporating the *true* matrix values A , B , and G with their controller estimates.

$$\dot{\omega} = \hat{A}\omega + \hat{G}z + \hat{B}u + \Delta A\omega + \Delta Gz + \Delta Bu \quad (8)$$

In (8), Δ denotes the difference between the true matrices and the estimated values used by the controller. To further develop the controller, the result in (8) is transformed into the more familiar adaptive controller A and B matrix form, as presented in (9).

$$\dot{\omega} = \hat{A}\omega + \hat{B}(u + f) \quad (9)$$

The term f is here introduced for convenience and comprises the model uncertainty g . It should be further noted that \hat{B} is assumed for a nominal aircraft and is thus taken as full-rank throughout the entirety of this paper. Comparing (8) and (9), the term f is then defined as

$$f = \hat{B}^{-1}g + \hat{B}^{-1}\hat{G}z, \quad (10)$$

where

$$g = (\Delta A\omega + \Delta Bu + \Delta Gz). \quad (11)$$

In much of the adaptive control literature an additional term is added to (10) to approximate non-parametric uncertainty. This term is significant from a stability proof and analysis perspective, but is ignored here for the purposes of developing the correct control laws and PMRAC predictor system.

To facilitate further development, the command filter in Fig. 2 is represented in state space form as in (12).

$$\dot{\omega}_f = A_m\omega_f + B_m r \quad (12)$$

Here the *reference input* r is provided from the pilot commanded roll and pitch rates and the outer-loop side-slip controller. The filter is assumed to be a first order filter, as illustrated in Fig. 2.

B. PMRAC Predictor System Development

For the PMRAC development presented here, the predictor system is defined using the following state vector:

$$x = \left[\left(\int_0^t \omega_f d\tau \right)^T \quad \omega_f^T \quad \left(\int_0^t \omega d\tau \right)^T \quad \omega^T \right]^T. \quad (13)$$

Combining (3), (7), (12), and (13) as well as introducing a currently undefined adaptive control term, the control law is rewritten as

$$u = K_x^T x + K_r^T r - K_z^T z - \hat{B}^{-1}u_{ad}, \quad (14)$$

where

$$K_x^T = [\hat{B}^{-1}K_I \quad \hat{B}^{-1}(K_p + A_m) \quad -\hat{B}^{-1}K_I \quad -\hat{B}^{-1}(K_p + \hat{A})], \quad (15)$$

$$K_r^T = \hat{B}^{-1}B_m, \quad (16)$$

and

$$K_z^T = \hat{B}^{-1}\hat{G}. \quad (17)$$

It should be observed that $x \in \mathfrak{R}^{12 \times 1}$, $K_x \in \mathfrak{R}^{12 \times 3}$, $K_r \in \mathfrak{R}^{3 \times 3}$, $K_z \in \mathfrak{R}^{7 \times 3}$, $u_{ad} \in \mathfrak{R}^{3 \times 1}$, and $u \in \mathfrak{R}^{3 \times 1}$.

The system in (9), given the above definitions, is now easily written as

$$\dot{x} = \bar{A}x + \bar{B}_m r + \bar{B}[K_x^T x + K_r^T r + \hat{B}^{-1}(g - u_{ad})] \quad (18)$$

In (18), the following definitions have been introduced:

$$\bar{A} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & A_m & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & \hat{A} \end{bmatrix}, \quad (19)$$

$$\bar{B}_m = [0 \quad B_m^T \quad 0 \quad 0]^T, \quad (20)$$

and

$$\bar{B} = [0 \quad 0 \quad 0 \quad \hat{B}^T]^T. \quad (21)$$

To complete the manipulation of the system dynamics, we can rewrite the system as

$$\dot{x} = A_r x + B_r r + B_l(g - u_{ad}), \quad (22)$$

where

$$A_r = (\bar{A} + \bar{B}K_x^T), \quad (23)$$

$$B_r = (\bar{B}_m + \bar{B}K_r^T), \quad (24)$$

and

$$B_l^T = [0 \quad 0 \quad 0 \quad I], \quad B_l^T \in \mathfrak{R}^{3 \times 12}. \quad (25)$$

It should be noted that the term g in (22) is time varying but that A_r and B_r are constant (time varying elements of these matrices end up cancelling to 0).

The reference system for the control architecture is then the following:

$$\dot{x}_r = A_r x_r + B_r r. \quad (26)$$

Inspection of this system shows that the reference system is merely the true states following the filtered states in (12) (assuming identical initial conditions for both). Additionally, the system in (26) is a linear time-invariant system (LTI).

For the PMRAC architecture, following the approach in [5], the predictor system is now the following:

$$\dot{\hat{x}} = A_p(\hat{x} - x) + A_r x + B_r r. \quad (27)$$

In (27), the matrix A_p is $\in \mathfrak{R}^{12 \times 12}$ and can be decomposed

into 4 sub-blocks that are each an element of $\mathfrak{R}^{6 \times 6}$, as shown in (28).

$$A_p = \begin{bmatrix} A_p(1,1) & A_p(1,2) \\ A_p(2,1) & A_p(2,2) \end{bmatrix}. \quad (28)$$

For this work, we define A_p as a positive multiplicative factor of A_r , which implies that $A_p(1,2)$ is a 0 matrix.

C. Adaptive Control Laws

To complete the adaptive architecture, we begin by defining two error signals:

$$\begin{aligned} e &= x_r - x \\ \hat{e} &= \hat{x} - x \end{aligned} \quad (29)$$

The respective error dynamics are then the following:

$$\begin{aligned} \dot{e} &= A_r e + B_l(u_{ad} - g) \\ \dot{\hat{e}} &= A_p \hat{e} + B_l(u_{ad} - g) \end{aligned} \quad (30)$$

Inspection of (30) reveals that the error terms associated with the filter-states are identically zero. More precisely, the reference model and predictor model reproduce the filter states in x exactly (assuming identical initial conditions and that $A_p(1,2) = 0$), thus these error terms effectively cancel. The error dynamics can then be represented in a reduced form as

$$\begin{aligned} \dot{e}_r &= A_e e_r + B_e(u_{ad} - g) \\ \dot{\hat{e}}_p &= A_{pe} \hat{e}_p + B_e(u_{ad} - g) \end{aligned} \quad (31)$$

where

$$\hat{e}_p = \left[\left(\int_0^t (\hat{\omega} - \omega) d\tau \right)^T \quad (\hat{\omega} - \omega)^T \right]^T, \quad (32)$$

$$e_r = \left[\left(\int_0^t (\omega_f - \omega) d\tau \right)^T \quad (\omega_f - \omega)^T \right]^T, \quad (33)$$

$$A_e = \begin{bmatrix} 0 & I \\ -K_I & -K_p \end{bmatrix}, \quad (34)$$

$$A_{pe} = A_p(2,2), \quad (35)$$

and

$$B_e^T = [0 \quad I]. \quad (36)$$

In (32), $\hat{\omega}$ is the fourth state of the predictor vector \hat{x} . Moreover, in (33) it is important to note that the filter states appear because the reference model specified roll, pitch, and yaw rates are precisely the filter states when the reference model and true system have identical initial conditions, as was discussed in the previous section.

The term g (as defined in (11)) may now be expressed as a function of x , u_{ad} , r , z and a series of time varying weights.

$$g = W_1^T x + W_2^T r + W_3^T z + W_4^T u_{ad} \quad (37)$$

The adaptive control signal is now defined as below:

$$u_{ad} = \hat{W}_1^T x + \hat{W}_2^T r + \hat{W}_3^T z + \hat{W}_4^T u_{ad} \quad (38)$$

The adaptive control laws are given, once again following the work of [5], as in (39).

$$\begin{aligned} \dot{\hat{W}}_1 &= \Gamma_1 \text{Proj}(\hat{W}_1, -x(e_r^T P B_e + \hat{e}_p^T P_p B_e)) \\ \dot{\hat{W}}_2 &= \Gamma_2 \text{Proj}(\hat{W}_2, -r(e_r^T P B_e + \hat{e}_p^T P_p B_e)) \\ \dot{\hat{W}}_3 &= \Gamma_3 \text{Proj}(\hat{W}_3, -u_{ad}(e_r^T P B_e + \hat{e}_p^T P_p B_e)) \\ \dot{\hat{W}}_4 &= \Gamma_4 \text{Proj}(\hat{W}_4, -z(e_r^T P B_e + \hat{e}_p^T P_p B_e)) \end{aligned} \quad (39)$$

In order to be precise on matrix dimensions, we note that $W_1 \in \mathfrak{R}^{12 \times 3}$, $W_2 \in \mathfrak{R}^{3 \times 3}$, $W_3 \in \mathfrak{R}^{3 \times 3}$, and $W_4 \in \mathfrak{R}^{7 \times 3}$. Moreover, as is consistent with most model adaptive control formulations, P and P_p are the solutions to the Lyapunov equations $A_e^T P + P A_e = -Q$ and $A_{pe}^T P_p + P_p A_{pe} = -Q_p$ respectively, where P , P_p , Q , and Q_p are all positive definite, symmetric matrices with Q and Q_p designer selected, tunable parameters. Finally, the “ Γ ” terms are the respective adaptive learning rates for each unknown parameter. It is here worth noting that the matrices in (34) and (35) are constant for the error dynamics, irrespective of the fact that the system is actually time varying.

D. Theoretical Analysis

To show the correctness of the development, three theoretical issues are addressed here: the model matching conditions of the general system in (22), the boundedness of the true and predicted system states, and the boundedness of the two error signals. As aforementioned, all uncertainty has been treated as parametric, thereby simplifying the presentation here. If non-parametric uncertainty were considered, the theoretical development would be further complicated.

1) Model Matching Conditions

The model matching conditions represent the necessary conditions for any control signal to fully cancel the system uncertainty g . If the model matching conditions are satisfied, it is possible for the system to perform exactly as specified in (26). In the current context, if (26) is realized the aircraft will behave as if the dynamic inversion is perfect; the system performance will be completely specified by the command filter in (12) and the choice of PI gains K_I and K_p . The model matching conditions for this system can be determined by combining (10), (14), and (22) to obtain the proceeding result.

$$\begin{aligned} \dot{x} &= \bar{A}x + \bar{B}_m r + \bar{B} [K_x^T x + K_r^T r - \hat{B}^{-1} u_{ad} \\ &\quad + \hat{B}^{-1} (\Delta \tilde{A}x + \Delta B u + \Delta G z)] \end{aligned} \quad (40)$$

In (40) the term $\tilde{\Delta A}$ is introduced to ensure dimensional consistency and is defined as below.

$$\tilde{\Delta A} = \begin{bmatrix} 0 & 0 & 0 & \Delta A \end{bmatrix} \quad (41)$$

Using the definition of each of the Δ terms, the expression in (40) can be reduced to

$$\dot{x} = \bar{A}x + \bar{B}_m r + B_l [(BK_x^T + \tilde{\Delta A})x + BK_r^T r - Bu_{ad} + (G - BK_z^T)z]. \quad (42)$$

Comparing (42) with (26) and temporarily assuming that the B matrix is known (this negates the need for the term \hat{W}_4^T in (38)), we see that the conditions in (43) must hold in order for the uncertainty to be fully canceled.

$$\begin{aligned} \bar{A} + B_l BK_x^T + B_l \tilde{\Delta A} &= A_r \\ \bar{B}_m + B_l BK_r^T &= B_r \\ G - BK_z^T &= 0 \end{aligned} \quad (43)$$

More precisely put, there must exist gains K_x , K_r , and K_z such that the expressions in (43) are satisfiable in order for the *reference* model in (34) to be fully-achievable. These conditions are relevant to any adaptive control technology in that they explicitly indicate that the control objectives (to track the reference model) may not be achievable if the system's B matrix is not of full rank (or the uncertainty is not in the span of B). However, because the B matrix is not known, the adaptive control signal includes itself and there is a fixed-point problem. This is commensurate with the development presented in [1]. It should be stressed that making specific assumptions about the uncertainty in B can eliminate the fixed-point problem.

2) Bounded Error Dynamics

The error dynamics are already presented in (30-36). To show that the error dynamics are bounded, we use the candidate Lyapunov function presented in (44).

$$\begin{aligned} V = & e_r^T P e_r + \hat{e}_p^T P_p \hat{e}_p + \text{trace}(\Delta W_1^T \Gamma_1^{-1} \Delta W_1) + \\ & \text{trace}(\Delta W_2^T \Gamma_2^{-1} \Delta W_2) + \text{trace}(\Delta W_3^T \Gamma_3^{-1} \Delta W_3) + \\ & \text{trace}(\Delta W_4^T \Gamma_4^{-1} \Delta W_4) \end{aligned} \quad (44)$$

Here, Δ denotes the difference between the predicted and true weight value. Examining (2), (11), and (37), it is clear that the unknown weights must be time varying. Taking the derivative of (44) and substituting the error dynamics in (30), as well as the control laws in (39) then yields

$$\begin{aligned} \dot{V} = & -e_r^T Q e_r - \hat{e}_p^T Q_p \hat{e}_p - 2\text{trace}(\Delta W_1^T \Gamma_1^{-1} \dot{W}_1) \\ & - 2\text{trace}(\Delta W_2^T \Gamma_2^{-1} \dot{W}_2) - 2\text{trace}(\Delta W_3^T \Gamma_3^{-1} \dot{W}_3) \\ & - 2\text{trace}(\Delta W_4^T \Gamma_4^{-1} \dot{W}_4) \end{aligned} \quad (45)$$

As a result of the time varying nature of the weights, the stability proof requires the existence of bounds on each of the weights as well as bounds on the derivatives of the

weights. The use of the projection operator in the control laws then ensures that the Δ terms are bounded. More precisely, we must assume that

$$W_1(t) \in \bar{W}_1, W_2(t) \in \bar{W}_2, W_3(t) \in \bar{W}_3, W_4(t) \in \bar{W}_4 \quad \forall t, \quad (46)$$

where \bar{W} denotes a known compact set and

$$\|W_1(t)\| \leq d_1, \|W_2(t)\| \leq d_2, \|W_3(t)\| \leq d_3, \|W_4(t)\| \leq d_4 \quad \forall t. \quad (47)$$

If these bounds exist, the system errors (both the predicted and actual errors) are uniformly ultimately bounded. Moreover, because the reference system is bounded-input bounded-output stable, it follows that both the true system states and the predicted system states are bounded as well. It should be noted that the result in (45) is derived from the approach taken in [6] for an L_1 adaptive controller.

IV. SIMULATION RESULTS

To investigate the functionality of the system, we use an up-scaled version of NASA's Generic Transport Model (GTM - developed at NASA Langley Research Center); this model is intended to represent a scaled version of a two engine commercial transport aircraft [7]. Using real aerodynamic data, the above controller is simulated in a full-nonlinear simulation environment. As an added caveat, data for the dynamic inversion is collected using the vortex-lattice code base VorView [8]; this ensures a separation between the true aircraft model and the dynamic inversion. For the purposes of this paper, we consider an aircraft performing a doublet maneuver in roll, pitch, and sideslip. In one scenario, we consider a nominal aircraft; in the second scenario, we consider an aircraft having lost control of the left elevator and its stab. Presented in Fig. 3 are the complete results of this comparison; Figure 4 presents an enlarged view of the pitch tracking.

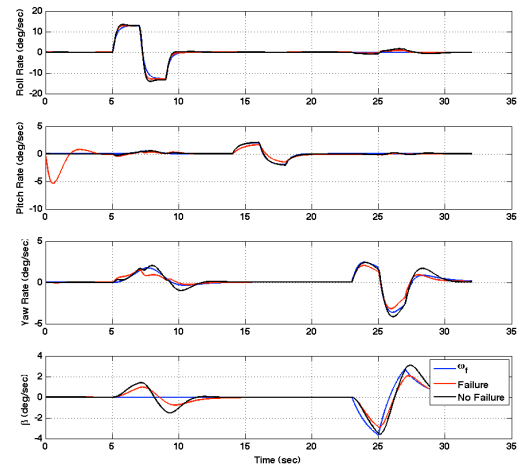


Fig. 3. Damage and undamaged aircraft performing doublet maneuver with baseline controller.

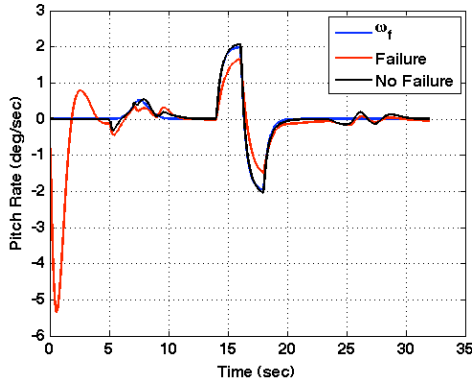


Fig. 4. Enlarged view of pitch tracking from Fig. 3.

From the results presented in Fig. 3 and Fig. 4, it is now desired to arrest the pitch deviation at onset of the damage as well as return the undamaged tracking performance by using an adaptive PMRAC controller. For the purpose of comparison, results are also presented for an MRAC controller using identical adaptive learning rates to that of the PMRAC controller. Results are presented in Fig. 5-7.

As can be seen from the results, the PMRAC augmented controller has demonstrably fewer oscillations than MRAC (at the given learning rates). Additionally, both controllers have reduced the initial pitch loss and recovered the pitch doublet tracking performance of the nominal aircraft. As is consistent with the literature, the control activity for PMRAC is also significantly reduced as compared to MRAC. It is worth noting that the MRAC performance may be improved with gain reduction and the inclusion of normalized learning rates.

V. CONCLUSION

This paper has presented a baseline nonlinear dynamic inversion controller with a PMRAC adaptive augmentation component. Simulation results have demonstrated that this architecture can improve tracking performance compared to both the un-augmented baseline controller and a traditional MRAC controller.

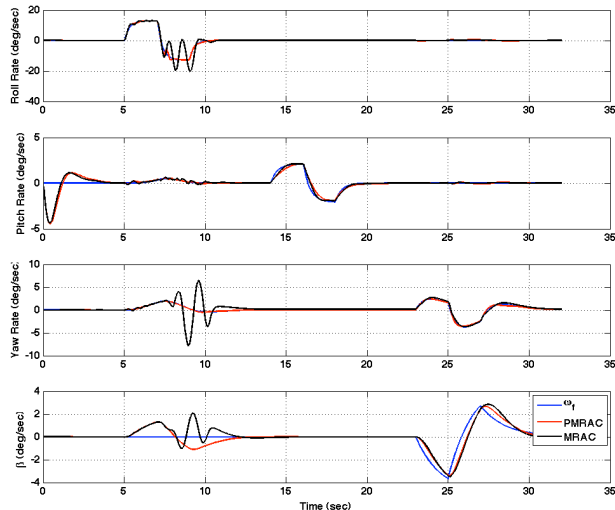


Fig. 5. Results for GTM aircraft with adaptive control augmentation.

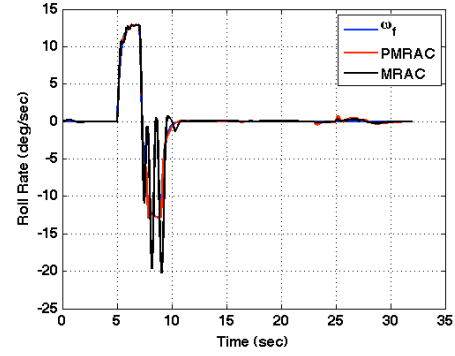


Fig. 6. Enlarged view of roll tracking from Fig. 5.

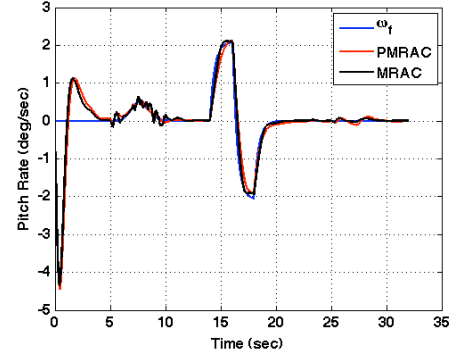


Fig. 7. Enlarged view of pitch tracking from Fig. 5.

ACKNOWLEDGMENT

Thanks must be given to Dr. Eugene Lavretsky of the Boeing Group and NASA Adaptive Controls and Evolvable Systems group members Kalmanje Krishnakumar (NASA), Nhan Nguyen (NASA), Greg Larchev, Shivanjli Sharma, and Vahram Stepanyan.

REFERENCES

- [1] R. T. Rysdyk and A. J. Calise, "Fault Tolerant Flight Control via Adaptive Neural Augmentation", *AIAA Guidance, Navigation, and Control Conference*, Aug. 1998.
- [2] J. T. Kaneshige, J. Bull, and J. J. Totah, "Generic Neural Flight Control and Autopilot System," *AIAA Guidance, Navigation, and Control Conference*, Aug. 2000.
- [3] N. Nguyen, K. Krishnakumar, J. T. Kaneshige, and P. Nespeca, "Flight Dynamics and Hybrid Adaptive Control of Damaged Aircraft", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, pp. 751-764, 2008.
- [4] S. A. Snell, D. F. Enns, and W. L. Garrard Jr., "Nonlinear Inversion Flight Control for a Superautonomous Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, July-August 1992.
- [5] E. Lavretsky, R. Gadiant, I. Gregory, "Predictor-Based Model Reference Adaptive Control," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, Aug. 10-13, 2009.
- [6] J. Wang, N. Hovakimyan, C. Cao, "L₁ Adaptive Augmentation of Gain-Scheduled Controller for Racetrack Maneuver in Aerial Refueling," *AIAA Guidance, Navigation, and Control Conference*, Aug. 2009, Chicago, IL.
- [7] T.L. Jordan, W. M. Langford, et al., "Development of a Dynamically Scaled Generic Transport Model Testbed for Flight Research Experiments," *AUVSI Unmanned Systems North America 2004*, Arlington, VA, 2004.
- [8] J. J. Totah, D. J. Kinney, J. T. Kaneshige, and S. Agabon, "An Integrated Vehicle Modeling Environment," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Portland, OR, Aug 9-11, 1999.