JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 3, May–June 2005

Adaptive Trajectory Control for Autonomous Helicopters

Eric N. Johnson* and Suresh K. Kannan†
Georgia Institute of Technology, Atlanta, Georgia 30332-0150

For autonomous helicopter flight, it is common to separate the flight control problem into an inner loop that controls attitude and an outer loop that controls the translational trajectory of the helicopter. In previous work, dynamic inversion and neural-network-based adaptation was used to increase performance of the attitude control system and the method of pseudocontrol hedging (PCH) was used to protect the adaptation process from actuator limits and dynamics. Adaptation to uncertainty in the attitude, as well as the translational dynamics, is introduced, thus, minimizing the effects of model error in all six degrees of freedom and leading to more accurate position tracking. The PCH method is used in a novel way that enables adaptation to occur in the outer loop without interacting with the attitude dynamics. A pole-placement approach is used that alleviates timescale separation requirements, allowing the outer-loop bandwidth to be closer to that of the inner loop, thus, increasing position tracking performance. A poor model of the attitude dynamics and a basic kinematics model is shown to be sufficient for accurate position tracking. The theory and implementation of such an approach, with a summary of flight-test results, are described.

Nomenclature

A, B	=	error dynamics system matrices
$\hat{A}_1$, $\hat{A}_2$, $\hat{B}$	=	estimate of vehicle dynamics as a linear system
$\boldsymbol{a}$	=	acceleration or activation potential
$\boldsymbol{a}(\cdot)$, $\hat{\boldsymbol{a}}(\cdot)$	=	translational dynamics and its estimate
b_v, b_w	=	neural network (NN) biases
$\boldsymbol{e}$	=	reference model tracking error
$\boldsymbol{g}(\cdot)$, $\hat{\boldsymbol{g}}(\cdot)$	=	actuator dynamics and its estimate
K, R	=	inner-loop, outer-loop gain matrices
n_1, n_2, n_3	=	number of NN inputs, hidden neurons, outputs
P, Q	=	Lyapunov equation matrices
$\boldsymbol{p}$	=	position, roll rate
$\boldsymbol{q}$	=	attitude quaternion, pitch rate
$\boldsymbol{r}$	=	filtered tracking error, yaw rate
V, W, Z	=	NN input, output, both weights
$\boldsymbol{v}$	=	velocity
$\boldsymbol{x}$	=	state vector
$\boldsymbol{x}_{\text{in}}$, $\bar{x}$	=	NN input
z_j	=	input to jth hidden neuron
α	=	angular acceleration
$\boldsymbol{\alpha}(\cdot)$, $\hat{\boldsymbol{\alpha}}(\cdot)$	=	attitude dynamics and its estimate
$\boldsymbol{\Delta}(\cdot)$	=	error the NN can approximate
$\bar{\boldsymbol{\Delta}}(\cdot)$	=	total function approximation error
$\boldsymbol{\delta}$, $\hat{\boldsymbol{\delta}}$	=	actuator deflections and its estimate
$\boldsymbol{\epsilon}_g(\cdot)$, $\bar{\epsilon}_g$	=	error NN cannot approximate and its bound
ζ	=	error dynamics damping
$\boldsymbol{\theta}_v$, $\boldsymbol{\theta}_w$	=	NN input-layer, output-layer thresholds
$\boldsymbol{\nu}_{\text{ad}}$	=	adaptive signal
$\boldsymbol{\nu}_r$	=	robustifying term
$\boldsymbol{\omega}$	=	angular velocity, error dynamics bandwidth

Subscripts

ad	=	adaptive
c	=	commanded
coll, ped	=	collective, pedal
cr	=	reference model dynamics
d	=	derivative
des	=	desired
f	=	force
h	=	hedge
i, o	=	inner loop, outer loop
lat, lon	=	lateral, longitudinal
m	=	moment
p	=	proportional
r	=	robustifying term, reference model

Introduction

UNMANNED helicopters are versatile machines that can perform aggressive maneuvers. This is evident from the wide range of acrobatic maneuvers executed by expert pilots. Helicopters have a distinct advantage over fixed-wing aircraft, especially in an urban environment, where hover capability is helpful. There is increased interest in the deployment of autonomous helicopters for military applications, especially in urban environments. These applications include reconnaissance, tracking of individuals or other objects of interest in a city, and search and rescue missions in urban areas. Autonomous helicopters must have the capability of planning routes and executing them. To be truly useful, these routes would include high-speed dashes, tight turns around buildings, avoiding dynamic obstacles and other required aggressive maneuvers. In planning[1] these routes, however, the tracking capability of the flight control system is a limiting factor because most current control systems still do not leverage the full flight envelope of small helicopters, at least, unless significant system identification and validation has been conducted.

Although stabilization and autonomous flight[2] has been achieved, the performance has generally been modest compared to a human pilot. This may be attributed to many factors, such as parametric uncertainty (changing mass and aerodynamic characteristics), unmodeled dynamics, actuator magnitude and rate saturation, and assumptions made during control design itself. Parametric uncertainty limits the operational envelope of the vehicle to where control designs are valid, whereas unmodeled dynamics and saturation can severely limit the achievable bandwidth of the system. The effect of uncertainty and unmodeled dynamics have been successfully handled using a combination of system identification[3–5] and robust

Presented as Paper 2002-4439 at the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, 5–8 August 2002; received 5 November 2003; revision received 15 April 2004; accepted for publication 27 April 2004.

*Lockheed Martin Assistant Professor of Avionics Integration, School of Aerospace Engineering, 270 Ferst Drive; eric.johnson@ae.gatech.edu. Member AIAA.

†Research Assistant, School of Aerospace Engineering, 270 Ferst Drive; suresh_kannan@ae.gatech.edu. Student Member AIAA.

control techniques.[6] Excellent flight and simulation results have been reported including acrobatic maneuvers[7] and modestly aggressive maneuvers.[6,8]

A key aspect in the effective use of unmanned air vehicles (UAVs) for military and civil applications is their ability to accommodate changing dynamics and payload configurations automatically without having to rely on substantial system identification efforts. Neural-network (NN)-based direct adaptive control has recently emerged as an enabling technology for practical flight control systems that allows online adaptation to uncertainty. This technology has been successfully applied to the recent U.S. Air Force reconfigurable control for tailless fighter aircraft (RESTORE), culminating in a successful flight demonstration[9,10] of the adaptive controller on the X-36. A combined inner–outer loop architecture was also applied for guidance and control of the X-33 (Ref. 11) and evaluated successfully in simulation for various failure cases.

This paper is concerned with the development of an adaptive controller for an autonomous helicopter using a neural network as the adaptive element. For autonomous helicopters, a primary objective is the accurate tracking of position commands. Much adaptive control work on helicopters has concentrated on improving the tracking performance of attitude commands.[12–14] Usually a simple outer loop employing basic relationships between attitude and linear acceleration is then used to control the translational dynamics. For many applications, this may be sufficient. However, when operating in an urban environment or flying in formation with other UAVs, the position tracking ability of the controller dictates the minimum proximity between the UAV and objects in its environment. In contrast to previous attitude control-only work, we introduce a coupled inner–outer loop adaptive design that can handle uncertainty in all six degrees of freedom. In synthesizing a controller (Fig. 1), the conventional conceptual separation between the inner loop and outer loop is made. The inner loop controls the moments acting on the aircraft by changing the lateral stick, δ_{lat}, longitudinal stick, δ_{lon}, and pedal, δ_{ped}, inputs. The outer loop controls the forces acting on the aircraft by varying the magnitude of the rotor thrust using the collective δ_{coll} input. The thrust vector is effectively oriented in the desired direction by commanding changes to the attitude of the helicopter using the inner loop.

The attitude and translational dynamics are input-to-state feedback linearized separately using dynamic inversion and linear controllers designed for the linearized dynamics. The effect of nonlinear parametric uncertainty arising due to approximate inversion is minimized using an adaptive element. A nonlinearly parameterized NN will be used to provide online adaptation. The design is such that actuator saturation limits are not avoided or prevented. When an adaptive element is introduced, a new problem arises by way of unwanted adaptation to plant input characteristics such as actuator saturation and dynamics. For example, the inner-loop attitude control sees actuator limits, rate saturation, and associated dynamics. To alleviate this problem, pseudocontrol hedging[11,15] (PCH) is used to modify the inner-loop reference model dynamics in a way that allows continued adaptation in the presence of these system characteristics. This same technique, PCH, is used to prevent adaptation to inner-loop dynamics and interaction between the inner and outer loops. Without hedging of the outer loop, adaptation to uncertainty in the translational dynamics would not be possible. A common assumption when designing control systems for air vehicles is the timescale separation[16] between the inner-loop attitude control and outer-loop trajectory control systems. The assumption allows the inner loop and outer loop to be designed separately but requires the outer-loop bandwidth to be much lower than that of the inner loop. This problem is alleviated by using a combination of PCH and gain selection by a combined analysis of the two loops. This allows the outer-loop bandwidth to be closer to that of the inner loop, thus, increasing position tracking performance. Additionally, to the authors knowledge, the flight results presented in this paper are the first where adaptation is used to compensate for modeling errors in all six degrees of freedom.

We first develop the adaptive controller architecture for a generic six-degree-of-freedom air vehicle, followed by a description of the NN and selection of linear compensator gains. The controller is then applied to the trajectory and attitude control of an unmanned helicopter. Practical discussions on the choice of parameters and reference model dynamics are provided. Finally, flight test results are presented.

Controller Development

Consider an air vehicle modeled as a nonlinear system of the form

$$\dot{\boldsymbol{p}} = \boldsymbol{v} \tag{1}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{a}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_f, \boldsymbol{\delta}_m) \tag{2}$$

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{\omega}) \tag{3}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_f, \boldsymbol{\delta}_m) \tag{4}$$

where $\boldsymbol{p} \in \mathcal{R}^3$ is the position vector, $v \in \mathcal{R}^3$ is the velocity of the vehicle, $q \in \mathcal{R}^4$ is the attitude quaternion, and $\omega \in \mathcal{R}^3$ is the angular velocity. Equation (2) represents translational dynamics and Eq. (4) represents the attitude dynamics. Equation (3) represents the quaternion propagation equations.[17] The use of quaternions, though not a minimal representation of attitude, avoids numerical and singularity problems that Euler angles based representations have. This enables the control system to be all-attitude capable as required for aggressive maneuvering. The state vector $\boldsymbol{x}$ may now be defined as

$$\boldsymbol{x} \triangleq [\boldsymbol{p}^T \quad \boldsymbol{v}^T \quad \boldsymbol{q}^T \quad \boldsymbol{\omega}^T]^T \tag{5}$$

The control vectors are denoted by $\boldsymbol{\delta}_f$ and $\boldsymbol{\delta}_m$ and represent actual physical actuators on the aircraft, where $\boldsymbol{\delta}_f$ denotes the primary force generating actuators and $\boldsymbol{\delta}_m$ denotes the primary

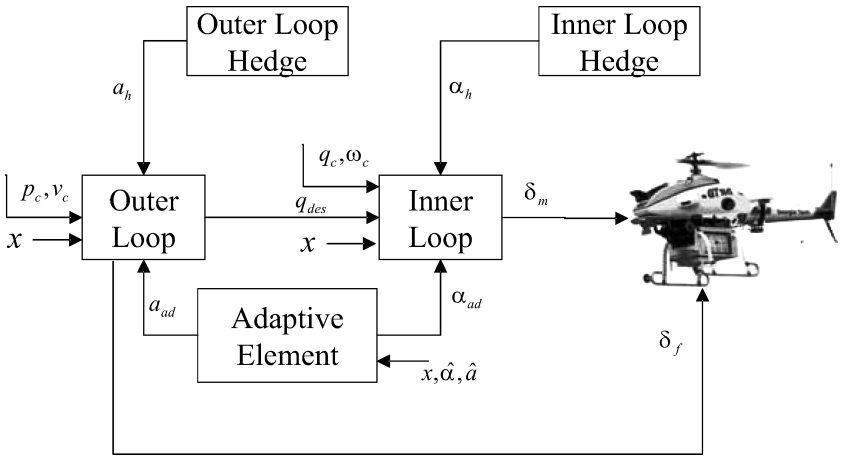

Fig. 1 Overall inner and outer loop with adaptation and hedging.

moment generating actuators. For a helicopter, the main force effector is the rotor thrust, which is controlled by changing main rotor collective δ_{coll}. Hence, $\boldsymbol{\delta}_f \in \mathcal{R}^1 = \delta_{\text{coll}}$. There are three primary moment control surfaces, the lateral cyclic δ_{lat}, longitudinal cyclic δ_{lon}, and tail rotor pitch, also called the pedal input δ_{ped}. Hence, $\boldsymbol{\delta}_m \in \mathcal{R}^3 = [\delta_{\text{lat}} \quad \delta_{\text{lon}} \quad \delta_{\text{ped}}]^T$. This classification of the controls as moment and force generating is an artifact of the inner-loop–outer-loop control design strategy. In general, both control inputs, $\boldsymbol{\delta}_f$ and $\boldsymbol{\delta}_m$, may each produce forces and moments. The helicopter is an underactuated system, and hence, the aircraft's attitude $\boldsymbol{q}$ is treated like a virtual actuator used to tilt the main rotor thrust to produce desired accelerations. When the consolidated control vector $\boldsymbol{\delta}$ is defined as

$$\boldsymbol{\delta} \triangleq \begin{bmatrix} \boldsymbol{\delta}_f^T & \boldsymbol{\delta}_m^T \end{bmatrix}^T \tag{6}$$

the actuators themselves may have dynamics represented by

$$\dot{\boldsymbol{\delta}} = \begin{bmatrix} \dot{\boldsymbol{\delta}}_m \\ \dot{\boldsymbol{\delta}}_f \end{bmatrix} = \begin{bmatrix} g_m\left(\boldsymbol{x}, \boldsymbol{\delta}_m, \boldsymbol{\delta}_{m_{\text{des}}}\right) \\ \boldsymbol{g}_f\left(\boldsymbol{x}, \boldsymbol{\delta}_f, \boldsymbol{\delta}_{f_{\text{des}}}\right) \end{bmatrix} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{\delta}_{\text{des}}) \tag{7}$$

where $g(\cdot)$ is assumed to be asymptotically stable but perhaps unknown.

When any actuator dynamics and nonlinearities are ignored, approximate feedback linearization of the system represented by Eqs. (1–4) is achieved by introducing the following transformation:

$$\begin{bmatrix} \boldsymbol{a}_{\text{des}} \\ \boldsymbol{\alpha}_{\text{des}} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{a}}\left(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \boldsymbol{q}_{\text{des}}, \boldsymbol{\delta}_{f_{\text{des}}}, \hat{\boldsymbol{\delta}}_m\right) \\ \hat{\boldsymbol{\alpha}}\left(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{m_{\text{des}}}\right) \end{bmatrix} \tag{8}$$

where, $\boldsymbol{a}_{\text{des}}$ and $\boldsymbol{\alpha}_{\text{des}}$ are commonly referred to as the pseudocontrol and represent desired accelerations. Here $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{\alpha}}$ represent an available approximation of $\boldsymbol{a}(\cdot)$ and $\boldsymbol{\alpha}(\cdot)$. Additionally, $\boldsymbol{\delta}_{f_{\text{des}}}$, $\boldsymbol{\delta}_{m_{\text{des}}}$, and $\boldsymbol{q}_{\text{des}}$ are the control inputs and attitude that are predicted to achieve the desired pseudocontrol. This form assumes that translational dynamics are coupled strongly with attitude dynamics, as is the case for a helicopter. From the outer-loop's point of view, $\boldsymbol{q}$ (attitude) is like an actuator that generates translational accelerations, and $\boldsymbol{q}_{\text{des}}$ is the desired attitude that the outer-loop inversion expects will contribute toward achieving the desired translational acceleration $\boldsymbol{a}_{\text{des}}$. The dynamics of $\boldsymbol{q}$ appears like actuator dynamics to the outer loop. The attitude quaternion $\boldsymbol{q}_{\text{des}}$ will be used to augment the externally commanded attitude $\boldsymbol{q}_c$ to achieve the desired translational accelerations. Because actuator positions are often not measured on small helicopters, estimates of the actuator positions $\hat{\boldsymbol{\delta}}_m$ and $\hat{\boldsymbol{\delta}}_f$ are used. For cases where the actuator positions are directly measured, they may be regarded as known $\hat{\boldsymbol{\delta}}_m = \boldsymbol{\delta}_m$ and $\hat{\boldsymbol{\delta}}_f = \boldsymbol{\delta}_f$. In fact, in the outer loop's case, the attitude $\boldsymbol{q}$ is measured using inertial sensors. When $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{\alpha}}$ are chosen such that they are invertible, the desired control and attitude may be written as

$$\begin{bmatrix} \boldsymbol{\delta}_{f_{\text{des}}} \\ \boldsymbol{q}_{\text{des}} \end{bmatrix} = \hat{\boldsymbol{a}}^{-1}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \boldsymbol{a}_{\text{des}}, \hat{\boldsymbol{\delta}}_m)$$

$$\boldsymbol{\delta}_{m_{\text{des}}} = \hat{\alpha}^{-1}(\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\alpha}_{\text{des}}) \tag{9}$$

where actuator estimates are given by actuator models

$$\dot{\hat{\boldsymbol{\delta}}} = \begin{bmatrix} \dot{\hat{\boldsymbol{\delta}}}_f \\ \dot{\hat{\boldsymbol{\delta}}}_m \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{g}}_f\left(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{f_{\text{des}}}\right) \\ \hat{\boldsymbol{g}}_m\left(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_m, \boldsymbol{\delta}_{m_{\text{des}}}\right) \end{bmatrix} = \hat{\boldsymbol{g}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}, \boldsymbol{\delta}_{\text{des}}) \tag{10}$$

In later sections, it will be shown that $\hat{\boldsymbol{\alpha}}$, can just be an approximate linear model of vehicle attitude dynamics and $\hat{\boldsymbol{a}}$ a set of simple equations relating translational accelerations to the attitude of the vehicle. Introducing the inverse control law Eq. (9) into Eq. (2) and Eq. (4) results in the following closed-loop translational and attitude dynamics:

$$\dot{\boldsymbol{v}} = \boldsymbol{a}_{\text{des}} + \bar{\Delta}_a(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) - \boldsymbol{a}_h, \qquad \dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}_{\text{des}} + \bar{\boldsymbol{\Delta}}_\alpha(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) - \boldsymbol{\alpha}_h \tag{11}$$

where

$$\bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) = \begin{bmatrix} \bar{\boldsymbol{\Delta}}_a(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) \\ \bar{\boldsymbol{\Delta}}_\alpha(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{\delta}) - \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) \\ \boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{\delta}) - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) \end{bmatrix} \tag{12}$$

are static nonlinear functions (model error) that arise due to imperfect model inversion and errors in the actuator model $\hat{\boldsymbol{g}}$. The signals $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$ represent the pseudocontrol that cannot be achieved due to actuator input characteristics such as saturation. If the model inversion were perfect and no saturation were to occur, $\bar{\boldsymbol{\Delta}}$, $\boldsymbol{a}_h$, and $\boldsymbol{\alpha}_h$ would vanish leaving only the pseudocontrols $\boldsymbol{a}_{\text{des}}$ and $\boldsymbol{\alpha}_{\text{des}}$. One may address model error and stabilize the linearized system by designing the pseudocontrols as

$$\boldsymbol{a}_{\text{des}} = \boldsymbol{a}_{\text{cr}} + \boldsymbol{a}_{\text{pd}} - \bar{\boldsymbol{a}}_{\text{ad}}, \qquad \boldsymbol{\alpha}_{\text{des}} = \boldsymbol{\alpha}_{\text{cr}} + \boldsymbol{\alpha}_{\text{pd}} - \bar{\boldsymbol{\alpha}}_{\text{ad}} \tag{13}$$

where a_{cr} and α_{cr} are outputs of reference models for the translational and attitude dynamics, respectively; $\boldsymbol{a}_{\text{pd}}$ and $\boldsymbol{\alpha}_{\text{pd}}$ are outputs of proportional–derivative (PD) compensators; and finally, $\bar{\boldsymbol{a}}_{\text{ad}}$ and $\bar{\boldsymbol{\alpha}}_{\text{ad}}$ are the outputs of an adaptive element (an NN) designed to cancel model error $\bar{\boldsymbol{\Delta}}$. The effects of input dynamics, represented by $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$ will first be addressed in the following section by designing the reference model dynamics such that they do not appear in the tracking error dynamics. The reference model, tracking error dynamics, and adaptive element are discussed in the following sections.

Reference Model and PCH

Normally, the reference model dynamics are of the form

$$\dot{\boldsymbol{p}}_r = \boldsymbol{v}_r, \qquad \dot{\boldsymbol{v}}_r = \boldsymbol{a}_{\text{cr}}(\boldsymbol{p}_r, \boldsymbol{v}_r, \boldsymbol{p}_c, \boldsymbol{v}_c)$$

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}(\boldsymbol{q}_r, \boldsymbol{\omega}_r), \qquad \dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{\text{cr}}(\boldsymbol{q}_r, \boldsymbol{\omega}_r, \boldsymbol{q}_c \oplus \boldsymbol{q}_{\text{des}}, \boldsymbol{\omega}_c) \tag{14}$$

where $\boldsymbol{p}_r$ and $\boldsymbol{v}_r$ are the outer-loop reference model states and q_r and ω_r are the inner-loop reference model states. The external command signal is $\boldsymbol{x}_c = [\boldsymbol{p}_c^T \quad \boldsymbol{v}_c^T \quad \boldsymbol{q}_c^T \quad \boldsymbol{\omega}_c^T]^T$. Note that the attitude desired by the outer loop is now added to the commands for the inner-loop controller. Here, $\boldsymbol{q}_c \oplus \boldsymbol{q}_{\text{des}}$ denotes quaternion addition.[17]

Any dynamics and nonlinearities associated with the actuators $\boldsymbol{\delta}_m$ and $\boldsymbol{\delta}_f$ have not yet been considered in the design. If they become saturated (position or rate), the reference models will continue to demand tracking as though full authority were still available. Furthermore, the inner loop appears like an actuator with dynamics to the outer loop. Practical operational limits on the maximum attitude of the aircraft may have also been imposed in the inner-loop reference model. This implies that the outer-loop desired attitude augmentation $\boldsymbol{q}_{\text{des}}$ may not actually be achievable, or at the very least is subject to the inner-loop dynamics. When an adaptive element such as an NN is introduced, these input dynamics and nonlinearities will appear in the tracking error dynamics resulting in the adaptive element attempting to correct for them and is undesirable.

PCH may be used to prevent the adaptive element from attempting to adapt to selected system input characteristics. One way to describe the PCH method is as follows: Move the reference models in the opposite direction (hedge) by an estimate of the amount the plant did not move due to system characteristics the control designer does not want the adaptive element to see.[15] This will prevent the characteristic from appearing in the model tracking error dynamics to be developed in the sequel. The reference model dynamics may be redesigned to include hedging as follows:

$$\dot{\boldsymbol{v}}_r = \boldsymbol{a}_{\text{cr}} - \boldsymbol{a}_h, \qquad \dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{\text{cr}} - \boldsymbol{\alpha}_h \tag{15}$$

$$\boldsymbol{a}_{\text{cr}} = \boldsymbol{a}_{\text{cr}}(\boldsymbol{p}_r, \boldsymbol{v}_r, \boldsymbol{p}_c, \boldsymbol{v}_c), \qquad \boldsymbol{\alpha}_{\text{cr}} = \boldsymbol{\alpha}_{\text{cr}}(\boldsymbol{q}_r, \boldsymbol{\omega}_r, \boldsymbol{q}_c \oplus \boldsymbol{q}_{\text{des}}, \boldsymbol{\omega}_c) \tag{16}$$

where $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$ are the difference between commanded pseudocontrol and achieved pseudocontrol,

$$\boldsymbol{a}_h = \hat{\boldsymbol{a}}\left(\boldsymbol{x}, \boldsymbol{q}_{\text{des}}, \boldsymbol{\delta}_{f_{\text{des}}}, \hat{\boldsymbol{\delta}}_m\right) - \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m) = \boldsymbol{a}_{\text{des}} - \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m) \tag{17}$$

$$\boldsymbol{\alpha}_h = \hat{\boldsymbol{\alpha}}\left(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{m_{\text{des}}}\right) - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m) = \boldsymbol{\alpha}_{\text{des}} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m) \tag{18}$$

Note that the hedge signals $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$, do not directly affect the reference model output $\boldsymbol{a}_{\rm cr}$ and $\boldsymbol{\alpha}_{\rm cr}$, but do so only through subsequent changes in the reference model states.

Remark 1. Choosing the reference model dynamics $\boldsymbol{a}_{\rm cr}$ and $\boldsymbol{\alpha}_{\rm cr}$ is important in determining the effect of actuator saturation on the system dynamics.[18,19]

Tracking Error Dynamics

One may define the tracking error vector $\boldsymbol{e}$ as

$$\boldsymbol{e} \triangleq \begin{bmatrix} \boldsymbol{p}_r - \boldsymbol{p} \\ \boldsymbol{v}_r - \boldsymbol{v} \\ \tilde{\boldsymbol{Q}}(\boldsymbol{q}_r, \boldsymbol{q}) \\ \boldsymbol{\omega}_r - \boldsymbol{\omega} \end{bmatrix} \tag{19}$$

where $\tilde{\boldsymbol{Q}} : \mathcal{R}^4 \times \mathcal{R}^4 \mapsto \mathcal{R}^4$ is a function[15] that, given two quaternions, results in an error angle vector with three components. An expression for $\tilde{\boldsymbol{Q}}$ is given by

$$\tilde{\boldsymbol{Q}}(\boldsymbol{p}, \boldsymbol{q}) = 2\,\mathrm{sgn}(q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4) \times \begin{bmatrix} -q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ -q_1 p_3 - q_2 p_4 + q_3 p_1 + q_4 p_2 \\ -q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1 \end{bmatrix} \tag{20}$$

The output of the PD compensators may be written as

$$\begin{bmatrix} \boldsymbol{a}_{\rm pd} \\ \boldsymbol{\alpha}_{\rm pd} \end{bmatrix} = \begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} \boldsymbol{e} \tag{21}$$

where $R_p, R_d \in \mathcal{R}^{3\times 3}$ and $K_p, K_d \in \mathcal{R}^{3\times 3}$ are linear gain positive definite matrices whose choice is discussed subsequently. The tracking error dynamics may be found by directly differentiating Eq. (19),

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \boldsymbol{v}_r - \boldsymbol{v} \\ \dot{\boldsymbol{v}}_r - \dot{\boldsymbol{v}} \\ \boldsymbol{\omega}_r - \boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}}_r - \dot{\boldsymbol{\omega}} \end{bmatrix} \tag{22}$$

When $\dot{\boldsymbol{e}}_2$ is considered,

$$\begin{aligned} \dot{\boldsymbol{e}}_2 &= \dot{\boldsymbol{v}}_r - \dot{\boldsymbol{v}} \\ &= \boldsymbol{a}_{\rm cr} - \boldsymbol{a}_h - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{\delta}) \\ &= \boldsymbol{a}_{\rm cr} - \boldsymbol{a}_{\rm des} + \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{\delta}) \\ &= \boldsymbol{a}_{\rm cr} - \boldsymbol{a}_{\rm pd} - \boldsymbol{a}_{\rm cr} + \bar{\boldsymbol{a}}_{\rm ad} + \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{\delta}) \\ &= -\boldsymbol{a}_{\rm pd} - [\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{\delta}) - \hat{\boldsymbol{a}}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \bar{\boldsymbol{a}}_{\rm ad}] \\ &= -\boldsymbol{a}_{\rm pd} - [\bar{\boldsymbol{\Delta}}_a(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) - \bar{\boldsymbol{a}}_{\rm ad}] \end{aligned} \tag{23}$$

$\dot{\boldsymbol{e}}_4$ may be found similarly. Then, the overall tracking error dynamics may now be expressed as

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}[\bar{\boldsymbol{\nu}}_{\rm ad} - \bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}})] \tag{24}$$

where $\bar{\boldsymbol{\Delta}}$ is given by Eq. (12),

$$\bar{\boldsymbol{\nu}}_{\rm ad} = \begin{bmatrix} \bar{\boldsymbol{a}}_{\rm ad} \\ \bar{\boldsymbol{\alpha}}_{\rm ad} \end{bmatrix} \tag{25}$$

$$A = \begin{bmatrix} 0 & I & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \tag{26}$$

and so the linear gain matrices must be chosen such that A is Hurwitz. Now, $\bar{\boldsymbol{\nu}}_{\rm ad}$ remains to be designed to cancel the effect of $\bar{\boldsymbol{\Delta}}$.

Remark 2. 1) Note that commands $\boldsymbol{\delta}_{m_{\rm des}}$, $\boldsymbol{\delta}_{f_{\rm des}}$, and $\boldsymbol{q}_{\rm des}$, do not appear in the tracking error dynamics. PCH allows adaptation to continue when the actual control signal has been replaced by any arbitrary signal. 2) If the actuator is considered ideal and the actual position and the commanded position are equal, addition of the PCH signal $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$ has no effect on any system signal.

Effect of Actuator Model on Error Dynamics

An important aspect of the PCH signal calculation given by Eqs. (17) and (18) is estimation of the actual actuator position at the current instant. The assumptions and model used to estimate the actuator positions in calculating the hedging signals play a role in what appears in the tracking error dynamics.

Actuator Positions are Measured

The simplest case arises when $\boldsymbol{\delta}$ is measured and available for feedback. In this case, models for the actuators are not needed. In fact, when all actuator signals are known, then $\bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}, \bar{\boldsymbol{\delta}}) = \bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}) = \boldsymbol{\Delta}(\boldsymbol{x}, \boldsymbol{\delta})$ and the tracking error dynamics of Eq. (24) is given by

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}[\bar{\boldsymbol{\nu}}_{\rm ad}(\boldsymbol{x}, \boldsymbol{\delta}) - \boldsymbol{\Delta}(\boldsymbol{x}, \boldsymbol{\delta})] \tag{27}$$

Note that with regard to the outer loop, the inner loop acts like an actuator with dynamics, at least with respect to achieving the desired attitude $q_{\rm des}$. The actual attitude quaternion q is available and appears as a part of the state measurement. Hence, it is always available as an input to the adaptive element, as well as in the calculation of the hedge signal.

Actuator Position is a Static Function of the Model and Plant States

If it can be assumed that actuator deflections have the form $\boldsymbol{\delta} = \boldsymbol{\delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$, for example, saturation occurs earlier than in the model of the actuator, the discrepancy appears as model error, which the NN can correct for. Thus, $\bar{\boldsymbol{\Delta}}[\boldsymbol{x}, \boldsymbol{\delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}), \hat{\boldsymbol{\delta}}] = \boldsymbol{\Delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ and the error dynamics take the form

$$\dot{\boldsymbol{e}} = A\boldsymbol{e} + B[\bar{\nu}_{\rm ad}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \Delta(\boldsymbol{x}, \hat{\boldsymbol{\delta}})] \tag{28}$$

Actuator Model has Error the NN Cannot Compensate

If actuator positions are not measured and an assumption such as $\boldsymbol{\delta} = \boldsymbol{\delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ cannot be made, the uncertainty $\bar{\Delta}$ may be expressed as

$$\bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) = \boldsymbol{\Delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) + \boldsymbol{\epsilon}_g(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) \tag{29}$$

where $\boldsymbol{\Delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ is model error the NN can approximate and $\boldsymbol{\epsilon}_g$ is the model error the NN cannot cancel when $\boldsymbol{\delta}$ is not available as an input to the network and has components independent of $\boldsymbol{x}$ and $\hat{\boldsymbol{\delta}}$. Errors in the actuator model that the NN can cancel include bias error in the actuator position estimate and erroneous values for when magnitude saturation occurs. Model errors that appear in $\boldsymbol{\epsilon}_g$ that the NN cannot cancel include parameters that affect the dynamics of the actuator such as actuator time constants and rate limits. The tracking error dynamics may now be expressed as

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{B}[\bar{\boldsymbol{\nu}}_{\rm ad}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \boldsymbol{\Delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}}) - \boldsymbol{\epsilon}_g(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}})] \tag{30}$$

where $\bar{\boldsymbol{\nu}}_{\rm ad}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ is designed to cancel $\boldsymbol{\Delta}(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ and $\boldsymbol{\epsilon}_g$ appears as unmodeled input dynamics to the control system.

Actuator Model is Conservative

One way to predict actuator position accurately is to impose conservative artificial limits on the desired actuator deflections, perhaps in software, and make the assumption that the real actuator tracks the conservative commands accurately. This amounts to always knowing $\boldsymbol{\delta}$, and the error dynamics take the form given by Eq. (27).

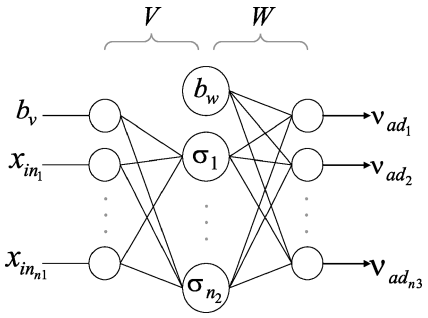

Fig. 2 NN with one hidden layer.

Thus far, the various components of Eq. (13) have been designed and PCH has been used to prevent any input dynamics from appearing in the error dynamics of Eq. (24). The only component yet to be designed is the adaptive element $\bar{\boldsymbol{\nu}}_{\text{ad}}(.)$ to approximate $\boldsymbol{\Delta}(.)$ and minimize the forcing term on the right-hand side of the error dynamics equations. In this paper, a single hidden layer NN is used, its structure and approximation capabilities are discussed in the following section.

Adaptive Element

Single hidden layer (SHL) perceptron NNs are universal approximators.[20–22] Hence, given a sufficient number of hidden layer neurons and appropriate inputs, it is possible to train the network online to cancel model error. Figure 2 shows the structure of a generic SHL network whose input–output map may be expressed as

$$\nu_{\text{ad}_k} = b_w\theta_{w_k} + \sum_{j=1}^{n_2} w_{jk}\sigma_j(z_j) \tag{31}$$

where $k = 1, \ldots, n_3$, b_w is the outer layer bias, θ_{w_k} is the kth threshold, w_{jk} is the outer-layer weights, and the scalar σ_j is a sigmoidal activation function,

$$\sigma_j(z_j) = 1/(1 + e^{-az_j}) \tag{32}$$

where a is the so called activation potential and may have a distinct value for each neuron. Here z_j is the input to the jth hidden layer neuron and is given by

$$z_j = b_v\theta_{v_j} + \sum_{i=1}^{n_1} v_{ij}x_{\text{in}_i} \tag{33}$$

where b_v is the inner-layer bias and θ_{v_j} is the jth threshold. Here, n_1, n_2, and n_3 are the number of inputs, hidden layer neurons, and outputs, respectively, and $x_{\text{in}_i}, i = 1, \ldots, n_1$, denotes the inputs to the NN. For convenience, define the following weight matrices:

$$V \triangleq \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \tag{34}$$

$$W \triangleq \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \tag{35}$$

$$Z \triangleq \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{36}$$

Additionally, define the $\boldsymbol{\sigma}(z)$ vector as

$$\boldsymbol{\sigma}^T(\boldsymbol{z}) \triangleq \begin{bmatrix} b_w & \sigma(z_1) & \cdots & \sigma\left(z_{n_2}\right) \end{bmatrix} \tag{37}$$

where $b_w > 0$ allows for the thresholds θ_w to be included in the weight matrix W. Also, $\boldsymbol{z} = V^T\bar{\boldsymbol{x}}$, where

$$\bar{\boldsymbol{x}}^T = \begin{bmatrix} b_v & \boldsymbol{x}_{\text{in}}^T \end{bmatrix} \tag{38}$$

where $b_v > 0$ is an input bias that allows for thresholds θ_v to be included in the weight matrix V. The input–output map of the SHL network may now be written in concise form as

$$\boldsymbol{\nu}_{\text{ad}} = W^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}) \tag{39}$$

The NN may be used to approximate a nonlinear function, such as $\boldsymbol{\Delta}(.)$. The universal approximation property[20] of NNs ensures that given an $\bar{\epsilon} > 0$, then $\forall\bar{\boldsymbol{x}} \in \mathcal{D}$, where $\mathcal{D}$ is a compact set, $\exists$ an $\bar{n}_2$ and an ideal set of weights (V^*, W^*) that brings the output of the NN to within an ϵ neighbourhood of the function approximation error. This ϵ is bounded by $\bar{\epsilon}$, which is defined by

$$\bar{\epsilon} = \sup_{\bar{x}\in\mathcal{D}} \|W^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}) - \boldsymbol{\Delta}(\bar{\boldsymbol{x}})\| \tag{40}$$

The weights (V^*, W^*) may be viewed as optimal values of (V, W) in the sense that they minimize $\bar{\epsilon}$ on $\mathcal{D}$. These values are not necessarily unique. The universal approximation property, thus, implies that if the NN inputs $\boldsymbol{x}_{\text{in}}$ are chosen to reflect the functional dependency of $\Delta(\cdot)$, then $\bar{\epsilon}$ may be made arbitrarily small given a sufficient number of hidden layer neurons n_2.

The adaptive signal $\bar{\boldsymbol{\nu}}_{\text{ad}}$ actually contains two terms,

$$\bar{\boldsymbol{\nu}}_{\text{ad}} = \boldsymbol{\nu}_{\text{ad}} + \boldsymbol{\nu}_r = \begin{bmatrix} \boldsymbol{a}_{\text{ad}} + \boldsymbol{a}_r \\ \boldsymbol{\alpha}_{\text{ad}} + \boldsymbol{\alpha}_r \end{bmatrix} \tag{41}$$

where $\boldsymbol{\nu}_{\text{ad}}$ is the output of the SHL NN described earlier. For an air vehicle with adaptation in all degrees of freedom, $\boldsymbol{\nu}_{\text{ad}} \in \mathcal{R}^6$, where the first three outputs, $\boldsymbol{a}_{\text{ad}}$, approximates $\boldsymbol{\Delta}_a$ and the last three outputs, $\boldsymbol{\alpha}_{\text{ad}}$, approximate $\boldsymbol{\Delta}_\alpha$, and is consistent with the definition of the error in Eq. (19). Here $\boldsymbol{\nu}_r = [\boldsymbol{a}_r^T, \boldsymbol{\alpha}_r^T]^T \in \mathcal{R}^6$ is a robustifying signal that arises in the proof of boundedness.

Boundedness

Associated with the tracking error dynamics given in Eq. (24) is the Lyapunov function

$$A^TP + PA + Q = 0 \tag{42}$$

Choose positive definite[15]

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \tag{43}$$

where

$$Q_1 = \begin{bmatrix} R_dR_p^2 & 0 \\ 0 & R_dR_p \end{bmatrix} > 0 \tag{44}$$

$$Q_2 = \begin{bmatrix} K_dK_p^2 & 0 \\ 0 & K_dK_p \end{bmatrix} > 0 \tag{45}$$

Making use of the property that R_p, R_d, K_p, and $K_d > 0$ and diagonal results in a positive definite solution for P. Hence,

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \tag{46}$$

where

$$P_1 = \begin{bmatrix} R_p^2 + \frac{1}{2}R_pR_d^2 & \frac{1}{2}R_pR_d \\ \frac{1}{2}R_pR_d & R_p \end{bmatrix} > 0 \tag{47}$$

$$P_2 = \begin{bmatrix} K_p^2 + \frac{1}{2}K_pK_d^2 & \frac{1}{2}K_pK_d \\ \frac{1}{2}K_pK_d & K_p \end{bmatrix} > 0 \tag{48}$$

The inputs to the NN have to be chosen to satisfy the functional dependence of $\Delta(\boldsymbol{x}, \hat{\boldsymbol{\delta}})$ and may be specified as

$$\bar{\boldsymbol{x}}^T = \begin{bmatrix} b_v & \boldsymbol{x}_{\text{in}}^T \end{bmatrix}, \qquad \boldsymbol{x}_{\text{in}}^T = \begin{bmatrix} \boldsymbol{x}_c^T & \boldsymbol{e}_r^T & \boldsymbol{e}^T & \boldsymbol{\nu}_{\text{ad}}^T & \|Z\|_F \end{bmatrix} \quad (49)$$

Assumption 1. The norm of the ideal weights (V^*, W^*) is bounded by a known positive value,

$$0 < \|Z^*\|_F \leq \bar{Z} \quad (50)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Assumption 2. The external vehicle state command $\boldsymbol{x}_c$ is bounded,

$$\|\boldsymbol{x}_c\| \leq \bar{x}_c \quad (51)$$

Assumption 3. The states of the reference model, remain bounded for permissible plant and actuator dynamics. Hence,

$$\|\boldsymbol{e}_r\| \leq \bar{e}_r \quad (52)$$

Assumption 4. The model error arising from using a dynamic model to estimate actuator position ϵ_g is assumed to be bounded as

$$\|\boldsymbol{\epsilon}_g\| \leq \bar{\epsilon}_g \quad (53)$$

Assumption 5. Note that $\boldsymbol{\Delta}$ depends on ν_{ad} through the pseudo-controls $\boldsymbol{a}_{\text{des}}$ and $\boldsymbol{\alpha}_{\text{des}}$, whereas $\boldsymbol{\nu}_{\text{ad}}$ has to be designed to cancel Δ. Hence the existence and uniqueness of a fixed-point-solution for $\boldsymbol{\nu}_{\text{ad}} = \boldsymbol{\Delta}(\boldsymbol{x}, \boldsymbol{\nu}_{\text{ad}})$ is assumed. Sufficient conditions[23] for this assumption are also available.

Theorem. Consider the system given by Eqs. (1–4) together with the inverse law (9) and assumptions 1–5, where

$$\boldsymbol{r} = (\boldsymbol{e}^T P B)^T \quad (54)$$

$$\bar{\boldsymbol{\nu}}_{\text{ad}} = \boldsymbol{\nu}_{\text{ad}} + \boldsymbol{\nu}_r \quad (55)$$

$$\boldsymbol{\nu}_{\text{ad}} = W^T \boldsymbol{\sigma}(V^T \bar{\boldsymbol{x}}) \quad (56)$$

$$\boldsymbol{\nu}_r = -K_r(\|Z\|_F + \bar{Z})(\|\boldsymbol{e}\|/\|\boldsymbol{r}\|)\boldsymbol{r} \quad (57)$$

with diagonal $K_r > 0 \in \mathcal{R}^{6\times 6}$, and where W and V satisfy the adaptation laws

$$\dot{W} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' V^T \bar{\boldsymbol{x}})\boldsymbol{r}^T + \kappa\|\boldsymbol{e}\| W]\Gamma_W \quad (58)$$

$$\dot{V} = -\Gamma_V[\bar{\boldsymbol{x}}(\boldsymbol{r}^T W^T \boldsymbol{\sigma}') + \kappa\|\boldsymbol{e}\| V] \quad (59)$$

with Γ_W and $\Gamma_V > 0$ and scalar $\kappa > 0$, guarantees that reference model tracking error $\boldsymbol{e}$ and NN weights (W, V) are uniformly ultimately bounded.

Proof. See the Appendix. □

Corollary. All plant states p, v, q, and ω are uniformly ultimately bounded.

Proof. If the ultimate boundedness of $\boldsymbol{e}$, W, and V from the theorem is taken together with assumption 3, the uniform ultimate boundedness of the plant states is immediate following the definition of the reference model tracking error in Eq. (19). □

Application to an Autonomous Helicopter

Consider the application of the combined inner- and outer-loop adaptive architecture to the trajectory control of a helicopter. The dynamics[3,5,24] of the helicopter may be modeled in the same form as Eqs. (1–4). Most small helicopters include a Bell–Hiller stabilizer bar, which provides provide lagged rate feedback and is a source of unmodeled dynamics. The nonlinear model used for simulation in this work included the stabilizer bar dynamics. Additionally, blade flapping and other aspects such as gear and engine dynamics were also modeled.

Approximate Model

An approximate model for the attitude dynamics of the helicopter was generated by linearizing the nonlinear model around hover and neglecting coupling between the attitude and translational dynamics as well as the stabilizer bar,

$$\boldsymbol{\alpha}_{\text{des}} = \hat{A}_1 \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \hat{A}_2 \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \hat{B} \left(\underbrace{\begin{bmatrix} \delta_{\text{lat}} \\ \delta_{\text{lon}} \\ \delta_{\text{ped}} \end{bmatrix}}_{\text{des}} - \underbrace{\begin{bmatrix} \delta_{\text{lat}} \\ \delta_{\text{lon}} \\ \delta_{\text{ped}} \end{bmatrix}}_{\text{trim}} \right) \quad (60)$$

or

$$\boldsymbol{\alpha}_{\text{des}} = \hat{A}_1 \boldsymbol{\omega}_B + \hat{A}_2 \boldsymbol{v}_B + \hat{B}(\boldsymbol{\delta}_{m_{\text{des}}} - \boldsymbol{\delta}_{m_{\text{trim}}}) \quad (61)$$

where $\hat{A}_1$ and $\hat{A}_2$ are the attitude and translational dynamics, respectively, $\boldsymbol{\omega}_B$ is the angular velocity of the body with respect to the Earth expressed in the body frame, $\boldsymbol{v}_B$ is the body velocity vector with respect to the Earth expressed in the body frame, and $\boldsymbol{\delta}_{m_{\text{trim}}}$ is the trim control vector that is consistent with the linear model. Choosing the control matrix $\hat{B}$ such that it is invertible, the moment controls may be evaluated as

$$\boldsymbol{\delta}_{m_{\text{des}}} = \hat{B}^{-1}(\boldsymbol{\alpha}_{\text{des}} - \hat{\boldsymbol{A}}_1 \omega_B - \hat{A}_2 \boldsymbol{v}_B) + \boldsymbol{\delta}_{m_{\text{trim}}} \quad (62)$$

The translational dynamics were modeled as a point mass with a thrust vector that may be oriented in a given direction as shown in Fig. 3. More involved inverses[25] may be used, but the simple relationships between thrust, attitude, and accelerations suffice when used with adaptation,

$$\boldsymbol{a}_{\text{des}} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{\text{coll}}} \end{bmatrix} \left(\delta_{\text{coll}_{\text{des}}} - \delta_{\text{coll}_{\text{trim}}} \right) + L_{bv}\boldsymbol{g} \quad (63)$$

where $Z_{\delta_{\text{coll}}}$ is the control derivative for acceleration in the vertical axis. L_{bv} is the direction cosine matrix that transforms a vector from the vehicle (or local) frame to the body frame and $\boldsymbol{g}$ is an assumed gravity vector. The desired specific force along the body z axis may be evaluated as

$$f_{\text{sf}} = (\boldsymbol{a}_{\text{des}} - L_{bv}\boldsymbol{g})_3 \quad (64)$$

The required collective input may be evaluated as

$$\delta_{\text{coll}_{\text{des}}} = f_{\text{sf}} \big/ Z_{\delta_{\text{coll}}} + \delta_{\text{coll}_{\text{trim}}} \quad (65)$$

The attitude augmentation required to orient the thrust vector to attain the desired translational accelerations are given by the following small-angle corrections from the current reference body attitude and attitude command:

$$\Delta\Phi_1 = a_{\text{des}_2}/f_{\text{sf}}, \qquad \Delta\Phi_2 = -a_{\text{des}_1} \big/ f_{\text{sf}}, \qquad \Delta\Phi_3 = 0 \quad (66)$$

For this simplified helicopter model, heading change has no effect on accelerations in the x, y plane, and hence, $\Delta\Phi_3 = 0$. These three

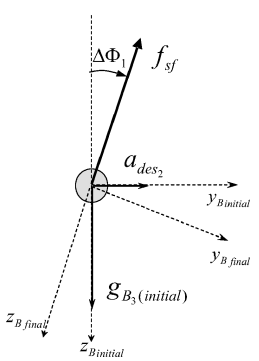

Fig. 3 Point mass model for outer-loop inversion.

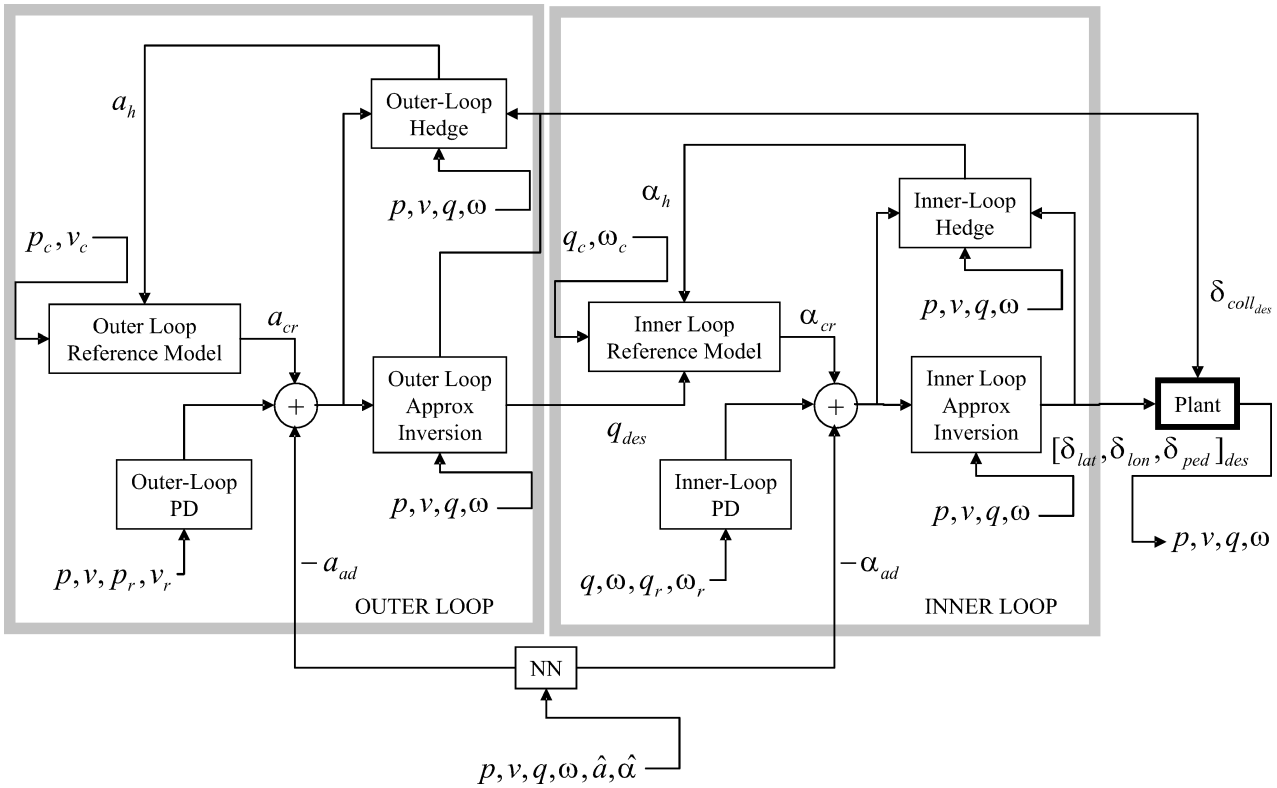

Fig. 4 Detailed inner- and outer-loop controller architecture for an autonomous helicopter.

correction angles may now be used to generate the attitude quaternion correction desired by the outer loop. Thus,

$$\boldsymbol{q}_{\text{des}} = \boldsymbol{q}(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3) \tag{67}$$

where $\boldsymbol{q}(.)$ is a function[17] that expresses an Euler-angles-based rotation as a quaternion. The overall detailed controller architecture is shown in Fig. 4.

Remark 3. If the desired specific force f_{sf} is close to zero, which occurs when the desired acceleration in the body z axis is the same as the component of gravity vector along that axis, then, Eq. (66) is undefined. To overcome this problem, one can impose a restriction where Eq. (66) is only computed if $|f_{\text{sf}}| > \bar{f}_{\text{sf}}$, where $\bar{f}_{\text{sf}} > 0$ and is a lower limit. Essentially it means, do not bother using attitude unless the desired specific force is greater than $\bar{f}_{\text{sf}}$.

Reference Model

A reasonable choice for the reference model dynamics is given by

$$\boldsymbol{a}_{\text{cr}} = R_p(\boldsymbol{p}_c - \boldsymbol{p}_r) + R_d(\boldsymbol{v}_c - \boldsymbol{v}_r), \qquad \dot{\boldsymbol{v}}_r = \boldsymbol{a}_{\text{cr}} - \boldsymbol{a}_h \tag{68}$$

$$\boldsymbol{\alpha}_{\text{cr}} = K_p[\tilde{\boldsymbol{Q}}(\boldsymbol{q}_c \oplus \boldsymbol{q}_{\text{des}}, \boldsymbol{q}_r)] + K_d(\boldsymbol{\omega}_c - \boldsymbol{\omega}_r)$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{\text{cr}} - \boldsymbol{\alpha}_h \tag{69}$$

where R_p, R_d, K_p, and K_d are the same gains used for the PD compensator in Eq. (21). If limits on the angular rate or translational velocities are to be imposed, then they may be easily included in the reference model dynamics by modifying $\boldsymbol{a}_{\text{cr}}$ and $\boldsymbol{\alpha}_{\text{cr}}$ to the following form

$$\boldsymbol{a}_{\text{cr}} = R_d\left[\boldsymbol{v}_c - \boldsymbol{v}_r + \text{sat}\left(R_d^{-1}R_p(\boldsymbol{p}_c - \boldsymbol{p}_r), v_{\text{lim}}\right)\right]$$

$$\boldsymbol{\alpha}_{\text{cr}} = K_d\left[\boldsymbol{\omega}_c - \boldsymbol{\omega}_r + \text{sat}\left(K_d^{-1}K_p\tilde{\boldsymbol{Q}}, \omega_{\text{lim}}\right)\right] \tag{70}$$

where the functional dependence of $\tilde{\boldsymbol{Q}}$ has been dropped for clarity and is the same as in Eq. (69). The function sat $(\cdot)$ is the saturation function and v_{lim} and ω_{lim} are the translational and angular rate limits respectively.

Remark 4. Note that there are no limits placed on the externally commanded position, velocity, angular rate, or attitude. For example, in the translational reference model, if a large position step is commanded, $\boldsymbol{p}_c = [1000, 0, 0]^T$ ft and $\boldsymbol{v}_c = [0, 0, 0]^T$ ft/s, the speed at which this large step will be achieved is v_{lim}. On the other hand if

$$\boldsymbol{p}_c = \int \boldsymbol{v}_c \, \text{d}t$$

and $\boldsymbol{v}_c = [60, 0, 0]^T$ ft/s, the speed of the vehicle will be 60 ft/s. Similarly, ω_{lim} dictates how fast large attitude errors will be corrected. Additionally, the aggressiveness with which translational accelerations will be pursued by tilting the body may be governed by limiting the magnitude of q_{des} to the scalar limit q_{lim}.

Choice of Gains R_p, R_d, K_p, and K_d

When the combined adaptive inner–outer-loop controller for position and attitude control is implemented, the poles for the combined error dynamics must be selected appropriately. The following analysis applies to the situation where inversion model error is compensated for accurately by the NN and we assume that the system is exactly feedback linearized. The inner loop and outer loop each represent a second-order system and the resulting position dynamics $p(s)/p_c(s)$ are fourth order in directions perpendicular to the rotor spin axis.

When the closed-loop longitudinal dynamics, near hover, are considered, and with an acknowledgment of an abuse of notation, it may be written as

$$\ddot{x} = a_{\text{des}} = \ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x) \tag{71}$$

$$\ddot{\theta} = \alpha_{\text{des}} = \ddot{\theta}_g + K_d(\dot{\theta}_g - \dot{\theta}) + K_p(\theta_g - \theta) \tag{72}$$

where R_p, R_d, K_p, and K_d are the PD compensator gains for the inner loop (pitch angle) and outer loop (fore–aft position). Now x is the position, θ the attitude, and θ_g the attitude command. Normally,

$\theta_g = \theta_c + \theta_{\text{des}}$, where θ_c is the external command and θ_{des} the outer-loop-generated attitude command. Here, we assume that the external attitude command and its derivatives are zero; hence, $\theta_g = \theta_{\text{des}}$. In the following development, the transfer function $x(s)/x_c(s)$ is found and used to place the poles of the combined inner–outer-loop system in terms of the PD compensator gains.

When contributions of $\dot{\theta}_g(s)$ and $\ddot{\theta}_g(s)$ are ignored, the pitch dynamics Eq. (72) may be rewritten in the form of a transfer function as

$$\theta(s) = [\theta(s)/\theta_g(s)]\theta_g(s) = \left[K_p \Big/ \left(s^2 + K_d s + K_p\right)\right]\theta_g(s) \quad (73)$$

If the outer-loop linearizing transformation used to arrive at Eq. (71) has the form $\ddot{x} = f\theta$, where $f = -g$ and g is gravity, it may be written as

$$s^2 x(s) = f\theta(s) \quad (74)$$

The outer-loop attitude command may be generated as

$$\theta_{\text{des}} = \ddot{x}_{\text{des}}/f = a_{\text{des}}/f \quad (75)$$

Note that $\theta_g = \theta_{\text{des}}$; if $\theta_c = 0$,

$$\theta_g = \theta_{\text{des}} = (1/f)[\ddot{x}_c + R_d(\dot{x}_c - \dot{x}) + R_p(x_c - x)] \quad (76)$$

When Eqs. (73) and (76) are used in Eq. (74),

$$s^2 x(s) = \frac{K_p[s^2 x_c + R_d s(x_c - x) + R_p(x_c - x)]}{s^2 + K_d s + K_p} \quad (77)$$

Rearranging Eq. (77) results in the following transfer function:

$$\frac{x(s)}{x_c(s)} = \frac{K_p s^2 + K_p R_d s + K_p R_p}{s^4 + K_d s^3 + K_p s^2 + K_p R_d s + K_p R_p} \quad (78)$$

One way to choose the gains is by examining a fourth-order characteristic polynomial written as the product of two second-order systems,

$$\begin{aligned}\Upsilon(s) &= \left(s^2 + 2\zeta_o\omega_o + \omega_o^2\right)\left(s^2 + 2\zeta_i\omega_i + \omega_i^2\right) \\ &= s^4 + (2\zeta_i\omega_i + 2\zeta_o\omega_o)s^3 + \left(\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2\right)s^2 \\ &\quad + \left(2\zeta_o\omega_o\omega_i^2 + 2\omega_o^2\zeta_i\omega_i\right)s + \omega_o^2\omega_i^2 \end{aligned} \quad (79)$$

Fig. 5 GTMax helicopter.

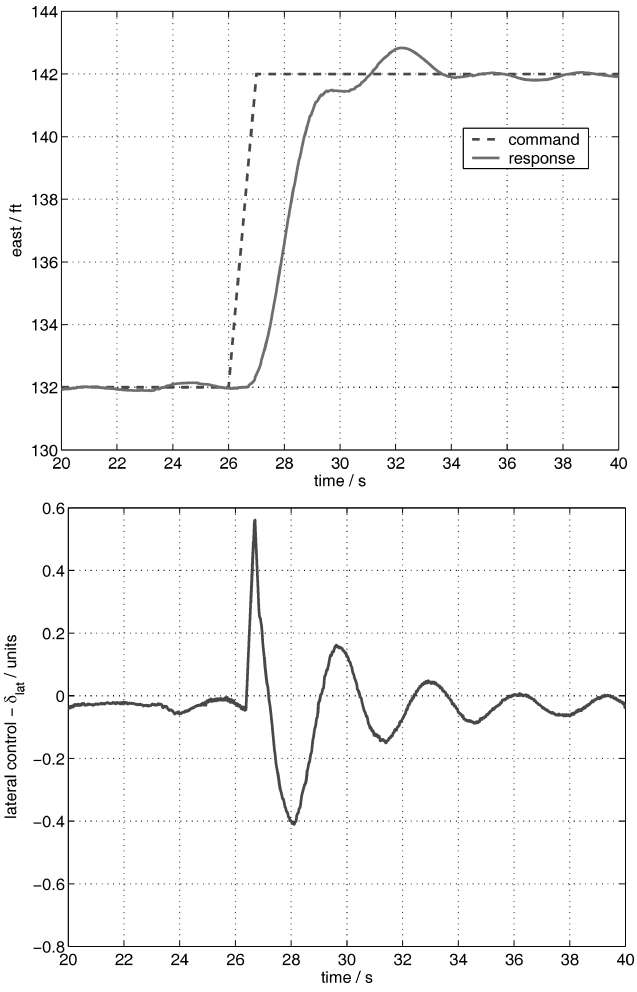

Fig. 6 Response to a 20-ft step in the lateral direction.

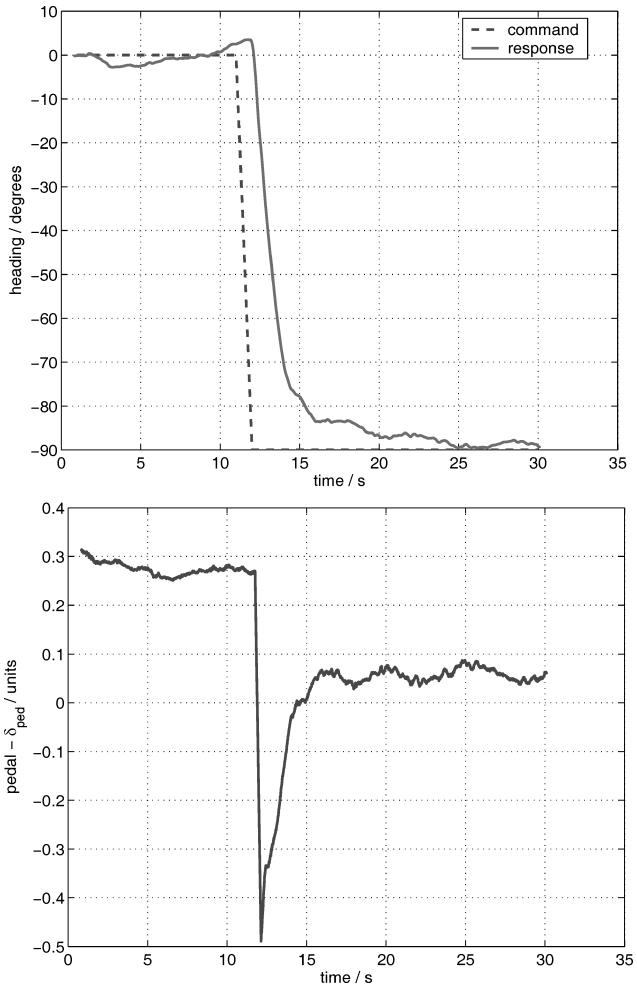

Fig. 7 Response to a 90-deg heading command.

where the subscripts i and o, represent the inner and outerloop values respectively.

Comparing the coefficients of the poles of Eq. (78) and Eq. (79) allows the gains to be expressed as a function of the desired pole locations for each axis in turn,

$$R_p = \frac{\omega_o^2\omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2}, \qquad R_d = 2\frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2}$$

$$K_p = \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2, \qquad K_d = 2\zeta_i\omega_i + 2\zeta_o\omega_o \quad (80)$$

Imposing Response Characteristics

One method[6,14] to evaluate the performance of the control system is to use the metrics given in Aeronautical Design Standard-33 (Ref. 26) handling qualities specifications. When there is no saturation the hedging signals, $\boldsymbol{a}_h$ and $\boldsymbol{\alpha}_h$ are zero. When it is assumed that the adaptation has reached its ideal values of (V^*, W^*), then

$$\dot{\boldsymbol{v}} = \boldsymbol{a}_{\rm cr} + \boldsymbol{a}_{\rm pd} + \boldsymbol{\epsilon}_a, \qquad \dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}_{\rm cr} + \boldsymbol{\alpha}_{\rm pd} + \boldsymbol{\epsilon}_a \quad (81)$$

where $\boldsymbol{\epsilon}_a$ and $\boldsymbol{\epsilon}_\alpha$ are bounded by $\bar{\epsilon}$. Additionally, the Lyapunov analysis provides guaranteed model following, which implies $a_{\rm pd}$ and $\boldsymbol{\alpha}_{\rm pd}$ are small. Thus, $\dot{\boldsymbol{v}} \approx \boldsymbol{a}_{\rm cr}$ and $\dot{\boldsymbol{\omega}} \approx \boldsymbol{\alpha}_{\rm cr}$. Hence, as long as the preceding assumptions are valid over the bandwidth of interest, the desired response characteristics may be encoded into the reference model $\boldsymbol{a}_{\rm cr}$ and $\boldsymbol{\alpha}_{\rm cr}$.

Results

The proposed guidance and control architecture was applied to the Georgia Institute of Technology Yamaha R-Max helicopter (GTMax) shown in Fig. 5. The basic GTMax helicopter weighs about 157 lb and has a main rotor radius of 5.05 ft. Nominal rotor speed is 850 rpm. Its practical payload capability is about 66 lb with a flight endurance of greater than 60 min. It is also equipped with a Bell–Hiller stabilizer bar. Its avionics package includes a Pentium 266 flight control computer, an inertial measurement unit (IMU), a global positioning system, a three-axis magnetometer, and a sonar altimeter. The control laws presented in this paper were first implemented in simulation[27] using a nonlinear helicopter model that included flapping and stabilizer bar dynamics. Wind and gust models were also included. Additionally, models of sensors with associated noise characteristics were implemented. Many aspects of hardware, such as the output of sensor model data as serial packets, were simulated. This introduced digitization errors as would exist in real life and also allowed testing of many flight specific components such as sensor drivers.[28] The navigation system consists of a 17-state Kalman filter to estimate variables such as attitude and terrain altitude. The navigation filter was executed at 100 Hz and corresponds to the highest rate at which the IMU is able to provide data. Controller calculations occurred at 50 Hz. The control laws were first implemented as C-code and tested in simulation. Because almost all aspects specific to flight testing were included in the simulation environment, a subset of the code from the simulation environment was implemented on the main flight computer. During flight, ethernet and serial-based data links provided a link to the ground station computer that allowed monitoring and uploading of way points. A simple kinematics-based trajectory generator (with limits on accelerations) was used to generate smooth consistent trajectories $(\boldsymbol{p}_c, \boldsymbol{v}_c, \boldsymbol{q}_c, \boldsymbol{\omega}_c)$ for the controller. Various moderately aggressive maneuvers were performed during flight to test the performance of the trajectory-tracking controller. Testing of the controller began with simple hover, followed by step

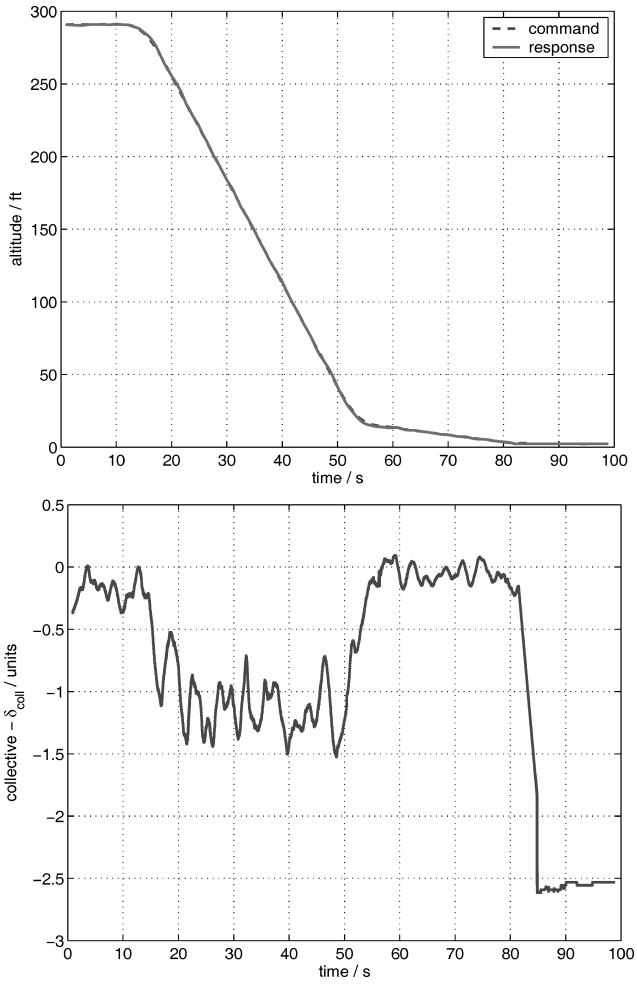

Fig. 8 Automatic landing maneuver.

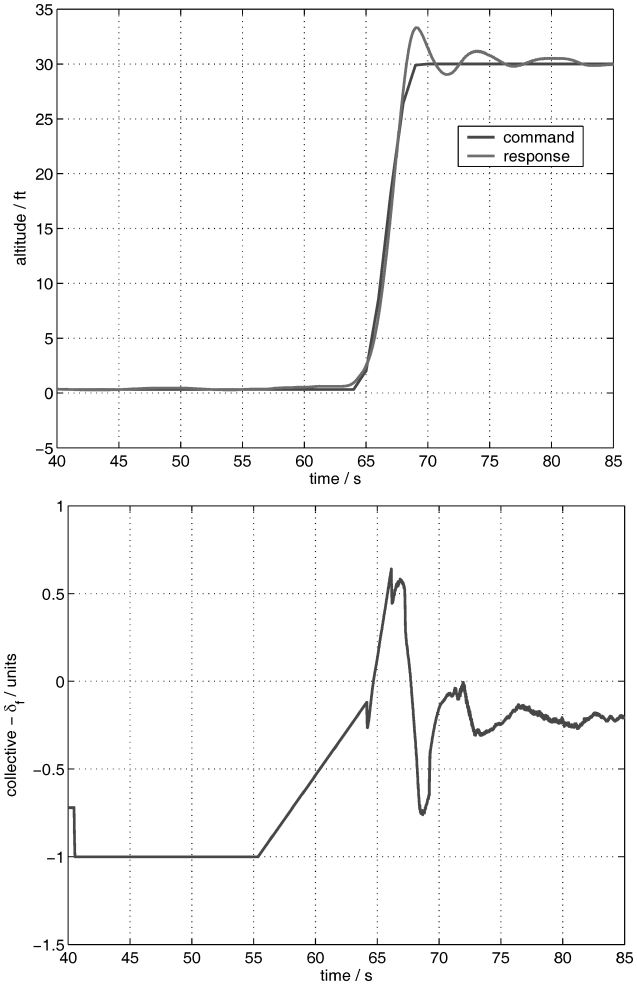

Fig. 9 Automatic takeoff maneuver.

responses and way-point navigation. Following initial flight tests, the aggressiveness of the trajectory was increased by relaxing acceleration limits in the trajectory generator and relaxing ω_{lim} and v_{lim} in the reference models. Tracking error performance was increased by increasing the desired bandwidth of the controllers. Selected results from these flight tests are provided in the following sections.

Parameter Selections

The controller parameters for the inner loop involved choosing K_p and K_d based on a natural frequency of 2.5, 2, and 3 rad/s for the roll, pitch, and yaw channels, respectively, and damping ratio of 1.0. For the outer loop, R_p and R_d were chosen based on a natural frequency of 2, 2.5, and 3 rad/s for the x, y, and z body axis, all with a damping ratio of unity. The NN was chosen to have five hidden layer neurons. The inputs to the network included body axis velocities and rates as well as the estimated pseudocontrols, that is, $\boldsymbol{x}_{\text{in}}^T = [\boldsymbol{v}_B^T,\ \boldsymbol{\omega}_B^T,\ \hat{\boldsymbol{a}}^T,\ \hat{\boldsymbol{\alpha}}^T]$. The output layer learning rates[15] Γ_W were set to unity for all channels, and a learning rate of $\Gamma_V = 10$ was set for all inputs. Limits on maximum translation rate and angular rate in the reference model dynamics were set to $v_{\text{lim}} = 10$ ft/s and $\omega_{\text{lim}} = 2$ rad/s. Additionally, attitude corrections from the outer loop, $\boldsymbol{q}_{\text{des}}$, was limited to 30 deg.

With regard to actuator magnitude limits, the Yamaha RMax helicopter has a radio-control transmitter that the pilot may use to fly the vehicle manually. The full deflections available on the transmitter sticks in each of the channels were mapped as $\delta_{\text{lat}}, \delta_{\text{lon}}, \delta_{\text{ped}} \in [-1, 1]$ corresponding to the full range of lateral tilt and longitudinal tilt of the swash plate and full range of tail rotor blade pitch. The collective was mapped as $\delta_{\text{coll}} \in [-2.5, 1]$, corresponding to the full range of main rotor blade pitch available to the human pilot. The dynamic characteristics of the actuators were not investigated in detail. Instead, conservative rate limits were artificially imposed in software. Noting that

$$\boldsymbol{\delta} = [\delta_{\text{coll}}, \delta_{\text{lat}}, \delta_{\text{lon}}, \delta_{\text{ped}}]$$

the actuator model used for PCH purposes, as well as artificially limiting the controller output, has the form

$$\dot{\hat{\boldsymbol{\delta}}} = \lim_{\lambda \to +\infty} \text{sat}[\lambda(\text{sat}(\boldsymbol{\delta}_{\text{des}}, \boldsymbol{\delta}_{\min}, \boldsymbol{\delta}_{\max}) - \hat{\boldsymbol{\delta}}), \dot{\boldsymbol{\delta}}_{\min}, \dot{\boldsymbol{\delta}}_{\max}] \qquad (82)$$

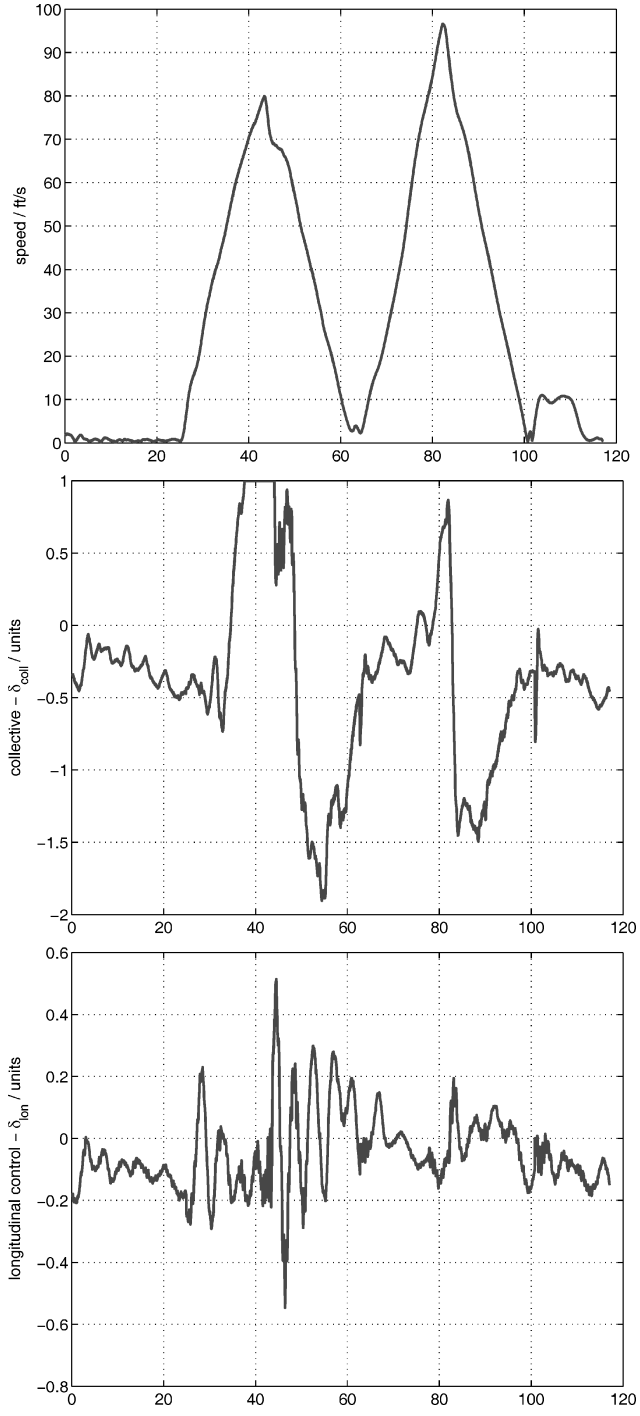

Fig. 10 High-speed forward flight up to 97 ft/s.

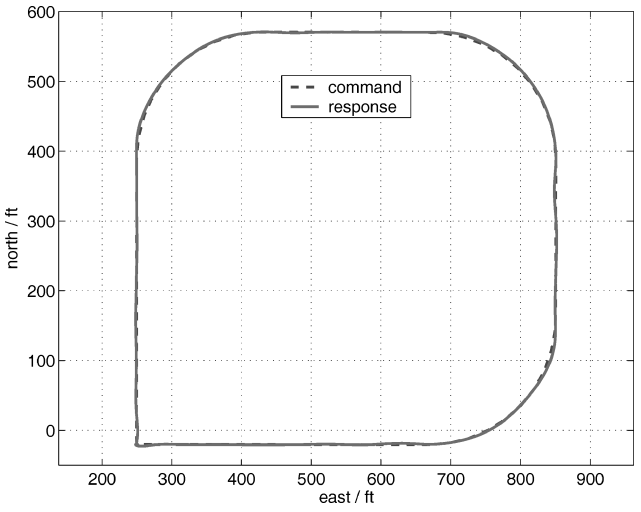

Fig. 11 Flying a square pattern at 30 ft/s.

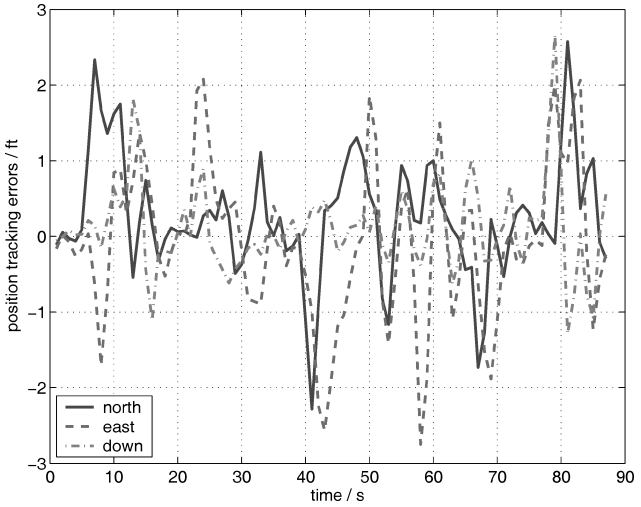

Fig. 12 Command tracking errors while flying a square pattern at 30 ft/s.

where $\hat{\boldsymbol{\delta}}$ is limited to lie in the interval $[\boldsymbol{\delta}_{\min}, \boldsymbol{\delta}_{\max}]$. The discrete implementation has the form

$$\hat{\boldsymbol{\delta}}[k+1] = \text{sat}[\hat{\boldsymbol{\delta}}[k] + \text{sat}(\text{sat}(\boldsymbol{\delta}_{\text{des}}, \boldsymbol{\delta}_{\min}, \boldsymbol{\delta}_{\max}) - \hat{\boldsymbol{\delta}}[k], \Delta T \dot{\boldsymbol{\delta}}_{\min}, \Delta T \dot{\boldsymbol{\delta}}_{\max}), \boldsymbol{\delta}_{\min}, \boldsymbol{\delta}_{\max}] \quad (83)$$

where ΔT is the sampling time. The magnitude limits were set to

$$\boldsymbol{\delta}_{\min} = [-2.5, -1, -1, -1]^T, \qquad \boldsymbol{\delta}_{\max} = [1, 1, 1, 1]^T \quad (84)$$

units, and the rate limits were set to

$$\dot{\boldsymbol{\delta}}_{\min} = [-4, -2, -2, -2]^T, \qquad \dot{\boldsymbol{\delta}}_{\max} = [4, 2, 2, 2]^T \quad (85)$$

units per second.

Flight Test

Finally, the controller was flight tested on the GTMax helicopter shown in Fig. 5. A lateral position step response is shown in Fig. 6. [Note that, during flight tests, variables were sampled at varying rates to conserve memory and datalink bandwidth. The trajectory commands p_c, v_c, q_c, and ω_c were sampled at 1 Hz, actuator deflections δ_{coll}, δ_{lon}, δ_{lat}, and δ_{ped} were sampled at 50 Hz; vehicle position and speed was sampled at 50 Hz. Because the command vector is sampled at a low rate (1 Hz), a step command appears as a fast ramp.] The vehicle heading was regulated due north during this maneuver. Lateral control deflections during the maneuver were recorded and are also shown. A step heading command response and pedal control history is shown in Fig. 7.

During takeoff and landing phases, a range sensor (sonar) is used to maintain and update the estimated local terrain altitude in the navigation system. The sonar is valid up to 8 ft above the terrain, sufficient for landing and takeoff purposes. Figure 8 shows the altitude and collective profile during a landing. The vehicle starts at an initial hover at 300 ft, followed by a descent at 7 ft/s, until the vehicle is 15 ft above the estimated terrain. The vehicle then descends at 0.5 ft/s until weight-on-skids is automatically detected, at which point the collective is slowly ramped down. Automatic takeoff (Fig. 9) is similar where the collective is slowly ramped up until weight-on-skids is no longer detected. Note that NN adaptation is active all times except when weight-on-skids is active. Additionally, when the weight is on the skids, the collective ramp-up during takeoff and ramp-down during landing is open loop.

The approximate model used to compute the dynamic inverse [Eqs. (63) and (60)] is based on a linear model of the dynamics in hover. To evaluate controller performance at different points of the envelope, the vehicle was commanded to track a trajectory that accelerated up to a speed of 100 ft/s. To account for wind, an upwind and downwind leg were flown. In the upwind leg the vehicle accelerated up to 80 ft/s, and during the backward leg, the vehicle accelerated up to a speed of 97 ft/s as shown in Fig. 10. Collective and longitudinal control deflections are also shown. In the upwind leg, the collective is saturated and the vehicle is unable to accelerate further. The longitudinal control deflections behave nominally as the vehicle accelerates and decelerates through a wide range of the envelope. The NN is able to adapt to rapidly changing flight conditions, from the baseline inverting design at hover through to the maximum speed of the aircraft. A conventional proportional–integral–derivative design would have required scheduling of gains throughout the speed range. More significantly, classical design would require accurate models at each point, which our design does not. In addition to flight at high speeds, tracking performance was evaluated at moderate speeds,

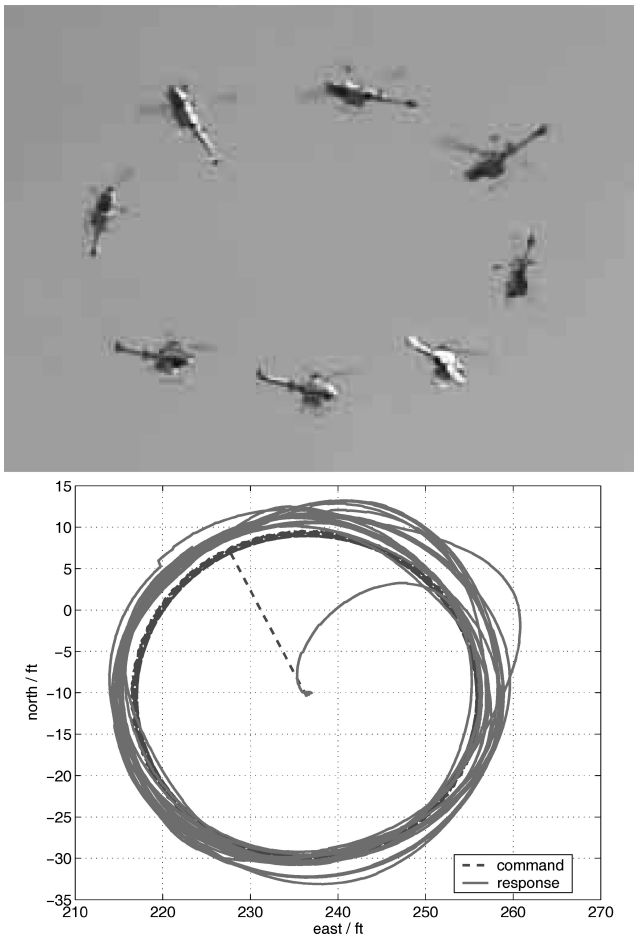

Fig. 13 Circular maneuver, with 360-deg heading changes during the circuit.

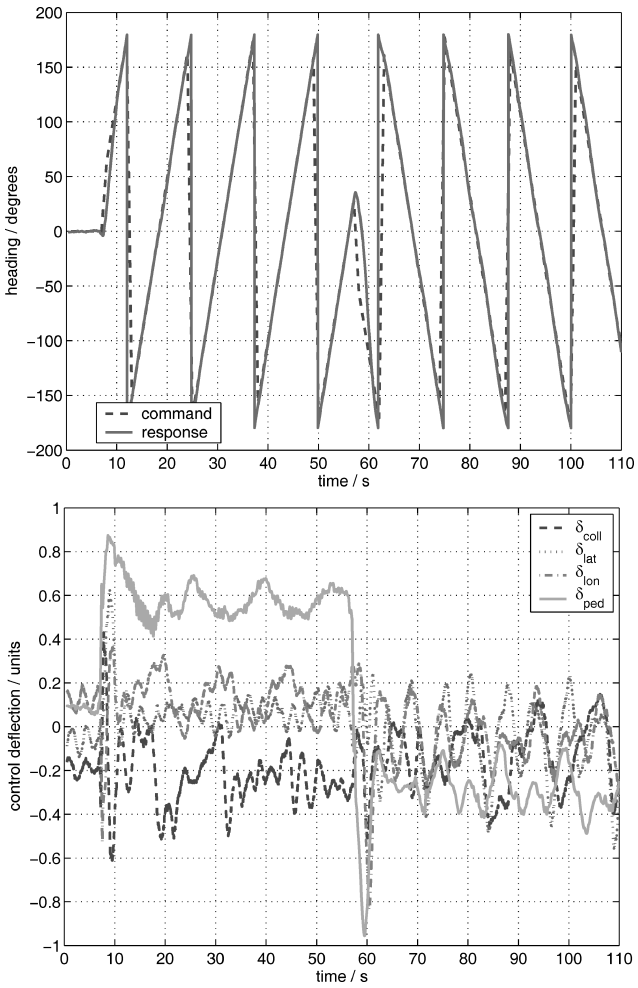

Fig. 14 Heading tracking during circular maneuver and control time history.

where a square pattern was flown at 30 ft/s for which position tracking is shown in Fig. 11. External command position tracking errors are shown in Fig. 12 with a peak total position error 3.3 ft and standard deviation of 0.8 ft.

Many maneuvers such as high-speed flight are quasi steady, in the sense that, once in the maneuver, control deflection changes are only necessary for disturbance rejection. To evaluate performance where the controls have to vary significantly to track the commanded trajectory, the helicopter was commanded to perform a circular maneuver in the north–east plane with constant altitude and a constantly changing heading. The trajectory equations for this maneuver are given by

$$\boldsymbol{p}_c = \begin{bmatrix} \frac{V}{\omega}\cos(\omega t) \\ \frac{V}{\omega}\sin(\omega t) \\ -h \end{bmatrix}, \qquad v_c = \begin{bmatrix} -V\sin(\omega t) \\ V\cos(\omega t) \\ 0 \end{bmatrix}, \qquad \psi_c = \omega t f \tag{86}$$

where t is current time and h is a constant altitude command. V is speed of the maneuver, ω is angular speed of the helicopter around the maneuver origin, and f is number of 360-deg changes in heading to be performed per circuit. If $\omega = \pi/2$ rad/s, the helicopter will complete the circular circuit once every 4 s. If $f = 1$, the helicopter will rotate anticlockwise 360 deg once per circuit. Figure 13 shows the response to such a trajectory with parameters $\omega = 0.5$ rad/s, $f = 1$, and $V = 10$ ft/s. After the initial transition into the circular maneuver, the tracking is seen to be within 5 ft. To visualize the maneuver easily, superimposed still images of the vehicle during the circular maneuver are shown. Both anticlockwise and clockwise heading changes during the maneuver were tested by changing the parameter from $f = 1$ (anticlockwise) to $f = -1$ (clockwise) at $t = 55$ s. Figure 14 shows that heading tracking is good in both cases. The time history of the pedal input δ_{ped} and all other controls during the maneuver is also shown and illustrates how the vehicle has to exercise all of its controls during this maneuver.

Next, the ability of the controller to track a previous manually flown maneuver was tested. First, a human pilot flew a figure eight, three-dimensional pattern with the vehicle. Vehicle state was recorded and was then played back as commands to the adaptive controller. A three-dimensional plot of the pilot and controller flown trajectories are shown in Fig. 15 along with projected ground track. Overall, the tracking in position was measured to be within 11.3 ft of the desired pilot flown trajectory with a standard deviation of 4.7 ft. Finally, a tactically useful maneuver was flown to test controller performance at high speeds and pitch attitudes. The objective of the maneuver is to make a 180-deg velocity change from a forward flight condition of 70 ft/s north to a 70 ft/s forward flight going south. The trajectory command and response in the north–altitude plane is shown in Fig. 16 along with the pitch angle. A time history of the altitude and the collective control deflection is shown in Fig. 17. During the maneuver, the helicopter is commanded to increase altitude by up to 50 ft to minimize saturation of the down collective. In the deceleration phase, the vehicle is able to track the command trajectory well; however, in accelerating to 70 ft/s going south, tracking performance suffers. In both the acceleration and deceleration phases, poor tracking corresponds with saturation of the collective control. The oscillations in altitude in Fig. 17 are because the vehicle is unable to maintain a lower descent rate due to

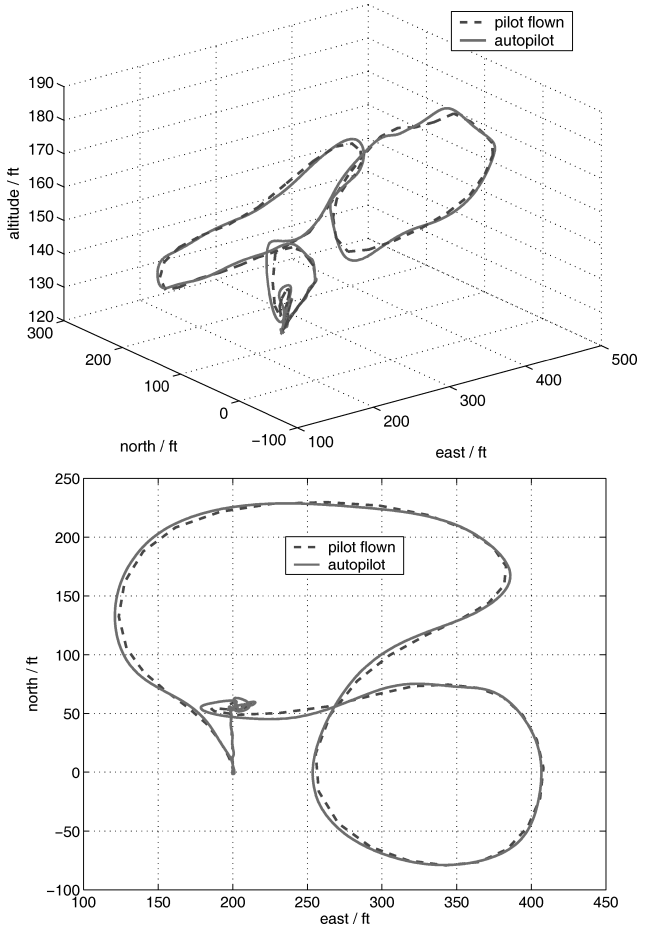

Fig. 15 Three-dimensional view and ground track view of trajectory initially flown manually by a pilot and then tracked by the controller.

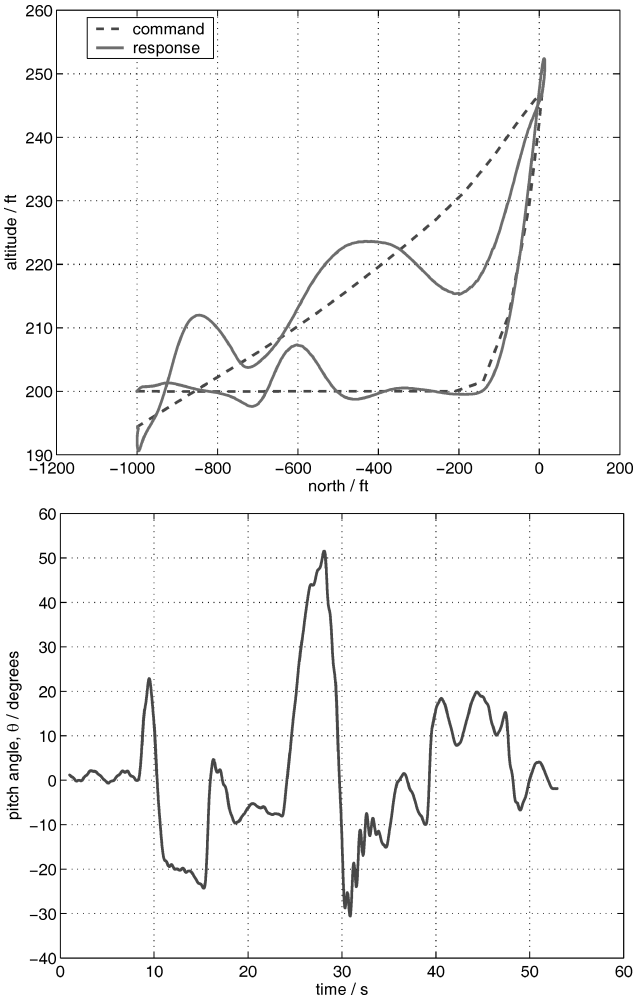

Fig. 16 North altitude and pitch angle profile during a 180-deg velocity change maneuver. Note north axis and altitude axis scales are not equal.

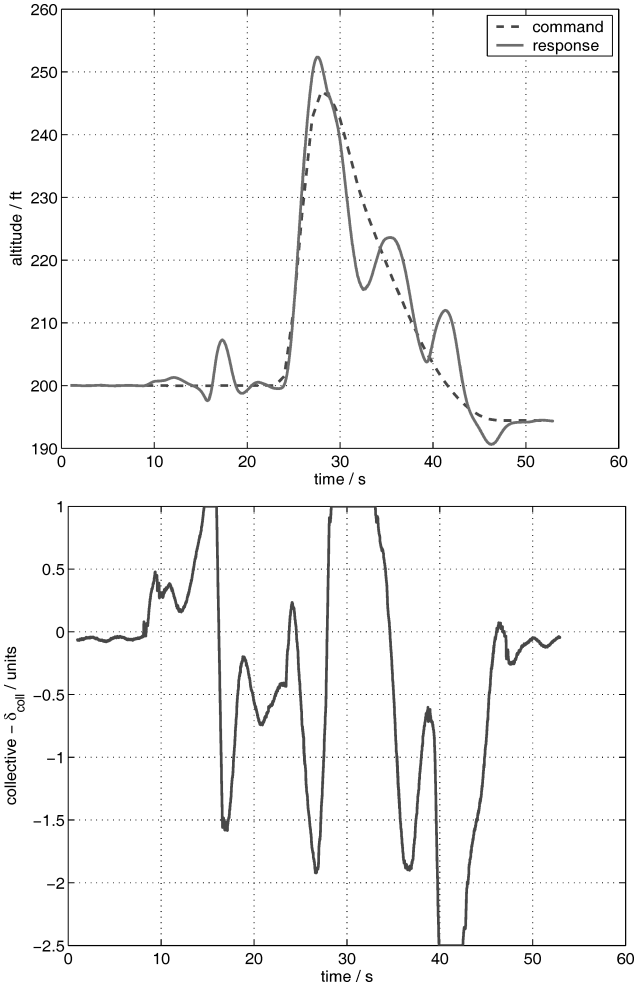

Fig. 17 Altitude and collective control history during a 180-deg velocity change maneuver.

saturation and is expected. The large pitch attitudes experienced is what the outer-loop inversion evaluates as being required to perform such rapid decelerations and accelerations. This experiment is an example of maneuvering where the commanded trajectory is much more aggressive than the capability of the vehicle and is reflected by the extended periods of saturation. It is possible to operate at the limits of the vehicle primarily due to PCH, which protects the adaptation process.

Conclusions

The adaptive controller developed in this paper is able to correct for modeling errors in both the attitude dynamics, as well as the translation dynamics. Using PCH in a novel way allows the outer loop to continue adapting correctly irrespective of the closed-loop attitude dynamics or any limits inserted into the inner-loop reference model. A consolidated external command consisting of position, velocity, attitude, and angular velocity may now be provided to the control system. If the commanded trajectory is not feasible, causing actuator saturation, the controller continues to operate at the actuator limits without affecting adaptation. Additionally, expressions for the poles of the combined inner–outer-loop error dynamics alleviates frequency separation requirements between the inner and outer loops, allowing a higher outer-loop bandwidth, leading to better overall trajectory tracking performance. Flight-test results over various ranges of the flight envelope illustrate that adaptation may be used successfully to correct for significant model error arising from very poor approximate models, in this case, a point mass model for translational dynamics and a linear hover attitude dynamics model. Tracking error is small except in situations where the actuators are saturated.

The control design presented here does not contain assumptions, that limit its application to small unmanned helicopters. Desired response characteristics may be incorporated into the design.

Appendix: Proof of the Theorem

Proof. In the following proof, an asterisk represents ideal values, where the variables $\tilde{W} \triangleq W^* - W$, $\tilde{V} \triangleq V^* - V$, $z = V^T\bar{\boldsymbol{x}}$, and $\tilde{z} = z^* - z$ hold. The arguments to the sigmoidal activation function $\boldsymbol{\sigma}$ are dropped for clarity and conciseness. When it is noted that the sigmoidal functions are bounded, the NN output may be bounded as

$$\nu_{\text{ad}} = W^T\boldsymbol{\sigma}(z) = (W^{*T} - \tilde{W}^T)\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}})$$
$$\|\boldsymbol{\nu}_{\text{ad}}\| \le \alpha_0(\bar{Z} + \|\tilde{Z}\|_F) \tag{A1}$$

for some constant α_0. This allows the inputs to the network to be bounded

$$\bar{\boldsymbol{x}} = \begin{bmatrix} b_v & x_c^T & e_r^T & \boldsymbol{e}^T & \nu_{\text{ad}}^T & \|Z\|_F \end{bmatrix}$$
$$\|\bar{\boldsymbol{x}}\| \le b_v + \bar{x}_c + \bar{e}_r + \|\boldsymbol{e}\| + \alpha_0(\bar{Z} + \|\tilde{Z}\|_F) + \bar{Z} + \|\tilde{Z}\|_F$$
$$= k_0 + k_1\|\tilde{Z}\|_F + \|\boldsymbol{e}\| \tag{A2}$$

where $k_1 = (1 + \alpha_0)$ and $k_0 \triangleq b_v + \bar{x}_c + \bar{e}_r + k_1\bar{Z}$. An expansion of $\boldsymbol{\sigma}(z)$ around the estimated weights is given by

$$\boldsymbol{\sigma}(z^*) = \boldsymbol{\sigma}(z) + \left.\frac{\partial\boldsymbol{\sigma}(s)}{\partial s}\right|_{s=z}(z^* - z) + \mathcal{O}^2(\tilde{z}) \tag{A3}$$

Noting that the derivative of the sigmoidal function σ' is bounded, the higher-order terms of this expansion may be bounded as follows:

$$\mathcal{O}^2(\tilde{z}) = \boldsymbol{\sigma}(z^*) - \boldsymbol{\sigma}(z) - \boldsymbol{\sigma}'\tilde{z}$$
$$\|\mathcal{O}^2(\tilde{z})\| \le 2\alpha_0 + \alpha_1\|\tilde{V}^T\|_F\|\bar{\boldsymbol{x}}\|$$
$$\le 2\alpha_0 + \alpha_1 k_0\|\tilde{Z}\|_F + \alpha_1 k_1\|\tilde{Z}\|_F^2 + \alpha_1\|\tilde{Z}\|_F\|\boldsymbol{e}\| \tag{A4}$$

By the substitution of $\bar{\nu}_{\text{ad}} = \boldsymbol{\nu}_{\text{ad}} + \boldsymbol{\nu}_r$ and $\bar{\boldsymbol{\Delta}} = \boldsymbol{\Delta} + \boldsymbol{\epsilon}_g = \boldsymbol{\nu}_{\text{ad}}^* + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g$, the error dynamics in Eq. (24) may be expressed as

$$\dot{\boldsymbol{e}} = A\boldsymbol{e} + B\left[\boldsymbol{\nu}_{\text{ad}} - \left(\boldsymbol{\nu}_{\text{ad}}^* + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g\right) + \boldsymbol{\nu}_r\right] \tag{A5}$$

Now,

$$\boldsymbol{\nu}_{\text{ad}}^* + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g - \boldsymbol{\nu}_{\text{ad}} = W^{*T}\boldsymbol{\sigma}^* - W^T\boldsymbol{\sigma} + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g$$
$$= W^{*T}[\boldsymbol{\sigma}(z) + \boldsymbol{\sigma}'\tilde{z} + \mathcal{O}^2(\tilde{z})] - W^T\boldsymbol{\sigma} + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g$$

and adding and subtracting $W^T\boldsymbol{\sigma}'z$ and $W^T\boldsymbol{\sigma}\boldsymbol{z}^*$

$$\boldsymbol{\nu}_{\text{ad}}^* + \epsilon + \epsilon_g - \nu_{\text{ad}} = \tilde{W}^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}'z) + W^T\boldsymbol{\sigma}'\tilde{z} + \boldsymbol{w} \tag{A6}$$

where

$$\boldsymbol{w} = \tilde{W}^T\boldsymbol{\sigma}'z^* + W^{*T}\mathcal{O}^2(\tilde{z}) + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_g \tag{A7}$$

the tracking error dynamics may finally be written as

$$\dot{\boldsymbol{e}} = A\boldsymbol{e} + B\{-[\tilde{W}^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}'z) + W^T\boldsymbol{\sigma}'\tilde{z} + \boldsymbol{w}] + \boldsymbol{\nu}_r\} \tag{A8}$$

When the bounds computed earlier are used, the disturbance term $\boldsymbol{w}$ may be bounded as

$$\|\boldsymbol{w}\| = c_0 + c_1\|\tilde{Z}\|_F + c_2\|\boldsymbol{e}\|\|\tilde{Z}\|_F + c_3\|\tilde{Z}\|_F^2 \tag{A9}$$

where c_0, c_1, c_2, and c_3 are computable constants given by

$$c_0 = 2\alpha_0\bar{Z} + \bar{\epsilon} + \bar{\epsilon}_g, \qquad c_1 = 2\alpha_1 k_0 \bar{Z}$$
$$c_2 = 2\alpha_1 k_1 \bar{Z}, \qquad c_3 = 2\alpha_1 \bar{Z} \tag{A10}$$

A Lyapunov candidate function is

$$L(\boldsymbol{e}, \tilde{W}, \tilde{V}) = \tfrac{1}{2}\left[\boldsymbol{e}^T P \boldsymbol{e} + \mathrm{tr}\left(\tilde{W}\Gamma_W^{-1}\tilde{W}^T\right) + \mathrm{tr}\left(\tilde{V}^T\Gamma_V^{-1}\tilde{V}\right)\right] \tag{A11}$$

When the weight update equations of Eq. (58) and Eq. (59) are used, the time derivative of L along trajectories can be expressed as

$$\dot{L} = -\tfrac{1}{2}\boldsymbol{e}^T Q\boldsymbol{e} + \boldsymbol{r}^T(-\boldsymbol{w} + \boldsymbol{\nu}_r) + \kappa\|\boldsymbol{e}\|\mathrm{tr}(\tilde{Z}^T Z) \tag{A12}$$

When $Z = Z^* - \tilde{Z}$ and $\|Z\|_F \geq \|\tilde{Z}\|_F - \bar{Z}$ are used along with the robustifying term of Eq. (57) and it is required that $\lambda_{\min}(K_r) > c_2$ and $\kappa > \|PB\|c_3$, $\dot{L}$ may be bounded as

$$\dot{L} \leq -\tfrac{1}{2}\boldsymbol{e}^T Q\boldsymbol{e} + \|\boldsymbol{r}\|\|\boldsymbol{w}\| - r^T K_r r(\|Z\|_F + \bar{Z})(\|\boldsymbol{e}\|/\|\boldsymbol{r}\|)$$
$$+ \kappa\bar{Z}\|\boldsymbol{e}\|\|\tilde{Z}\|_F - \kappa\|e\|\|\tilde{Z}\|_F^2 \tag{A13}$$

$$\dot{L} \leq -\tfrac{1}{2}\lambda_{\min}(Q)\|\boldsymbol{e}\|^2 + \|\boldsymbol{r}\|\|\boldsymbol{w}\| - \lambda_{\min}(K_r)\|\tilde{Z}\|_F\|\boldsymbol{e}\|\|\boldsymbol{r}\|$$
$$+ \kappa\bar{Z}\|\boldsymbol{e}\|\|\tilde{Z}\|_F - \kappa\|\boldsymbol{e}\|\|\tilde{Z}\|_F^2 \tag{A14}$$

$$\dot{L} \leq -\tfrac{1}{2}\lambda_{\min}(Q)\|\boldsymbol{e}\|^2 + c_0\|PB\|\|\boldsymbol{e}\| + (\|PB\|c_1 + \kappa\bar{Z})\|\boldsymbol{e}\|\|\tilde{Z}\|_F$$
$$- (\lambda_{\min}(K_r) - c_2)\|\boldsymbol{e}\|\|\boldsymbol{r}\|\tilde{Z}\|_F - (\kappa - \|PB\|c_3)\|\boldsymbol{e}\|\|\tilde{Z}\|_F^2 \tag{A15}$$

$$\dot{L} \leq -\tfrac{1}{2}\lambda_{\min}(Q)\|\boldsymbol{e}\|^2 - (\kappa - \|PB\|c_3)\|\boldsymbol{e}\|\|\tilde{Z}\|_F^2 + a_0\|\boldsymbol{e}\|$$
$$+ a_1\|\boldsymbol{e}\|\|\tilde{Z}\|_F \tag{A16}$$

where

$$a_0 \triangleq c_0\|PB\|, \qquad a_1 \triangleq (\|PB\|c_1 + \kappa\bar{Z}) \tag{A17}$$

By the selection of $\lambda_{\min}(Q)$, κ, and learning rates Γ_W and Γ_V, $\dot{L} \leq 0$ everywhere outside a compact set that is entirely within the largest level set of L, which in turn lies entirely within the compact set D (Ref. 15). It can be shown that $\dot{L} \leq 0$ when

$$\|\tilde{Z}\|_F \geq Z_m = \frac{a_1 + \sqrt{a_1^2 + 4a_0(\kappa - \|PB\|c_3)}}{2(\kappa - \|PB\|c_3)} \tag{A18}$$

or

$$\|\boldsymbol{e}\| \geq \frac{(a_0 + a_1 Z_m)}{\frac{1}{2}\lambda_{\min}(Q)} \tag{A19}$$

Thus, for initial conditions within D, the tracking error $\boldsymbol{e}$, and NN weights $\tilde{W}$ and $\tilde{V}$ are uniformly ultimately bounded,[29] with the tracking error bound given by Eq. (A19) treated as an equality. □

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency's Software Enabled Control Program under Contracts 33615-98-C-1341 and 33615-99-C-1500 with John S. Bay as Program Manager, William Koenig of the U.S. Air Force Research Laboratory (AFRL) as Contract Monitor and in part by AFRL Contract F33615-00-C-3021. We also acknowledge the contributions of J. Eric Corban, Henrik Christophersen, Joerg Diettrich, Jeong Hur, Wayne Pickell, and Alison Proctor, who made the flight tests possible.

References

[1]Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.

[2]Sanders, C. P., DeBitetto, P. D., Feron, E., Vuong, H. F., and Levenson, N., "Hierarchical Control of Small Autonomous Helicopters," *37th IEEE Conference on Decision and Control*, Vol. 4, 1998, pp. 3629–3634.

[3]Gavrilets, V., Mettler, B., and Feron, E., "Nonlinear Model for a Small-Sized Acrobatic Helicopter," AIAA Paper 2001-4333, Aug. 2001.

[4]La Civita, M., Messner, W. C., and Kanade, T., "Modeling of Small-Scale Helicopters with Integrated First-Principles and System-Identification Techniques," *Proceedings of the 58th Forum of the American Helicopter Society*, Vol. 2, 2002, pp. 2505–2516.

[5]Mettler, B., *Identification Modeling and Characteristics of Miniature Rotorcraft*, Kluwer, Dordrecht, The Netherlands, 2002.

[6]La Civita, M., Papageorgiou, G., Messner, W. C., and Kanade, T., "Design and Flight Testing of a High Bandwidth $\mathcal{H}_\infty$ Loop Shaping Controller for a Robotic Helicopter," AIAA Paper 2002-4846, Aug. 2002.

[7]Gavrilets, V., Mettler, B., and Feron, E., "Control Logic for Automated Aerobatic Flight of Miniature Helicopter," AIAA Paper 2002-4834, Aug. 2002.

[8]La Civita, M., Papageorgiou, G., Messner, W. C., and Kanade, T., "Design and Flight Testing of a Gain-Scheduled $\mathcal{H}_\infty$ Loop Shaping Controller for Wide-Envelope Flight of a Robotic Helicopter," *Proceedings of the 2003 American Control Conference*, 2003, pp. 4195–4200.

[9]Calise, A. J., Lee, S., and Sharma, M., "Development of a Reconfigurable Flight Control Law for Tailless Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 896–902.

[10]Brinker, J., and Wise, K., "Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 903–909.

[11]Johnson, E. N., and Calise, A. J., "Limited Authority Adaptive Flight Control for Reusable Launch Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 906–913.

[12]Kim, N., Calise, A. J., Hovakimyan, N., Prasad, J., and Corban, J. E., "Adaptive Output Feedback for High-Bandwidth Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 6, 2002, pp. 993–1002.

[13]Leitner, J., Calise, A. J., and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 972–979.

[14]Rysdyk, R. T., and Calise, A. J., "Adaptive Model Inversion Flight Control for Tiltrotor Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 3, 1999, pp. 402–407.

[15]Johnson, E. N., "Limited Authority Adaptive Flight Control," Ph.D. Dissertation, School of Aerospace Engineering, Georgia Inst. of Technology, Atlanta, GA, Dec. 2000.

[16]Rysdyk, R. T., and Calise, A. J., "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Controls Systems Magazine*, Vol. 18, No. 6, 1998, pp. 14–25.

[17]Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulaion*, Wiley, New York, 2003.

[18]Johnson, E. N., and Kannan, S. K., "Nested Saturation with Guaranteed Real Poles," *American Control Conference*, Vol. 1, 2003, pp. 497–502.

[19]Kannan, S. K., and Johnson, E. N., "Adaptive Control with a Nested Saturation Reference Model," AIAA Paper 2003-5324, Aug. 2003.

[20]Hornik, K., Stinchombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.

[21]Spooner, J. T., Maggiore, M., Ordóñez, R., and Passino, K. M., *Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques*, Wiley, New York, 2002.

[22]Lewis, F. L., "Nonlinear Network Structures for Feedback Control (Survey Paper)," *Asian Journal of Control*, Vol. 1, No. 4, 1999, pp. 205–228.

[23]Calise, A. J., Hovakimyan, N., and Idan, M., "Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks," *Automatica*, Special issue on Neural Networks for Feedback Control, Vol. 37, No. 8, 2001, pp. 1201–1211.

[24]Munzinger, C., "Development of a Real-Time Flight Simulator for an Experimental Model Helicopter," M.S. Thesis, School of Aerospace Engineering, Georgia Inst. of Technology, Atlanta, July 1997.

[25]Lipp, A. M., and Prasad, J. V. R., "Synthesis of a Helicopter Nonlinear Flight Controller Using Approximate Model Inversion," *Mathematical and Computer Modelling*, Vol. 18, Aug. 1993, pp. 89–100.

[26]*Handling Qualities Requirements for Military Rotorcraft*, Aeronautical Design Standard ADS-33E, U.S. Army Aviation and Missile Command, Redstone Arsenal, AL, March 2000.

[27]Johnson, E. N., and Kannan, S. K., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," AIAA Paper 2002-4439, Aug. 2002.

[28]Johnson, E. N., and Schrage, D. P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 1, 2004, pp. 5–18.

[29]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.