JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, March 2009

Hybrid Intelligent Flight Control with Adaptive Learning Parameter Estimation

Nhan Nguyen* and Kalmanje Krishnakumar†
NASA Ames Research Center, Moffett Field, CA 94035

DOI: 10.2514/1.35929

This paper presents a recent development of a hybrid adaptive control method that further extends the intelligent flight control technology to improve command-tracking performance of aircraft operating in off-nominal flight conditions. The hybrid method is based on adaptive learning laws for online parameter estimation of aircraft plant dynamics in conjunction with an existing neural net direct adaptation strategy. Two parameter estimation learning laws are presented: 1) an indirect adaptive learning law derived from the Lyapunov stability theory, and 2) a recursive least-squares learning law. Simulations of a damaged generic aircraft demonstrate the effectiveness of the proposed hybrid intelligent flight control method.

Nomenclature

$\mathbf{A}$	state matrix for tracking error
$\mathbf{B}$	control matrix for tracking error
$\boldsymbol{C}$	neural network basis functions
$\boldsymbol{e}$	tracking error
$\mathbf{F}_1, \mathbf{F}_2$	state transition matrices for angular rates and trim states
$\mathbf{G}$	control influence matrix for flight control surfaces
$\mathbf{I}$	identity matrix
J	cost function
$\mathbf{K}_P, \mathbf{K}_I$	proportional and integral gain matrices
L, V	Lyapunov candidate functions
$\mathbf{P}$	Lyapunov matrix
p, q, r	roll, pitch, and yaw rates
$\mathbf{Q}$	semi-positive definite matrix
$\mathbf{R}$	positive definite matrix
T_0	persistent excitation sampling period
u, v, w, V	forward, lateral, vertical, and absolute airspeeds
$\boldsymbol{u}_e, \boldsymbol{u}_{\text{ad}}$	pseudo-control and neural net command augmentation control vectors
$\mathbf{W}, \mathbf{W}_{\omega}, \mathbf{W}_{\sigma}, \mathbf{W}_{\delta}$	neural network weight matrices
$\alpha, \beta, \theta, \phi$	angles of attack, sideslip, pitch, and bank
$\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1$	persistent excitation levels
$\beta, \beta_{\omega}, \beta_{\sigma}, \beta_{\delta}$	composite neural network basis functions

Received 29 November 2007; revision received 4 March 2008; accepted for publication 10 April 2008.

* Computer Scientist, Intelligent Systems Division, Mail Stop 269-1, AIAA Senior Member, Nhan.T.Nguyen@nasa.gov
† Computer Scientist, Intelligent Systems Division, Mail Stop 269-1, AIAA Associate Fellow, Kalmanje.S.Krishnakumar @nasa.gov

Γ, $\mathbf{\Lambda}$	learning rate constant and matrix
Δ	difference operator
Δ	error bound
$\boldsymbol{\delta}$	control surface deflection vector
δ_T, δ_a, δ_e, δ_r	engine throttle, and aileron, elevator, and rudder deflections
$\boldsymbol{\varepsilon}$	dynamic inversion error
ζ_p, ζ_q, ζ_r	reference-model roll, pitch, and yaw damping ratios
η, ϑ, λ, ν, φ, ξ	recursive least-squares forgetting factor parameters
$\boldsymbol{\theta}$, $\boldsymbol{\phi}$	input and neural net weight vectors
$\lambda_{\min}$, $\lambda_{\max}$	minimum and maximum eigenvalue
μ, $\boldsymbol{\eta}$	e-modification parameter constant and matrix
$\boldsymbol{\sigma}$	trim state vector
$\boldsymbol{\omega}$, $\boldsymbol{\omega}_e$, $\boldsymbol{\omega}_d$	angular rate, angular rate error, and desired angular rate vectors
$\boldsymbol{\omega}_c$, $\boldsymbol{\omega}_m$, $\boldsymbol{\omega}_n$	angular rate command vector, reference model vector, reference model frequency matrix
ω_p, ω_q, ω_r	reference model roll, pitch, and yaw frequencies

Subscripts

i	sampling period
j	time step

Superscripts

$\hat{}$	estimated value
$\tilde{}$	neural network weight matrix variation
*	ideal value

I. Introduction

AN aircraft flight control system is carefully designed to provide good stability and command-following performance to achieve mission objectives. Most conventional flight control systems use extensive gain-scheduling to achieve desired handling qualities. While this approach has proved to be very successful, the development process can be expensive and often results in aircraft-specific implementations. Moreover, under off-nominal flight conditions owing to external factors such as wind shear and turbulence or aircraft degradation resulting from failures or damage, a flight control system will need to be able to adapt to these sources of uncertainty. For example, during a damage situation, numerous effects may be present that can overwhelm a pilot's ability to control an aircraft safely. These effects may include aerodynamic changes, structural degradation, engine damage, reduced flight control effectiveness, and others. The conventional gain-scheduling approach is not sufficiently flexible to provide this online adaptation to account for these unknown dynamics.

Over the past several years, various adaptive control techniques have been investigated [1–9]. Fault-tolerant adaptive flight control has been developed to provide a possibility for maintaining aircraft stability and performance by means of enabling a flight control system to adapt to system uncertainties [10,11]. Meanwhile, a large area of research in neural net adaptive control has emerged and spans many different applications such as spacecraft and aircraft flight control. An essential element of a neural net adaptive flight control is a neural network system designed to accommodate changes in aircraft dynamics owing to system uncertainties. Neural network is known to be a good universal approximator of many nonlinear functions that can be used to model system uncertainties. In the implementation of a neural net adaptive flight control, the neural network is typically incorporated within a direct adaptive control architecture to provide an augmentation to a pilot command. The neural network estimates the system uncertainties and outputs directly a command augmentation signal that compensates for changes in aircraft dynamics.

Research in adaptive control has spanned several decades, but challenges in obtaining robustness in the presence of unmodeled dynamics, parameter uncertainties, or disturbances as well as the issues with verification and validation of adaptive flight control software still remain [2,12]. Adaptive control laws may be divided into direct and indirect approaches. Indirect adaptive control methods typically employ parameter identification techniques to compute

control parameters [13]. Parameter identification techniques such as recursive least-squares and neural networks have been used in indirect adaptive control methods [3]. In recent years, model-reference direct adaptive control using neural networks has been a topic of great research interest [4–6]. Lyapunov stability theory has been used to establish stability of neural network learning laws.

NASA has been developing an intelligent flight control technology based on the work by Rysdyk and Calise [4]. Recently, this technology has been demonstrated on an F-15 fighter aircraft [14]. The intelligent flight control uses a model-reference, direct adaptive, dynamic inversion control approach. The neural net direct adaption is designed to provide consistent handling qualities without requiring extensive gain-scheduling or explicit system identification. This particular architecture uses both pre-trained and online learning neural networks and a reference model to specify desired handling qualities. Pretrained neural networks are used to provide estimates of aerodynamic stability and control characteristics. Online learning neural networks are used to compensate for errors and adapt to changes in aircraft dynamics. As a result, consistent handling qualities may be achieved across flight conditions.

One potential problem with a neural net direct adaptive flight control is the possibility of a high gain control owing to aggressive learning, which is characterized by a large learning rate for the neural network learning law to reduce the error signal rapidly. The high gain control signal can potentially result in a control augmentation command that may saturate the control authority or excite unmodeled dynamics of the plant and can adversely affect the stability of the direct adaptive learning law. Moreover, during off-nominal flight conditions, the aircraft plant dynamics are degraded and this can consequently present a problem for a pilot to safely navigate the aircraft within a reduced flight envelope. For example, changes in stability and control derivatives can potentially cause a pilot to apply excessive or incorrect stick commands that could worsen the aircraft handling qualities. Direct adaptive control approaches accommodate changes in plant dynamics implicitly but do not provide an explicit means for ascertaining the knowledge of the plant dynamics, which can potentially be used to improve the control law. Furthermore, the improved knowledge of the plant dynamics can potentially be used to develop fault detection and isolation strategies, as well as emergency flight planning to provide guidance laws for energy management during a descent, approach, and landing.

Toward this end, we introduce a new method for augmenting the existing direct adaptive intelligent flight control that provides an opportunity to perform an online parameter estimation of the aircraft plant dynamics explicitly. We call this approach a hybrid intelligent flight control. Two adaptive learning laws are developed to provide parameter estimation of the plant dynamics that will be used in the dynamic inversion controller. As a result, the control command will result in a smaller dynamic inversion error so that the learning of the direct adaptation neural network can be reduced.

II. Hybrid Intelligent Flight Control Architecture

Figure 1 illustrates the proposed hybrid intelligent flight control. The control architecture comprises: 1) a reference model that translates rate commands into desired acceleration commands; 2) a proportional–integral (PI) feedback control for rate stabilization and tracking; 3) a dynamic inversion controller that computes actuator commands using

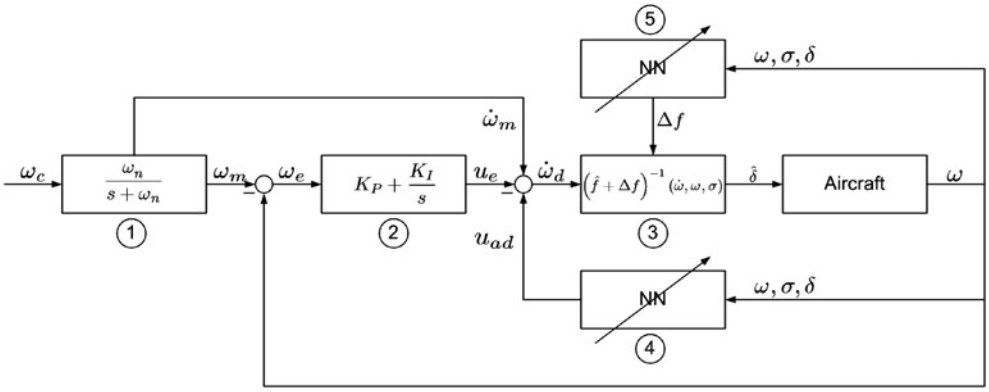

Fig. 1 Hybrid intelligent flight control.

desired acceleration commands; 4) a neural net direct adaptation owing to Rysdyk and Calise [4], and 5) a neural net indirect adaptation that performs parameter estimation of the aircraft plant dynamics. In the present study, we have developed two learning laws for the parameter estimation: 1) an indirect adaptive learning law based the Lyapunov stability theory and 2) a recursive least-squares learning law based on the optimal estimation theory.

Under off-nominal flight conditions such as failure and or damage events, aircraft plant dynamics can deviate significantly from the nominal behaviors. Consequently, this can pose a significant challenge to a flight control system. To develop an adaptive control strategy, consider the linearized equations of angular motion of an aircraft operating in off-nominal flight conditions as

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}^* + \Delta\dot{\boldsymbol{\omega}} = \mathbf{F}_1\boldsymbol{\omega} + \mathbf{F}_2\boldsymbol{\sigma} + \mathbf{G}\boldsymbol{\delta} \tag{1}$$

where $\dot{\boldsymbol{\omega}}^*$ is the ideal angular acceleration; $\Delta\dot{\boldsymbol{\omega}}$ is the perturbation term owing to changes in aircraft plant dynamics and uncertainties; and $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{G}$ are the plant matrices.

The ideal aircraft plant dynamics is expressed in a state space form as

$$\dot{\boldsymbol{\omega}}^* = \mathbf{F}_1^*\boldsymbol{\omega} + \mathbf{F}_2^*\boldsymbol{\sigma} + \mathbf{G}^*\boldsymbol{\delta} \tag{2}$$

where $\boldsymbol{\omega} = [p\ q\ r]^\top$ is a vector of the aircraft roll, pitch, and yaw rates; $\boldsymbol{\sigma} = [\Delta\alpha\ \Delta\beta\ \Delta\phi\ \Delta\delta_T]^\top$ is a vector of the variations in trim angle of attack, sideslip angle, bank angle, and engine throttle; $\boldsymbol{\delta} = [\delta_a\ \delta_e\ \delta_r]^\top$ is a vector of the aileron, elevator, and rudder deflections; and $\mathbf{F}_1^*$, $\mathbf{F}_2^*$, and $\mathbf{G}^*$ are the nominal aircraft plant matrices which are assumed to be known.

The plant uncertainty is casted in the form of

$$\Delta\dot{\boldsymbol{\omega}} = \Delta\mathbf{F}_1\boldsymbol{\omega} + \Delta\mathbf{F}_2\boldsymbol{\sigma} + \Delta\mathbf{G}\boldsymbol{\delta} \tag{3}$$

where $\Delta\mathbf{F}_1$, $\Delta\mathbf{F}_2$, and $\Delta\mathbf{G}$ are the unknown plant matrices owing to system uncertainties.

A reference model is used to specify desired handling qualities. The control adaptation must be able to accommodate system uncertainties using the available flight control surfaces. A reference model is used to filter a rate command $\boldsymbol{\omega}_c$ into a reference angular rate $\boldsymbol{\omega}_m$ and a reference angular acceleration $\dot{\boldsymbol{\omega}}_m$ via a first-order model

$$\dot{\boldsymbol{\omega}}_m + \boldsymbol{\omega}_n\boldsymbol{\omega}_m = \boldsymbol{\omega}_n\boldsymbol{\omega}_c \tag{4}$$

where $\boldsymbol{\omega}_n = \text{diag}(\omega_p, \omega_q, \omega_r)$ is a reference model frequency matrix.

The reference model frequency parameters must be chosen appropriately to obtain a good transient response that satisfies position and rate limits on the control surface deflection. Generally, the goal of an adaptive flight control is to be able to follow the reference model of the nominal aircraft. It is possible, however, that the adaptive flight control may not be able to track the reference model if the aircraft performance is significantly reduced under adverse conditions. In such a situation, it may become necessary to change the reference model so that the flight control system can better respond to a pilot command. The adjustment of the reference model parameters can be performed using various methods such as the adaptive-critic approach [8] to ensure that the flight control can track the reference model to achieve desired handling qualities. Thus, an effective control strategy may combine both the reference model adjustment as well as the adaptive flight control.

The reference model angular rate $\boldsymbol{\omega}_m$ is compared with the actual angular rate output $\boldsymbol{\omega}$ to form a tracking error signal $\boldsymbol{\omega}_e = \boldsymbol{\omega}_m - \boldsymbol{\omega}$. A pseudo-feed back control $\boldsymbol{u}_e$ is constructed using a PI feedback scheme to regulate errors in the roll , pitch, and yaw rates. The error dynamics must have a good frequency response that can track the reference model without an interference with actuator dynamics. The pseudo-control vector $\boldsymbol{u}_e$ is computed as

$$\boldsymbol{u}_e = \mathbf{K}_P\boldsymbol{\omega}_e + \mathbf{K}_I\int_0^t \boldsymbol{\omega}_e\,\mathrm{d}\tau \tag{5}$$

To maintain a reasonable tracking performance and loop gains, it is found that the proportional and integral gains can be selected to achieve a second-order error dynamics according to

$$\mathbf{K}_P = \text{diag}(2\zeta_p\omega_p, 2\zeta_q\omega_q, 2\zeta_r\omega_r) \tag{6}$$

$$\mathbf{K}_I = \text{diag}(\omega_p^2, \omega_q^2, \omega_r^2) \tag{7}$$

A dynamic inversion controller is computed to obtain an estimated control surface deflection command $\hat{\boldsymbol{\delta}}$ to achieve the desired angular acceleration $\dot{\boldsymbol{\omega}}_d$ using an estimated plant dynamics

$$\hat{\boldsymbol{\delta}} = \hat{\mathbf{G}}^{-1}(\dot{\boldsymbol{\omega}}_d - \hat{\mathbf{F}}_1\boldsymbol{\omega} - \hat{\mathbf{F}}_2\boldsymbol{\sigma}) \tag{8}$$

$$\hat{\boldsymbol{\delta}} = \mathbf{G}^{-1}(\dot{\boldsymbol{\omega}}_d - \mathbf{F}_1\boldsymbol{\omega} - \mathbf{F}_2\boldsymbol{\sigma}) \tag{9}$$

where $\hat{\mathbf{F}}_1 = \mathbf{F}_1^* + \Delta\hat{\mathbf{F}}_1$, $\hat{\mathbf{F}}_2 = \mathbf{F}_2^* + \Delta\hat{\mathbf{F}}_2$, $\hat{\mathbf{G}} = \mathbf{G}^* + \Delta\hat{\mathbf{G}}$ are the estimated off-nominal plant matrices and $\hat{\mathbf{G}}$ is assumed to be invertible.

As the true damaged plant dynamics are different from the ideal plant dynamics, the dynamic inversion controller incurs a dynamic inversion error equal to

$$\boldsymbol{\varepsilon} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_d = \Delta\boldsymbol{\varepsilon} - \Delta\hat{\mathbf{F}}_1\boldsymbol{\omega} - \Delta\hat{\mathbf{F}}_2\boldsymbol{\sigma} - \Delta\hat{\mathbf{G}}\hat{\boldsymbol{\delta}} \tag{10}$$

where $\Delta\boldsymbol{\varepsilon} = \Delta\dot{\boldsymbol{\omega}}$.

In order for the dynamic inversion controller to track the reference model angular acceleration $\dot{\boldsymbol{\omega}}_m$, the desired angular acceleration $\dot{\boldsymbol{\omega}}_d$ is set equal to

$$\dot{\boldsymbol{\omega}}_d = \dot{\boldsymbol{\omega}}_m + \boldsymbol{u}_e - \boldsymbol{u}_{\text{ad}} \tag{11}$$

where $\boldsymbol{u}_{\text{ad}}$ is a direct adaptive control augmentation signal designed to cancel out the dynamic inversion error so that, in an ideal setting, as the tracking error goes to zero asymptotically, the desired angular acceleration $\dot{\boldsymbol{\omega}}_d$ is equal to the reference model angular acceleration $\dot{\boldsymbol{\omega}}_m$.

The tracking error dynamics then becomes

$$\dot{\boldsymbol{e}} = \mathbf{A}\boldsymbol{e} + \mathbf{B}\boldsymbol{u}_{\text{ad}} + \mathbf{B}\Delta\hat{\mathbf{F}}_1\boldsymbol{\omega} + \mathbf{B}\Delta\hat{\mathbf{F}}_2\boldsymbol{\sigma} + \mathbf{B}\Delta\hat{\mathbf{G}}\hat{\boldsymbol{\delta}} - \mathbf{B}\Delta\boldsymbol{\varepsilon} \tag{12}$$

where $\boldsymbol{e} = \left[\int_0^t \boldsymbol{\omega}_e \,\mathrm{d}\tau \;\; \boldsymbol{\omega}_e\right]^\top$ is the tracking error and

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_I & -\mathbf{K}_P \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$$

The direct adaptive signal $\boldsymbol{u}_{\text{ad}}$ is computed from a single-layer sigma-pi neural network

$$\boldsymbol{u}_{\text{ad}} = \mathbf{W}^\top\boldsymbol{\beta}(\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3, \boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{C}_6) \tag{13}$$

where $\boldsymbol{\beta}$ is a vector of basis functions computed using Kronecker products with $\boldsymbol{C}_i$, $i = 1, \ldots, 6$, as inputs into the neural network consisting of control commands, sensor feedback, and bias terms.

In particular, the Kronecker product terms are

$$\boldsymbol{C}_1 = \rho V^2 \begin{bmatrix} \boldsymbol{\omega}^\top & \alpha\boldsymbol{\omega}^\top & \beta\boldsymbol{\omega}^\top \end{bmatrix} \tag{14}$$

$$\boldsymbol{C}_2 = \rho V^2 \begin{bmatrix} 1 & \alpha & \beta & \alpha^2 & \beta^2 & \alpha\beta \end{bmatrix} \tag{15}$$

$$\boldsymbol{C}_3 = \rho V^2 \begin{bmatrix} \boldsymbol{\delta}^\top & \alpha\boldsymbol{\delta}^\top & \beta\boldsymbol{\delta}^\top \end{bmatrix} \tag{16}$$

$$\boldsymbol{C}_4 = \begin{bmatrix} p\boldsymbol{\omega}^\top & q\boldsymbol{\omega}^\top & r\boldsymbol{\omega}^\top \end{bmatrix} \tag{17}$$

$$\boldsymbol{C}_5 = \begin{bmatrix} u\boldsymbol{\omega}^\top & v\boldsymbol{\omega}^\top & w\boldsymbol{\omega}^\top \end{bmatrix} \tag{18}$$

$$\boldsymbol{C}_6 = \begin{bmatrix} 1 & \theta & \phi & \delta_T \end{bmatrix} \tag{19}$$

The basis function is then expressed as

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 & \boldsymbol{C}_3 & \boldsymbol{C}_4 & \boldsymbol{C}_5 & \boldsymbol{C}_6 \end{bmatrix}^\top \tag{20}$$

Figure 2 illustrates the sigma-pi neural network with an optional hidden layer.

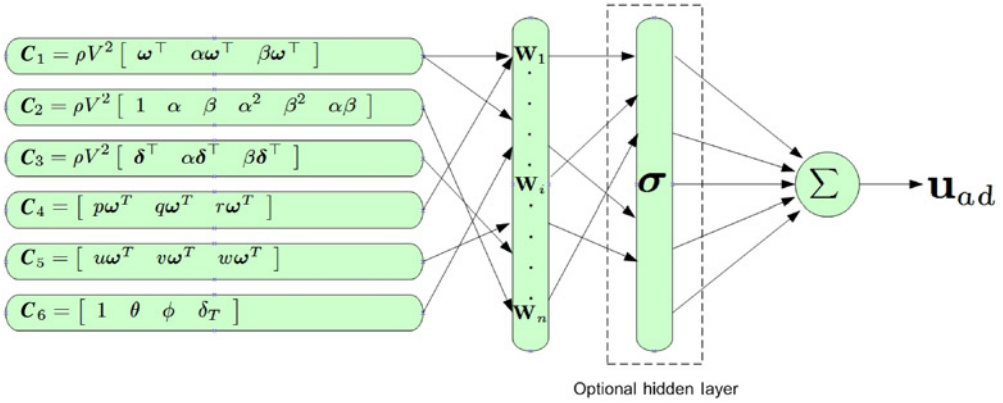

Fig. 2 Sigma-pi neural network.

The update law for the neural net weights $\mathbf{W}$ is [4]

$$\dot{\mathbf{W}} = -\Gamma\left(\boldsymbol{\beta}\boldsymbol{e}^\top\mathbf{P}\mathbf{B} + \mu\left\|\boldsymbol{e}^\top\mathbf{P}\mathbf{B}\right\|\mathbf{W}\right) \tag{21}$$

where $\Gamma > 0$ is the learning rate, $\mu > 0$ is a parameter to improve robustness based on the e-modification scheme [7], $\mathbf{P}$ solves the Lyapunov equation $\mathbf{A}^\top\mathbf{P} + \mathbf{P}^\top\mathbf{A} = -\mathbf{Q}$ for some positive-definite matrix $\mathbf{Q}$, and $\|.\|$ is a Frobenius norm.

The key feature of the hybrid intelligent flight control is a neural net indirect adaptation strategy designed to perform online parameter estimation of the plant matrices $\hat{\mathbf{F}}_1$, $\hat{\mathbf{F}}_2$, $\hat{\mathbf{G}}$ using a suitable adaptive learning law. The parameter estimation enables the dynamic inversion controller to be updated continuously. As the adaptive learning law reduces the estimation error, the accuracy of the dynamic inversion controller increases, thereby resulting in a decrease in the tracking error. Any residual tracking error can then be handled by the neural net direct adaptive control.

A. Indirect Adaptive Learning Law

In this approach, we would like to update the dynamic inversion controller using an indirect adaptive learning law that performs online estimation of the unknown plant dynamics owing to system uncertainties. The indirect adaptive learning law is given by

$$\dot{\mathbf{\Phi}} = -\mathbf{\Lambda}\left(\boldsymbol{\theta}\boldsymbol{e}^\top\mathbf{P}\mathbf{B} + \boldsymbol{\eta}\left\|\boldsymbol{e}^\top\mathbf{P}\mathbf{B}\right\|\mathbf{\Phi}\right) \tag{22}$$

where $\mathbf{\Phi}^\top = \left[\mathbf{W}_\omega^\top\ \mathbf{W}_\sigma^\top\ \mathbf{W}_\delta^\top\right]$ is a neural net weight matrix, $\boldsymbol{\theta}^\top = [\boldsymbol{\omega}^\top\boldsymbol{\beta}_\omega^\top\ \boldsymbol{\sigma}^\top\boldsymbol{\beta}_\sigma^\top\ \hat{\boldsymbol{\delta}}^\top\boldsymbol{\beta}_\delta^\top]$ is an input matrix, $\mathbf{\Lambda} = \mathrm{diag}(\Gamma_\omega, \Gamma_\sigma, \Gamma_\delta) > 0$ is a learning rate matrix, and $\boldsymbol{\eta} = \mathrm{diag}(\mu_\omega, \mu_\sigma, \mu_\delta)$ is the e-modification parameter matrix which provides robustness to unmodeled dynamics [7].

The estimated plant dynamics are then updated as

$$\hat{\mathbf{F}}_1 = \mathbf{F}_1^* + \mathbf{W}_\omega^\top\boldsymbol{\beta}_\omega \tag{23}$$

$$\hat{\mathbf{F}}_2 = \mathbf{F}_2^* + \mathbf{W}_\sigma^\top\boldsymbol{\beta}_\sigma \tag{24}$$

$$\hat{\mathbf{G}} = \mathbf{G}^* + \mathbf{W}_\delta^\top\boldsymbol{\beta}_\delta \tag{25}$$

where $\mathbf{W}_\omega$, $\mathbf{W}_\sigma$, and $\mathbf{W}_\delta$ are the corresponding neural net weight matrices; and $\boldsymbol{\beta}_\omega$, $\boldsymbol{\beta}_\sigma$, and $\boldsymbol{\beta}_\delta$ are neural net basis functions which can be any suitable subsets of $\boldsymbol{\beta}$ such as

$$\boldsymbol{\beta}_\omega = \rho_a V^2 \begin{bmatrix} 1 & \alpha & \beta \\ 1 & \alpha & \beta \\ 1 & \alpha & \beta \end{bmatrix}^\top \tag{26}$$

$$\boldsymbol{\beta}_{\boldsymbol{\sigma}} = \rho_a V^2 \tag{27}$$

$$\boldsymbol{\beta}_{\delta} = \boldsymbol{\beta}_{\boldsymbol{\omega}} \tag{28}$$

The proof of this indirect adaptive learning law is provided in Appendix A.

The input signals that feed into the neural net indirect adaptive learning law are $\boldsymbol{\omega}$, $\boldsymbol{\sigma}$, and $\hat{\boldsymbol{\delta}}$. As the sigma-pi neural network is linear in parameters, a potential problem can exist if the input signals for some reason become unbounded. Physically, this scenario can correspond to large input signals which can lead to a high-gain control situation. Suppose that $\boldsymbol{\omega}, \boldsymbol{\sigma}, \hat{\boldsymbol{\delta}} \notin \mathcal{L}_\infty$, the indirect adaptive learning law will become unstable because $\boldsymbol{\omega}$, $\boldsymbol{\sigma}$, and $\hat{\boldsymbol{\delta}}$ are unbounded. In this case, we must modify the indirect adaptive learning law to handle large input signals by a normalization method [13]. Toward that end, we note that $\boldsymbol{\omega}(1 + \boldsymbol{\omega}^\top \mathbf{R}\boldsymbol{\omega})^{-1} \in \mathcal{L}_\infty$ even though $\boldsymbol{\omega} \notin \mathcal{L}_\infty$ as

$$\lim_{\omega \to \infty} \frac{\boldsymbol{\omega}}{1 + \boldsymbol{\omega}^\top \mathbf{R}\boldsymbol{\omega}} = 0 \tag{29}$$

where $\mathbf{R}$ is a positive-definite weight matrix.

The indirect adaptive learning law for unbounded input signals can therefore be normalized as

$$\dot{\boldsymbol{\Phi}} = -\boldsymbol{\Lambda}_a \left(\boldsymbol{\theta} \boldsymbol{e}^\top \mathbf{P}\mathbf{B} + \eta \left\| \boldsymbol{e}^\top \mathbf{P}\mathbf{B} \right\| \boldsymbol{\Phi} \right) \tag{30}$$

where $\boldsymbol{\Lambda}_a$ is a normalized learning rate matrix defined as

$$\boldsymbol{\Lambda}_a = \boldsymbol{\Lambda} \left(\mathbf{I} + \boldsymbol{\theta}^\top \mathbf{R}\boldsymbol{\theta} \right)^{-1} \tag{31}$$

and $\mathbf{I}$ is the identity matrix.

Thus, the effect of the normalized learning rate is such that it reduces the learning process when the input signals become large to prevent any instability in the learning algorithm. As the learning algorithm become stable, the learning rate can increase to decrease the error further.

B. Recursive Least-Squares Learning Law

Another neural net learning law for online parameter estimation is the well-known recursive least-squares method. If the dynamic inversion error can be estimated, then a recursive least-squares learning law can be applied to determine the weight matrices $\mathbf{W}_{\boldsymbol{\omega}}$, $\mathbf{W}_{\boldsymbol{\sigma}}$, and $\mathbf{W}_{\delta}$. Suppose the estimated dynamic inversion error can be written as

$$\boldsymbol{\varepsilon} = \boldsymbol{\Phi}^\top \boldsymbol{\theta} + \Delta\boldsymbol{\varepsilon} \tag{32}$$

where $\Delta\boldsymbol{\varepsilon}$ is the estimation error of $\Delta\dot{\boldsymbol{\omega}}$.

Then, the estimated dynamic inversion error is

$$\hat{\boldsymbol{\varepsilon}} = \dot{\hat{\boldsymbol{\omega}}} - \mathbf{F}_1^* \boldsymbol{\omega} - \mathbf{F}_2^* \boldsymbol{\sigma} - \mathbf{G}^* \hat{\boldsymbol{\delta}} \tag{33}$$

where $\dot{\hat{\boldsymbol{\omega}}}$ is the estimated angular acceleration which may be subject to computational errors.

Generally, the angular acceleration may not be available from rate gyro sensors. In such a case, one method of estimating $\dot{\hat{\boldsymbol{\omega}}}$ is to use a backward finite-difference method

$$\dot{\hat{\boldsymbol{\omega}}}_j = \frac{\tilde{\boldsymbol{\omega}}_j - \tilde{\boldsymbol{\omega}}_{j-1}}{\Delta t} \tag{34}$$

to estimate $\dot{\hat{\boldsymbol{\omega}}}$ at the jth time step, but this method can result in a significant error if Δt is either too small or too large. The derivative computation will introduce an error source $\Delta\boldsymbol{\varepsilon}$. If the error is unbiased, that is, it can be characterized as a white noise about the mean value, then the recursive least-squares learning law can be applied to estimate the off-nominal plant dynamics.

The recursive least-squares learning law is given by

$$\dot{\mathbf{\Phi}} = (1+\xi)^{-1}\mathbf{R}\boldsymbol{\theta}\left(\hat{\boldsymbol{\varepsilon}}^{\top} - \boldsymbol{\theta}^{\top}\mathbf{\Phi}\right) \tag{35}$$

where

$$\dot{\mathbf{R}} = -(1+\xi)^{-1}\boldsymbol{\theta}^{\top}\mathbf{R} \tag{36}$$

$$\xi = \boldsymbol{\theta}^{\top}\mathbf{R}\boldsymbol{\theta} \tag{37}$$

The matrix $\mathbf{R}$ is called the covariance matrix and the recursive least-squares learning law has a very similar form to the Kalman filter with Eq. (36) as a differential Riccati equation for a zero-order plant dynamics. The recursive least-squares learning law can be shown to be stable in Appendix B along with the proof of the recursive least-squares learning law.

In practice, the recursive least-squares learning law can be used to estimate the plant dynamics either continuously or discretely at every n data samples. Continuous time estimation requires solving the differential equations (35) and (36) at each time step. On the other hand, the discrete-time sampling estimation provides more flexibility in that the estimation can be executed after a specified number of data points have been collected. This would ensure that the signals contained in the sampled data are sufficiently rich to enable an accurate convergence. In the implementation of the hybrid intelligent flight control algorithm, we use the following discrete-time recursive least-squares learning law with an adaptive or directional forgetting factor [15]:

$$\mathbf{\Phi}_{i+1} = \mathbf{\Phi}_i + (1+\xi_i)^{-1}\mathbf{R}_i\boldsymbol{\theta}_i\left(\hat{\boldsymbol{\varepsilon}}_{i+1}^{\top} - \boldsymbol{\theta}_i^{\top}\mathbf{\Phi}_i\right) \tag{38}$$

$$\mathbf{R}_{i+1} = \mathbf{R}_i - \left(\psi_{i+1}^{-1} + \xi_{i+1}\right)^{-1}\mathbf{R}_i\boldsymbol{\theta}_i\boldsymbol{\theta}_i^{\top}\mathbf{R}_i \tag{39}$$

where ψ is defined as

$$\psi_{i+1} = \varphi_{i+1} - \xi_i^{-1}(1-\varphi_{i+1}) \tag{40}$$

The value of the adaptive directional forgetting factor $\boldsymbol{\varphi}$ is then calculated at each sampling period as

$$\boldsymbol{\varphi}_{i+1}^{-1} = 1 + (1+\nu)\ln\left\{(1+\xi_{i+1}) + \left[\frac{\eta_{i+1}(1+\vartheta_{i+1})}{1+\xi_{i+1}+\eta_{i+1}} - 1\right]\frac{\xi_{i+1}}{1+\xi_{i+1}}\right\} \tag{41}$$

where ν is a constant; and η and ϑ are parameters with the following update laws

$$\eta_{i+1} = \lambda_{i+1}^{-1}\left\|\hat{\boldsymbol{\varepsilon}}_{i+1} - \mathbf{\Phi}_i^{\top}\boldsymbol{\theta}_i\right\|^2 \tag{42}$$

$$\vartheta_{i+1} = \varphi_{i+1}(1+\vartheta_i) \tag{43}$$

$$\lambda_{i+1} = \varphi_{i+1}\left[\lambda_i + (1+\xi_{i+1})\left\|\hat{\boldsymbol{\varepsilon}}_{i+1} - \mathbf{\Phi}_i^{\top}\boldsymbol{\theta}_i\right\|^2\right] \tag{44}$$

III. Simulations

To evaluate the hybrid intelligent flight control with the indirect adaptive and the recursive least-squares learning laws, simulations of a damaged twin-engine generic aircraft, as shown in Fig. 3, are performed in the MATLAB-environment [16]. The damage is simulated with a 25% loss of the left wing. In general, damage to the airframe can result in changes in aerodynamic chatacteristics of an aircraft that can cause a contraction in the original flight envelope. Adaptive control augments a flight control system to provide necessary aircraft stabilization to ease the pilot's workload. For safe flight, the role of guidance is also critical, given that the damaged aircraft may no longer be able to perform nomal flight maneuvers. The ability to generate feasible flight plans that assist the pilot in making a sound decision to navigate the aircraft to a safe landing is the ultimate goal of flight guidance [17].

Fig. 3 Generic damaged aircraft.

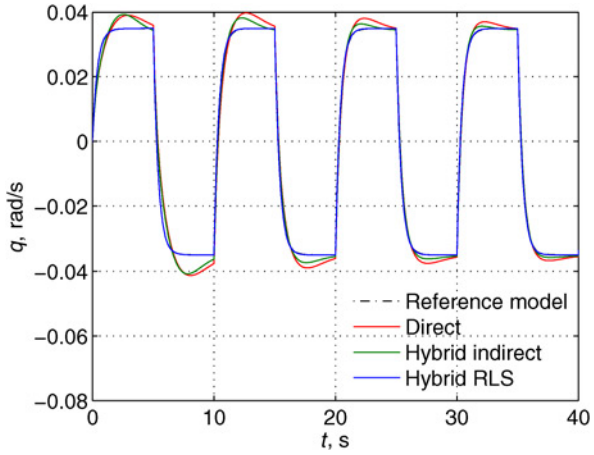

Fig. 4 Pitch doublet tracking performance.

For the simulation, a pilot pitch command is simulated with a series of step input pitch doublets. The tracking performance of the hybrid intelligent flight control is as shown in Fig. 4. It can be seen that the indirect adaptive learning law provides some degree of improvement in the tracking performance over the direct adaptive control. In contrast, the recursive least-squares learning law performs much better than both the direct and indirect adaptive learning laws as it provides a very accurate tracking performance of pitch channel command.

Figures 5 and 6 are the responses of the roll and yaw rates during the pitch doublet maneuver. Owing to asymmetric wing damage, the roll performance of the aircraft should be expected to be most affected. The hybrid recursive least-squares learning law is able to maintain both the roll and yaw rates near zero much better than both the direct and hybrid indirect adaptive learning laws. Figure 7 shows the norms of the tracking error in the roll, pitch, and yaw rates for the three adaptive control laws. The hybrid recursive least-squares learning law achieves the smallest tracking error norm, while the hybrid indirect adaptive learning law shows a significant reduction in the tracking error norm over the neural net direct adaptive control.

Figures 8 and 9 illustrate the convergence of the parameter estimation of the stability derivative $C_{\mathrm{m,q}}$ and the control derivative $C_{\mathrm{m},\delta_{\mathrm{e}}}$ for the recursive least-squares learning law. We note that the value of $\mathbf{R}$ controls the learning

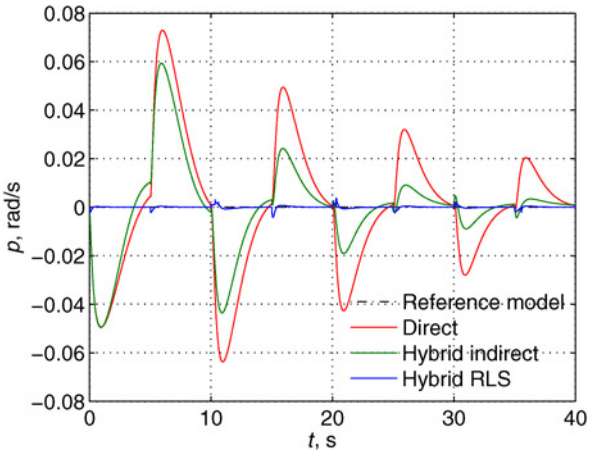

Fig. 5 Roll response.

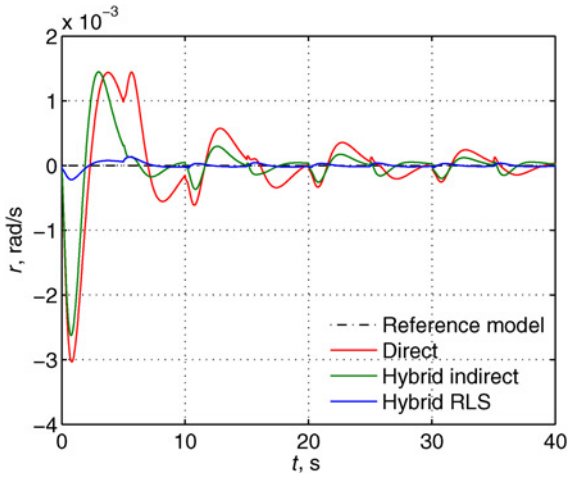

Fig. 6 Yaw response.

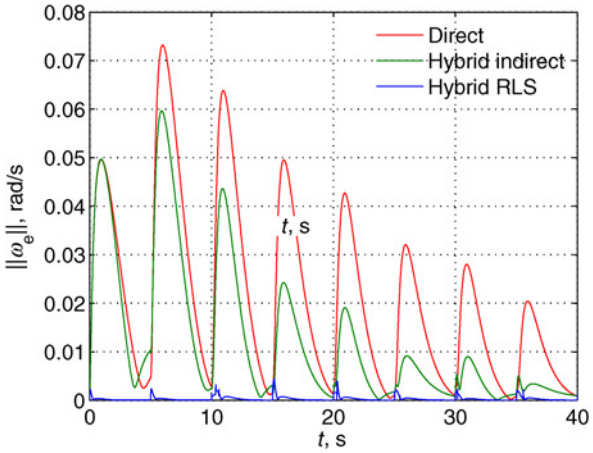

Fig. 7 Tracking error norm.

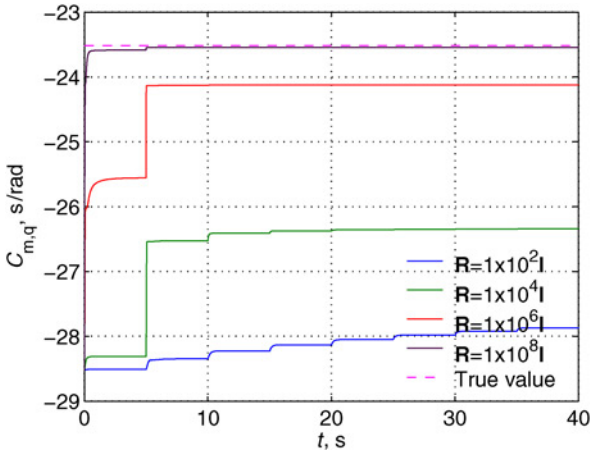

Fig. 8 RLS learning law convergence of stability derivative $C_{m,q}$.

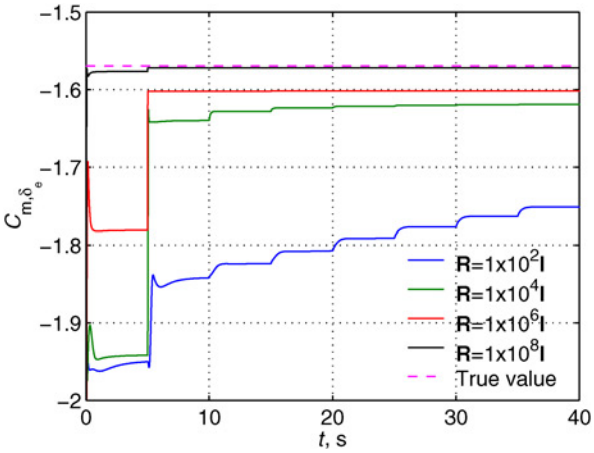

Fig. 9 RLS learning law convergence of control derivative C_{m,δ_e}.

rate of the recursive least-squares learning law. As **R** increases, the stability and control derivative estimates converge very rapidly to their correct values after only a very short time.

Figures 10 and 11 illustrate the learning characteristics of the indirect adaptive learning law. Comparing with Figs. 8 and 9, the parameter convergence of the indirect adaptive learning law is much poorer than the recursive least-squares learning law. As the learning rate matrix $\mathbf{\Lambda}$ increases further, the parameters do not seem to converge as with the recursive least-squares learning law. The poor parameter estimation of the indirect adaptive learning law can be explained by the fact that the learning law is established by the Lyapunov stability theory strictly to maintain stability of the tracking error without regards for minimizing the modeling or dynamic inversion error. The good performance of the recursive least-squares learning law is attributable to the optimal estimation process built into the learning law. Thus, while the indirect adaptive learning law can improve the command-tracking performance over the neural net direct adaptive control, the recursive least-squares learning law appears much more promising in its ability to perform parameter estimation with an accurate and rapid parameter convergence that can be used to refine the plant dynamics in the intelligent flight control. After updating the plant dynamics, a more accurate dynamic inversion controller can be computed. This will then result in a reduction in the modeling error, leading to an improved command-tracking performance.

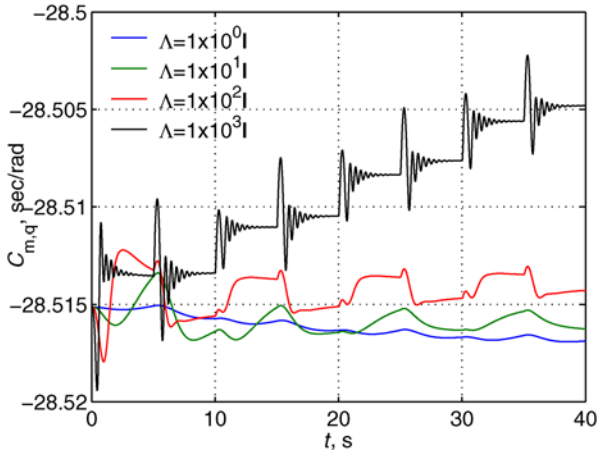

Fig. 10 Indirect adaptive learning law convergence of stability derivative $C_{m,q}$.

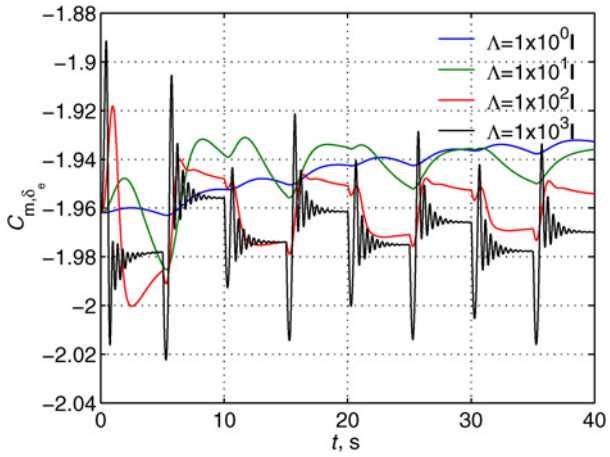

Fig. 11 Indirect adaptive learning law convergence of control derivative C_{m,δ_e}.

IV. Conclusions

This paper presents a recent development of a hybrid intelligent flight control method to provide an opportunity to perform parameter estimation of aircraft plant dynamics in conjunction with an existing neural net direct adaptation strategy. A potential benefit offered by this proposed method is that it can reduce the possibility of a high gain control in the current direct adaptive intelligent flight control. The online parameter estimation of the plant dynamics is provided by two adaptive learning laws: an indirect adaptive learning law based on the Lyapunov stability theory and a recursive least-squares learning law based on the optimal estimation method. A control simulation of a damaged generic aircraft shows that the hybrid intelligent flight control with the indirect adaptive learning law can provide an improvement in the tracking performance over the neural net direct adaptation strategy. The results further show that the recursive least-squares learning law provides an excellent parameter estimation with an acurate and rapid parameter convergence. In summary, the hybrid intelligent flight control with the recursive least-squares learning law appears quite promising as compared to that with the indirect adaptive learning law.

Appendix A

$\mathbf{A}$ is assumed to be Hurwitz. Let $\mathbf{W} = \mathbf{W}^* + \tilde{\mathbf{W}}$ and $\boldsymbol{\Phi} = \boldsymbol{\Phi}^* + \tilde{\boldsymbol{\Phi}}$ where the asterisk symbol denotes the ideal weight matrices that cancel out the residual error $\Delta\boldsymbol{\varepsilon}$ and the tilde symbol denotes the weight deviations. The ideal weight matrices are unknown but they may be assumed constant and bounded to stay within a $\boldsymbol{\Delta}$-neighborhood of the error $\boldsymbol{\varepsilon}$ such that

$$\boldsymbol{\Delta} = \sup_{\boldsymbol{\omega},\boldsymbol{\sigma},\boldsymbol{\delta}} \left| \mathbf{W}^{*\top}\boldsymbol{\beta} + \boldsymbol{\Phi}^{\top}\boldsymbol{\theta} - \Delta\boldsymbol{\varepsilon} \right| \tag{A1}$$

We define the following Lyapunov candidate function

$$V = \boldsymbol{e}^{\top}\mathbf{P}\boldsymbol{e} + \mathrm{tr}\left(\frac{\tilde{\mathbf{W}}^{\top}\tilde{\mathbf{W}}}{\Gamma} + \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{\Phi}} \right) \tag{A2}$$

where $\mathbf{P} \geqslant \mathbf{0}$ and tr(.) denotes the trace operation.

The time derivative of the Lyapunov candidate function is computed as

$$\dot{V} = \dot{\boldsymbol{e}}^{\top}\mathbf{P}\boldsymbol{e} + \boldsymbol{e}^{\top}\mathbf{P}\dot{\boldsymbol{e}} + 2\mathrm{tr}\left(\frac{\tilde{\mathbf{W}}^{\top}\dot{\tilde{\mathbf{W}}}}{\Gamma} + \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\Lambda}^{-1}\dot{\tilde{\boldsymbol{\Phi}}} \right) \tag{A3}$$

Defining $\mathbf{A}^{\top}\mathbf{P} + \mathbf{P}^{\top}\mathbf{A} = -\mathbf{Q}$, we obtain

$$\dot{V} \leqslant -\boldsymbol{e}^{\top}\mathbf{Q}\boldsymbol{e} + 2\boldsymbol{e}^{\top}\mathbf{P}\mathbf{B}\left(\tilde{\mathbf{W}}^{\top}\boldsymbol{\beta} + \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\theta} + \boldsymbol{\Delta} \right) + 2\mathrm{tr}\Big[-\tilde{\mathbf{W}}^{\top}\boldsymbol{\beta}\boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} - \mu\tilde{\mathbf{W}}^{\top}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\mathbf{W}^* + \tilde{\mathbf{W}} \right) + \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\Lambda}^{-1}\dot{\tilde{\boldsymbol{\Phi}}} \Big] \tag{A4}$$

Now adding and subtracting $2\mathrm{tr}\left(\tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\eta}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \boldsymbol{\Phi} \right)$ to and from the right-hand side yields

$$\begin{aligned} \dot{V} \leqslant & -\boldsymbol{e}^{\top}\mathbf{Q}\boldsymbol{e} + 2\boldsymbol{e}^{\top}\mathbf{P}\mathbf{B}\left(\tilde{\mathbf{W}}^{\top}\boldsymbol{\beta} + \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\theta} + \boldsymbol{\Delta} \right) + 2\mathrm{tr}\Big[-\tilde{\mathbf{W}}^{\top}\boldsymbol{\beta}\boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} - \mu\tilde{\mathbf{W}}^{\top}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\mathbf{W}^* + \tilde{\mathbf{W}} \right) \\ & - \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\eta}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\boldsymbol{\Phi}^* + \tilde{\boldsymbol{\Phi}} \right) + \tilde{\boldsymbol{\Phi}}^{\top}\left(\boldsymbol{\Lambda}^{-1}\dot{\tilde{\boldsymbol{\Phi}}} + \boldsymbol{\eta}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \boldsymbol{\Phi} \right) \Big] \end{aligned} \tag{A5}$$

Completing the square yields

$$\begin{aligned} 2\mathrm{tr}\Big[-\mu\tilde{\mathbf{W}}^{\top}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\mathbf{W}^* + \tilde{\mathbf{W}} \right) - \tilde{\boldsymbol{\Phi}}^{\top}\boldsymbol{\eta}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\boldsymbol{\Phi}^* + \tilde{\boldsymbol{\Phi}} \right) \Big] = & -2\mu\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\left\| \frac{\mathbf{W}^*}{2} + \tilde{\mathbf{W}} \right\|^2 - \left\| \frac{\mathbf{W}^*}{2} \right\|^2 \right) \\ & - 2\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \left(\left\| \boldsymbol{\eta}\left(\frac{\boldsymbol{\Phi}^*}{2} + \tilde{\boldsymbol{\Phi}} \right) \right\|^2 - \left\| \frac{\boldsymbol{\eta}\boldsymbol{\Phi}^*}{2} \right\|^2 \right) \end{aligned} \tag{A6}$$

Let $\lambda_{\min}$ and $\lambda_{\max}$ be the minimum and maximum eigenvalue operators on $\mathbf{Q}$ and $\mathbf{P}$. As $\|\mathbf{B}\| = 1$ and $\dot{\tilde{\boldsymbol{\Phi}}} = \dot{\boldsymbol{\Phi}}$, we establish that

$$\begin{aligned} \dot{V} \leqslant & -\lambda_{\min}(\mathbf{Q})\|\boldsymbol{e}\|^2 + \frac{\lambda_{\max}(\mathbf{P})\|\boldsymbol{e}\|}{2}\left(4\|\boldsymbol{\Delta}\| + \mu\left\| \mathbf{W}^* \right\|^2 + \left\| \boldsymbol{\eta}\boldsymbol{\Phi}^* \right\|^2 \right) \\ & - 2\lambda_{\min}(\mathbf{P})\|\boldsymbol{e}\| \left(\mu\left\| \frac{\mathbf{W}^*}{2} + \tilde{\mathbf{W}} \right\|^2 + \left\| \boldsymbol{\eta}\left(\frac{\boldsymbol{\Phi}^*}{2} + \tilde{\boldsymbol{\Phi}} \right) \right\|^2 \right) + 2\mathrm{tr}\Big[\tilde{\boldsymbol{\Phi}}^{\top}\left(\boldsymbol{\Lambda}^{-1}\dot{\boldsymbol{\Phi}} + \boldsymbol{\theta}\boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} + \boldsymbol{\eta}\left\| \boldsymbol{e}^{\top}\mathbf{P}\mathbf{B} \right\| \boldsymbol{\Phi} \right) \Big] \end{aligned} \tag{A7}$$

To guarantee that $\dot{V} \leqslant 0$, we require that the trace operator be equal to zero, thus resulting in the indirect adaptive learning law in Eq. (22). In addition, we also require that the error tracking norm is bounded from below by

$$\|\boldsymbol{e}\| > \frac{\lambda_{\max}(\mathbf{P})}{2\lambda_{\min}(\mathbf{Q})}\left(4\|\boldsymbol{\Delta}\| + \mu\left\| \mathbf{W}^* \right\|^2 + \left\| \boldsymbol{\eta}\boldsymbol{\Phi}^* \right\|^2 \right) \tag{A8}$$

The time rate of change of the Lyapunov candidate function is then strictly negative and therefore it would guarantee that the signals are bounded as

$$V(\infty) \leqslant V(0) - 2\lambda_{\min}(\mathbf{P}) \int_0^\infty \|\boldsymbol{e}\| \left(\mu \left\| \frac{\mathbf{W}^*}{2} + \tilde{\mathbf{W}} \right\|^2 + \left\| \boldsymbol{\eta} \left(\frac{\boldsymbol{\Phi}^*}{2} + \tilde{\boldsymbol{\Phi}} \right) \right\|^2 \right) \mathrm{d}t \tag{A9}$$

Thus, the value of V as $t \to \infty$ and the tracking error $\boldsymbol{e}$ are decreasing and uniformly bounded. Furthermore, if $\boldsymbol{\Delta} = \mathbf{0}$, $\mu = 0$, and $\boldsymbol{\eta} = \mathbf{0}$, we establish by means of the LaSalle–Yoshizawa theorem that $\lim_{t\to\infty} \|\boldsymbol{e}\| \to 0$ so that $\|\dot{\boldsymbol{\Phi}}\| \to 0$ as $t \to \infty$. This means that in theory the indirect adaptive learning law will result in a convergence of the estimated $\Delta\hat{\mathbf{F}}_1$, $\Delta\hat{\mathbf{F}}_2$, and $\Delta\hat{\mathbf{G}}$ to their steady-state values if there is no neural network approximation error. In addition, for the online estimation to converge to their correct values, the input signals must be sufficiently rich to excite all frequencies of interest in the plant dynamics. This condition is known as a persistent excitation (PE) and is defined as [13]

$$\boldsymbol{\alpha}_0 \leqslant \int_t^{t+T_0} \boldsymbol{\theta}(\tau)\, \boldsymbol{\theta}^\top(\tau)\, \mathrm{d}\tau \leqslant \boldsymbol{\alpha}_1 \tag{A10}$$

where $\boldsymbol{\alpha}_0,\ \boldsymbol{\alpha}_1,\ T_0 > 0$.

The effect of the e-modification parameter $\boldsymbol{\eta}$ is to increase the negative time rate of change of the Lyapunov candidate function so that as long as the effects of unmodeled dynamics and or disturbances do not exceed the value of $\dot{V}$, the adaptive signals should remain bounded.

Appendix B

The recursive least-squares learning law can be derived from the optimal estimation method by considering the following cost function

$$J(\boldsymbol{\Phi}) = \frac{1}{2} \int_0^t (1+\xi)^{-1} \left\| \hat{\boldsymbol{\varepsilon}} - \boldsymbol{\Phi}^\top \boldsymbol{\theta} \right\| \mathrm{d}\tau \tag{B1}$$

To minimize the cost function, we compute its gradient with respect to the weight matrix and set it to zero, thus resulting in

$$\frac{\partial J}{\partial \boldsymbol{\Phi}}^\top = -\int_0^t (1+\xi)^{-1} \boldsymbol{\theta} \left(\hat{\boldsymbol{\varepsilon}}^\top - \boldsymbol{\theta}^\top \boldsymbol{\Phi} \right) \mathrm{d}\tau = \mathbf{0} \tag{B2}$$

Equation (56) is then written as

$$\int_0^t (1+\xi)^{-1} \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}} \mathrm{d}\tau\, \boldsymbol{\Phi} = \int_0^t (1+\xi)^{-1} \boldsymbol{\theta}\hat{\boldsymbol{\varepsilon}}^{\mathrm{T}} \mathrm{d}\tau \tag{B3}$$

Let

$$\mathbf{R}^{-1} = \int_0^t (1+\xi)^{-1} \boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}} \mathrm{d}\tau > \mathbf{0} \tag{B4}$$

We note that

$$\mathbf{R}^{-1}\mathbf{R} = \mathbf{I} \Rightarrow \dot{\mathbf{R}}^{-1}\mathbf{R} + \mathbf{R}^{-1}\dot{\mathbf{R}} = \mathbf{0} \tag{B5}$$

Differentiating Eq. (57) yields

$$\mathbf{R}^{-1}\dot{\boldsymbol{\Phi}} + (1+\xi)^{-1} \boldsymbol{\theta}\boldsymbol{\theta}^\top \boldsymbol{\Phi} = (1+\xi)^{-1} \boldsymbol{\theta}\hat{\boldsymbol{\varepsilon}}^\top \tag{B6}$$

From Eqs. (59) and (60), we obtain the recursive least-squares learning law according to Eqs. (35) and (36).

The recursive least-squares learning law can be shown to be stable and result in bounded signals. To show this, we let $\mathbf{\Phi} = \mathbf{\Phi}^* + \tilde{\mathbf{\Phi}}$ with the asterisk and tilde symbols denoting ideal weights and weight deviations, respectively. We choose the following Lyapunov candidate function

$$L = V + \text{tr}\left(\tilde{\mathbf{\Phi}}^\top \mathbf{R}^{-1} \tilde{\mathbf{\Phi}}\right) \tag{B7}$$

where V is the Lyapunov candidate function for the neural net direct adaptive control and we have established that $\dot{V} \leqslant 0$. The time rate of change of the Lyapunov candidate function is computed as

$$\dot{L} = \dot{V} + \text{tr}\left(2\tilde{\mathbf{\Phi}}^\top \mathbf{R}^{-1} \dot{\tilde{\mathbf{\Phi}}} + \tilde{\mathbf{\Phi}}^\top \dot{\mathbf{R}}^{-1} \tilde{\mathbf{\Phi}}\right) \tag{B8}$$

The weight matrix $\mathbf{\Phi}$ can be shown to converge to the ideal weights $\mathbf{\Phi}^*$ [13] so that

$$\dot{\tilde{\mathbf{\Phi}}} = -(1+\xi)^{-1} \mathbf{R}\boldsymbol{\theta}\boldsymbol{\theta}^\top \tilde{\mathbf{\Phi}} \tag{B9}$$

Substituting Eq. (B9) into Eq. (B8) results in

$$\dot{L} = \dot{V} - (1+\xi)^{-1} \text{tr}\left(\tilde{\mathbf{\Phi}}^\top \boldsymbol{\theta}\boldsymbol{\theta}^\top \tilde{\mathbf{\Phi}}\right) \leqslant 0 \tag{B10}$$

As $\dot{L}$ is negative semi-definite, the recursive least-squares learning law is stable.

References

[1] Steinberg, M. L., 'A Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 1999, AIAA Paper 1999–4044.

[2] Rohrs, C. E., Valavani, L., Athans, M., and Stein, G., "Robustness of Continuous-Time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," *IEEE Transactions on Automatic Control*, Vol. 30, No. 9, 1985, pp. 881–889. doi: 10.1109/TAC.1985.1104070

[3] Eberhart, R. L., and Ward, D. G., "Indirect Adaptive Flight Control System Interactions," *International Journal of Robust and Nonlinear Control*, Vol. 9, No. 14, 1999, pp. 1013–1031.

[4] Rysdyk, R. T., and Calise, A. J., "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 1998, AIAA Paper 1998–4483.

[5] Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26–33.

[6] Johnson, E. N., Calise, A. J., El-Shirbiny, H. A., and Rysdyk, R. T., "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 2000, AIAA Paper 2000-4157.

[7] Narendra, K. S., and Annaswamy, A. M., "A New Adaptive Law for Robust Adaptation Without Persistent Excitation," *IEEE Transactions on Automatic Control*, Vol. 32, No. 2, 1987, pp. 134–145. doi: 10.1109/TAC.1987.1104543

[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, Reston, VA, 2003, AIAA Paper 2003-5790.

[9] Hovakimyan, N., Kim, N., Calise, A. J., Prasad, J. V. R., and Corban, E. J., "Adaptive Output Feedback for High-Bandwidth Control of an Unmanned Helicopter," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, 2001, AIAA Paper 2001-4181.

[10] Ahmed-Zaid, F., Ioannou, P., Gousman, K., and Rooney, R., "Accommodation of Failures in the F-16 Aircraft using Adaptive Control," *IEEE Control Systems Magazine*, Vol. 11, No. 1, 1991, pp. 73–78. doi: 10.1109/37.103360

[11] Boskovic, J. D., and Mehra, R. K., "A Multiple Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures," *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 4, 2002, pp. 712–724.

[12] Jacklin, S. A., Schumann, J. M., Gupta, P. P., Richard, R., Guenther, K., and Soares, F., "Development of Advanced Verification and Validation Procedures and Tools for the Certification of Learning Systems in Aerospace Applications," *AIAA Infotech@Aerospace Conference*, AIAA, Reston, VA, 2005, AIAA Paper 2005-6912.

[13] Ioannu, P. A., and Sun, J. *Robust Adaptive Control*, Prentice–Hall, Upper Saddle River, NJ 1996.

[14] Williams-Hayes, P. S., "Flight Test Implementation of a Second Generation Intelligent Flight Control System," Technical Report NASA TM-2005-213669, 2005.

[15] Bobal, V., *Digital Self-Tuning Controllers: Algorithms, Implementation, and Applications*, Springer-Verlag, Berlin/New york/Heidelberg, 2005.

[16] Bailey, R. M, Hostetler, R. W., Barnes, K. N., Belcastro, C. M., and Belcastro, C. M., "Experimental Validation: Subscale Aircraft Ground Facilities and Integrated Test Capability," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 2005, AIAA Paper 2005-6433.

[17] Atkins, E., "Dynamic Waypoint Generation Given Reduced Flight Performance," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reston, VA, 2004, AIAA Paper 2004-779.

Ella Atkins
Associate Editor