JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 2, March–April 2008

Flight-Test Results of Autonomous Airplane Transitions Between Steady-Level and Hovering Flight

Eric N. Johnson,* Allen Wu,† James C. Neidhoefer,‡
Suresh K. Kannan,§ and Michael A. Turbe†
Georgia Institute of Technology, Atlanta, Georgia 30332-0150

DOI: 10.2514/1.29261

Linear systems can be used to adequately model and control an aircraft in either ideal steady-level flight or in ideal hovering flight. However, constructing a single unified system capable of adequately modeling or controlling an airplane in steady-level flight and in hovering flight, as well as during the highly nonlinear transitions between the two, requires the use of more complex systems, such as scheduled-linear, nonlinear, or stable adaptive systems. This paper discusses the use of dynamic inversion with real-time neural network adaptation as a means to provide a single adaptive controller capable of controlling a fixed-wing unmanned aircraft system in all three flight phases: steady-level flight, hovering flight, and the transitions between them. Having a single controller that can achieve and transition between steady-level and hovering flight allows utilization of the entire low-speed flight envelope, even beyond stall conditions. This method is applied to the GTEdge, an eight-foot wingspan, fixed-wing unmanned aircraft system that has been fully instrumented for autonomous flight. This paper presents data from actual flight-test experiments in which the airplane transitions from high-speed, steady-level flight into a hovering condition and then back again.

Nomenclature

$\hat{\mathbf{A}}_1$, $\hat{\mathbf{A}}_2$, $\hat{\mathbf{B}}$	=	linearized vehicle dynamics
a	=	acceleration or activation potential
$\mathbf{a}(\cdot)$, $\hat{\mathbf{a}}(\cdot)$	=	translational dynamics and estimate
b_v, b_w	=	neural network biases
$\mathbf{e}$	=	error between reference model and plant
g	=	acceleration due to gravity
$\mathbf{K}$, $\mathbf{R}$	=	inner-loop, outer-loop gain matrices
n_1	=	number of neural network inputs
n_2	=	number of neural neurons
n_3	=	number of neural network outputs
$\underset{\sim}{\boldsymbol{p}}$	=	position vector
$\underset{\sim}{\mathbf{Q}}(\cdot)$	=	attitude error angle function
$\mathbf{q} = [q_0, q_1, q_2, q_3]$	=	attitude quaternion
$\mathbf{q}(\cdot)$	=	Euler rotation to quaternion transform
$T_{i \to b}$	=	transformation from inertial frame to body frame
$u(t)$	=	control vector at time t
u, v, w	=	body axis velocity components
$\mathbf{V}$, $\mathbf{W}$, v, w	=	neural network input and output weights
$\mathbf{v}$	=	velocity
$X_{\delta_{\text{thr}}}$	=	throttle control derivative
x, $\boldsymbol{x}$	=	state variable, state vector
x_{in}, $\bar{\mathbf{x}}$	=	neural network input
z_j	=	input to jth hidden-layer neuron
α	=	angular acceleration or angle of attack
$\boldsymbol{\alpha}(\cdot)$, $\hat{\boldsymbol{\alpha}}(\cdot)$	=	attitude dynamics and estimate
$\Gamma_{\mathbf{v}}$, $\Gamma_{\mathbf{w}}$	=	neural network learning rate matrices
$\boldsymbol{\Delta}(\cdot)$	=	total function approximation error
$\Delta\Phi$	=	attitude correction
δ_f	=	throttle actuator deflection
$\boldsymbol{\delta}_m$	=	elevator, aileron, and rudder actuator deflection
δ, $\hat{\delta}$	=	actuator deflection and estimate
ζ	=	damping ratio
θ, θ_v, θ_w	=	neural network thresholds
κ	=	E-modification parameter
$\boldsymbol{\nu}_{\mathbf{ad}}$, $\bar{\boldsymbol{\nu}}_{\mathbf{ad}}$	=	adaptive element signals
$\boldsymbol{\nu}_r$	=	robustifying signal
$\boldsymbol{\sigma}$, $\boldsymbol{\sigma}'$	=	neuron sigmoidal function, gradient
$\boldsymbol{\omega}$	=	angular velocity
ω	=	natural frequency

I. Introduction

FIXED-WING aircraft with the ability to hover have the potential of providing a nearly stationary surveillance platform while maintaining the high-speed, maneuverable, long-endurance dash capabilities associated with such vehicles. There is currently significant commercial and military interest in developing such systems, which would be well suited for a variety of missions, especially in urban or other constrained environments. Military applications include the provision of persistent intelligence, surveillance, and reconnaissance (ISR) with the ability to stare to enhance target identification. In the commercial sector, a hovering mode could enhance the utility of unmanned aircraft systems (UASs) used for border patrol, traffic monitoring, and hazardous site inspection. The potential for performing tail-sitting takeoffs and landings also increases the operational domains of such aircraft.

Although the ability to transition fixed-wing UASs between steady-level and hovering flight has been demonstrated with remotely piloted vehicles, performing such maneuvers autonomously presents unique challenges. One such challenge results from the highly nonlinear nature of the actual transition between the two flight regimes. Thus, although linear systems could be used to model and control the aircraft while in either steady-level flight or ideal (nearly stationary) hovering flight, transitioning the aircraft between the two would require the use of a more complex system (such as a nonlinear, scheduled-linear, or a stable adaptive system) capable of modeling (or controlling) the nonlinearities that occur during transition.

Received 11 December 2006; revision received 25 May 2007; accepted for publication 25 May 2007.

*Lockheed Martin Assistant Professor of Avionics Integration, Aerospace Engineering Department.

†Graduate Research Assistant, Aerospace Engineering Department.

‡Research Engineer, Aerospace Engineering Department.

§Senior Research Scientist, Guided Systems Technologies, 75 Fifth Street, Northwest, Suite 336.

The specific contributions of this work include 1) the use of dynamic inversion with adaptation as a means to provide stable aircraft control in steady-level flight and hovering flight, as well as during the transition between the two regimes, with a single unified controller, and 2) detailed hover-transition flight-test results generated using a fully instrumented research unmanned air vehicle (UAV) flying in windy conditions. In this approach, a neural network is adapted in real time to account for errors in a single vehicle model that is linearized about a hover condition. Additionally, pseudocontrol hedging (PCH) allows the neural network to continue adapting when actuator nonlinearities, such as saturation, occur [1–3].

Work on hovering fixed-wing aircraft is currently being performed by several research groups. William E. Green and Paul Y. Oh at Drexel University[¶**] have performed autonomous flight tests of a fixed-wing, micro-UAS hovering in an urban environment. These experiments involved the use of a Microstrain inertial measurement unit (IMU), a PIC16F87 microcontroller, and a linear controller to maintain hover; however, transitions to and from hover were performed manually. Aerovironment's SkyTote UAS[††] [4] is a UAS with potential capabilities to hover, take off, and land vertically, and also transition into conventional horizontal fixed-wing flight. A prototype SkyTote is complete and hover testing is currently underway. Aurora Flight Sciences' GoldenEye family of UASs[‡‡§§] have been used in numerous flight tests throughout 2005 and 2006 in which autonomous transitions were made from vertical hovering flight to horizontal flight and back again. Although the GoldenEye has the ability to fly in a fixed-wing configuration, its wing is not actually "fixed," in that it has the ability to vary the angle of incidence of its wings during flight. The University of Sydney is developing a T-wing UAS [5,6] capable of vertical takeoffs and landings as well as sustained forward fixed-wing flight. To date, the T-wing has been flown in hover mode both manually and under automatic control using command augmentation system (CAS) controllers. Many other widely varying methods for autonomously guiding and controlling UASs have been developed [7–10].

Throughout this paper the term "hovering" is used to indicate a flight regime in which groundspeed is very small (or even zero). Thus, hovering in this context usually indicates a flight regime beyond wing-stall, in which engine thrust is the primary force keeping the aircraft aloft. It is recognized that it is possible for a fixed-wing airplane to fly at zero groundspeed while its wings are not stalled; however, in most cases, the high-wind conditions required to achieve this would preclude safe flight.

This paper provides a description of the GTEdge UAS and the guidance and adaptive control architectures used to successfully transition the aircraft between steady-level and hovering flight during actual flight tests. A description of the GTEdge is given first, followed by a detailed discussion of the guidance system and the neural adaptive controller. The next section gives a justification for the use of adaptive control and a description of the role of the adaptive element in this application. Finally, flight-test results are presented, followed by a discussion of limitations in the present architecture and suggestions for improving performance.

II. Flight-Test Hardware Description

The GTEdge UAS (see Fig. 1) consists of four major subsystems: 1) the baseline commercial off-the-shelf (COTS) airframe, 2) the avionics used for autonomous guidance, navigation, and control, 3) the software that runs onboard the flight control computer, and 4) the ground control station used for issuing commands to the vehicle. This section describes the specifications and role that each subsystem plays in the context of the operation of the GTEdge.

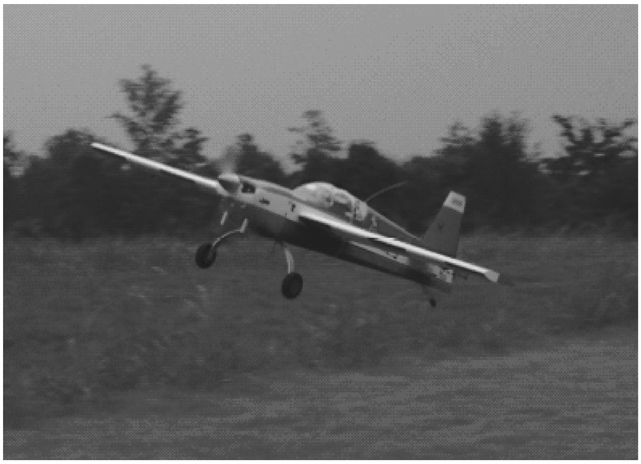


Fig. 1 GTEdge research UAS is a modified 33% scale Edge 540T, selected for its ability to carry moderate payloads and perform aggressive aerobatic maneuvers.

A. GTEdge Airframe

The GTEdge is a modified commercially available Aeroworks 33% scale Edge 540T aircraft. This baseline airframe was selected for its off-the-shelf availability and for its aerobatic capabilities, including a thrust-to-weight ratio greater than one, which allows hovering and accelerated vertical climbs. Payload requirements governed the selection of the airplane's scale. The GTEdge has the following physical characteristics: 1) a wing span of 8.75 ft, wing area of 13 ft^2, and length of 7.8 ft; 2) engine type is gasoline, Desert Aircraft DA100 100 cc engine, 9.8 hp; 3) dry weight without payload is 35 lb; 4) throttle, elevator, aileron, rudder actuated by JR8611A ultratorque digital servos; 5) actuators powered by lithium-ion batteries; 6) endurance is approximately 30 min. at steady flight of 70 ft/s.

B. FCS20-based Avionics Suite

A small, integrated guidance, navigation, and control (GN&C) hardware and software system, referred to as the Flight Control System Version 20 (FCS20) [11,12], recently developed by the Georgia Institute of Technology, is the cornerstone of the GTEdge avionics suite. This miniature computer uses a floating point digital signal processor (DSP) for high-level serial processing and a field programmable gate array (FPGA) for low-level parallel processing, along with microelectromechanical systems (MEMS) sensors.

The basic modules of the FCS20 are the EC20 processor board and the SB20 sensor/power board. The EC20 processor board (see Fig. 2) handles FCS20 processing and internal and external communications. Embedded flash memory included in the processor board allows for high data rate onboard data recording during flight. Pulse-width modulation signals used for driving vehicle actuators are generated by the FPGA.

The SB20 sensor/power board was designed to be compatible with the EC20 processor board and provides three main functions: supplying regulated and filtered power to the system, supporting onboard or external navigation sensors, and serving as an interface to external components. The eight main sensor components of the SB20 consist of three analog devices ADXR300 rate gyros, two analog devices ADXL210E two-axis $10g$ accelerometers, a mBlox global positioning system (GPS) module, and Freescale absolute and differential pressure sensors.

Other significant components of the GTEdge avionics suite include a Novatel OEM4 differential GPS and a Freewave spread

[¶] Green, W. E., and Oh, P. Y., "Autonomous Hovering of a Fixed-Wing Micro Air Vehicle," http://prism2.mem.drexel.edu/~billgreen/Publications/finalGreenIcra2006.pdf, Drexel Univ., Philadelphia (cited Dec. 2006).

[**] Green, W. E., and Oh, P. Y., "Micro Air Vehicle to Fly in Caves, Tunnels, and Forests," http://www.pp..drexel.edu/~weg22/fwHovering/fixedWingHovering.html, Drexel Univ., Philadelphia (cited Dec. 2006).

[††] U.S. Air Force Research Laboratory Web site, http://www.afrl.af.mil/accomprpt/may04/accompmay04.asp (cited December 2006).

[‡‡] Aurora Flight Sciences Web site http://www.aurora.aero/GE50/index.html (cited December 2006).

[§§] International Online Defence Magazine, http://www.defense-update.-com/products/g/goldeneye50.htm (cited December 2006).

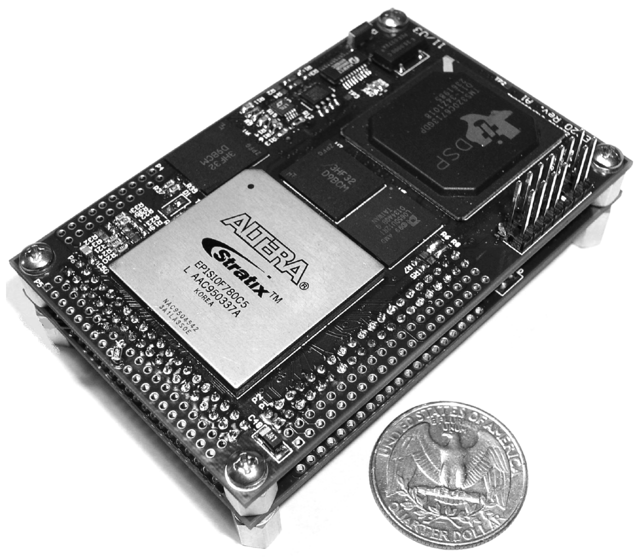


Fig. 2 FCS20 with EC20 processor board on top.

spectrum transceiver for communications with the ground control station. A diagram of the complete system can be seen in Fig. 3.

C. Extended-Kalman-Filter-Based Navigation System

An integral component of the FCS20 is the 16-state extended Kalman filter (EKF) which uses data from the sensors on the SB20 processor board to generate a navigation solution that closely estimates the state of the system [13,14]. The EKF serves several important functions, including 1) estimating the orientation of the system from accelerations and angular rates, 2) removing process and measurement noise from the measurements, and 3) providing state estimates at 100 Hz, even though the GPS updates at a slower rate of 10 Hz. Similar EKF implementations can be found for comparison [15].

D. Ground Control Station

The ground control station (GCS) is a laptop that communicates with the GTEdge over a wireless serial link. The GTEdge sends a primary message packet containing information needed for operation down to the GCS at 10 Hz, and another message packet containing less vital information once every second. These message packets provide the GCS with data used to display vehicle information to the GCS operator and are also recorded on the GCS to provide redundant data recording to supplement the data recorded by the GTEdge onboard computer. GCS operators can also issue commands such as flight plans or requests for specific data by sending the appropriate message type from the GCS to the GTEdge.

III. Guidance and Control Systems

An adaptive neural net-based controller [2] developed and flight tested on other UASs [16] is used to control the GTEdge in steady-level flight, hovering flight, and the transition between. An overview of the controller architecture and the motivation for its use in this application are included here. A proof of the underlying theory is also available [17].

In this implementation, the controller consists of an outer loop for tracking translational states and an inner loop responsible for vehicle attitude dynamics. Both the inner and outer loops are organized in similar fashion into four primary components: a reference model, an approximate inversion, a linear proportional plus derivative (PD) compensator, and a hedging block, as shown in Fig. 4. Each loop uses feedback linearization and, more specifically, dynamic inversion, as the control strategy. The reference model is nonlinear and selected to impose a desired closed-loop response and also impose limits on the evolution of states. Any step changes in the external command thus appear as continuous signals to the linear PD compensator.

Dynamic inversion, when used alone, requires accurate system models for all flight regimes and these models can be costly and difficult to obtain [3]. However, in this application, a simple model of the vehicle, linearized about a hover condition, is used in the approximate dynamic inversion, and a neural network is trained online to correct for the modeling errors. In this approach, certain nonlinear effects (such as actuator saturation) can create difficulties for the adaptive element [1,2]. Thus, a technique called pseudocontrol hedging (PCH) has been implemented to keep the network from continuously trying to adapt to these effects, by adjusting the reference model with a hedging signal.

In the cascaded inner- and outer-loop architecture shown in Fig. 4, the inner loop appears to the outer loop as an actuator that generates translational accelerations by means of a commanded attitude. Thus, in the case of a fixed-wing aircraft, the outer loop generates a throttle command and an attitude augmentation to be added to the external attitude command for the inner loop to achieve the desired vehicle position and velocity. The inner loop combines the desired attitude from the outer loop with the commanded attitude from the trajectory generator to compute the appropriate moment actuator deflections. In Fig. 4, $\boldsymbol{a}$ and $\boldsymbol{\alpha}$ are used to denote linear and angular acceleration, respectively.

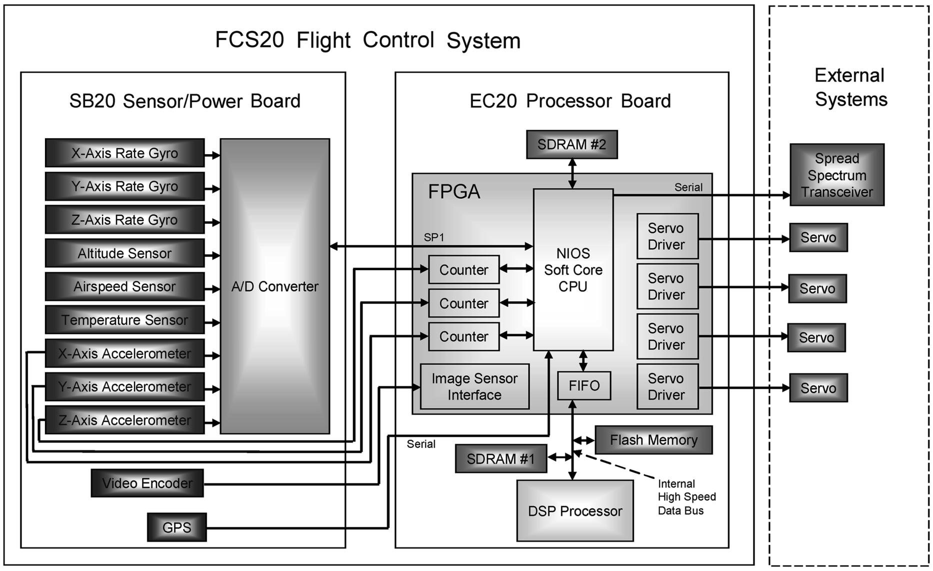


Fig. 3 FCS20 data flow diagram.

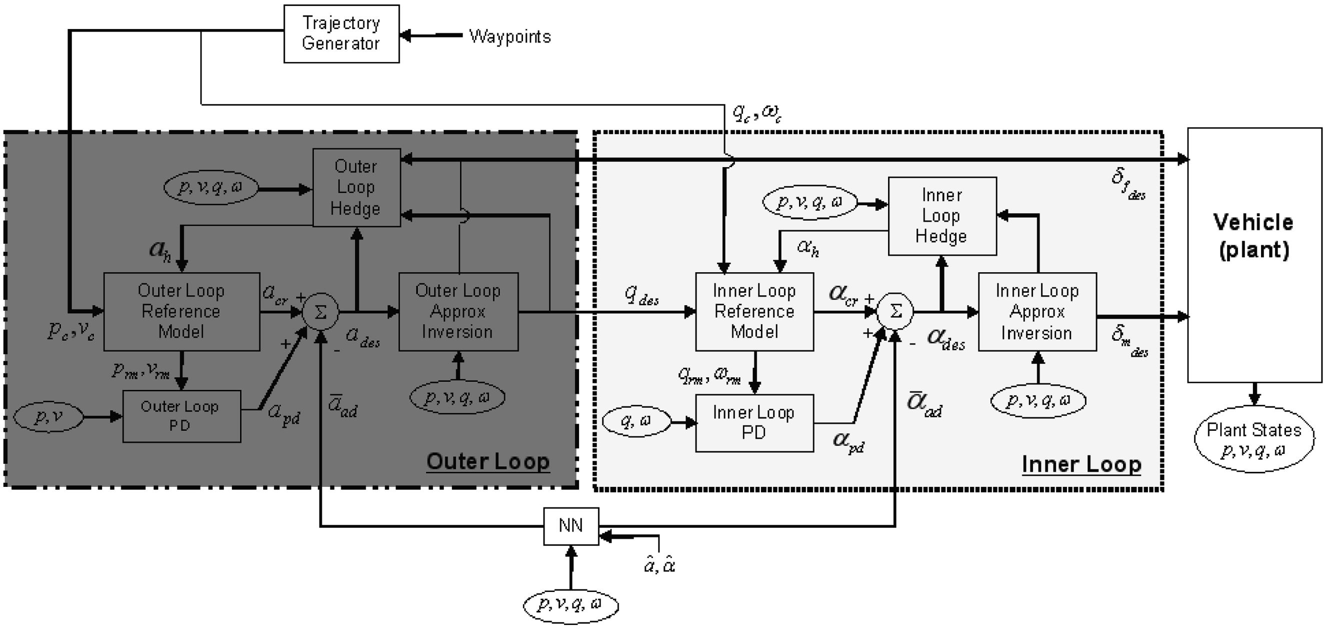


Fig. 4 Architecture used for control of a fixed-wing UAS during steady-level and hovering flight, as well as the transition between them.

A. Need for Adaptation

As mentioned earlier, although linear systems can be used to adequately model and control an aircraft either in steady-level flight or in ideal (nearly stationary) hovering flight, the same cannot be said for the highly nonlinear transition between these two regimes. Accurately modeling or controlling an aircraft during this transition would require the use of systems capable of dealing with these nonlinearities, such as nonlinear systems, multiple scheduled-linear systems, or stable adaptive systems.

In this work, a stable adaptive controller was used to address this challenge; the adaptive element within the controller was a single-hidden-layer perceptron neural network (NN) used to model the error between the linear hover model and the actual nonlinear aircraft system. Using an adaptive controller offers potential advantages over both nonlinear systems and multiple scheduled-linear systems. For example, traditional methods (such as Lyapunov synthesis [18]) for developing a stable nonlinear (nonadaptive) controller can prove difficult in nonideal applications; likewise, the cost of developing the multiple linear systems required for scheduling can preclude the usefulness of that approach.

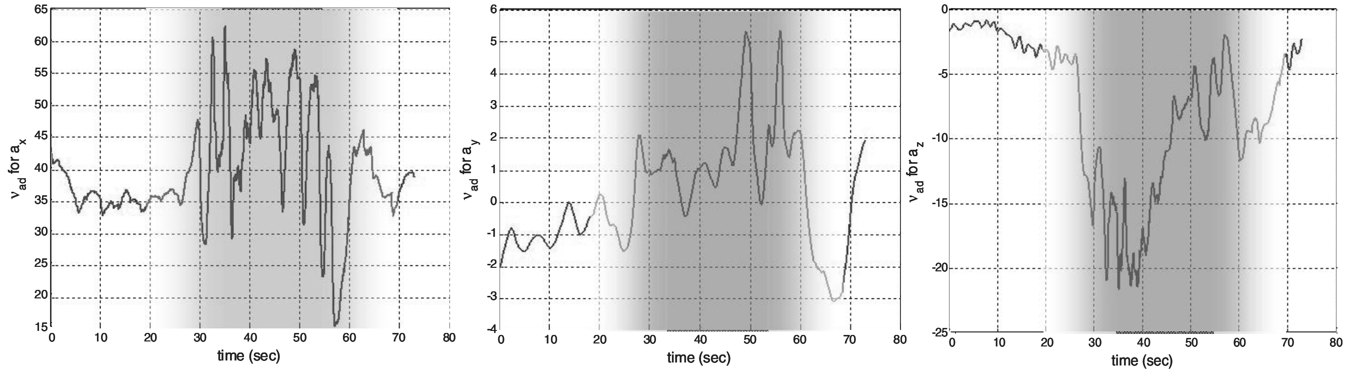
Based on the controller formulation used in this work, certain characteristics of the neural network output during flight can be predicted. In particular, because during both steady-level and hovering flight, the aircraft can be adequately represented with a linear model; the NN output (which represents the error between a linear model and the actual model) should have relatively constant characteristics during each of these two flight phases.

Figures 5 and 6 show the NN output during an actual flight test (more data from the same test is presented in Sec. IV). Because this data was taken in actual flight conditions (not in an ideal simulation), the influence of process noise (wind, turbulence, etc.), measurement noise, lag, and other real-world effects can clearly be seen in the overall noise level of the data. In addition, these same real-world effects make it much more difficult for the closed-loop aircraft to achieve perfect "steady-level" or "stationary hovering" flight. However, the aforementioned trend is still clearly visible in the data, particularly in Figs. 5a, 6a, and 6c: namely, the NN output exhibits relatively constant characteristics (e.g., amplitude and frequency) during steady-level flight (the nonshaded areas in the plots) which are different than the relatively constant characteristics exhibited during hovering flight (the shaded areas in the plots) and the flight phases; including the gray transitions, which approximately cover the transitions between steady-level and hovering flight, are readily identifiable in the data. It is also noted here that in each experiment the neural network weights were initialized to zero so that, during the experiments, significant and meaningful adaptation of the weights occurred.

B. Controller Synthesis

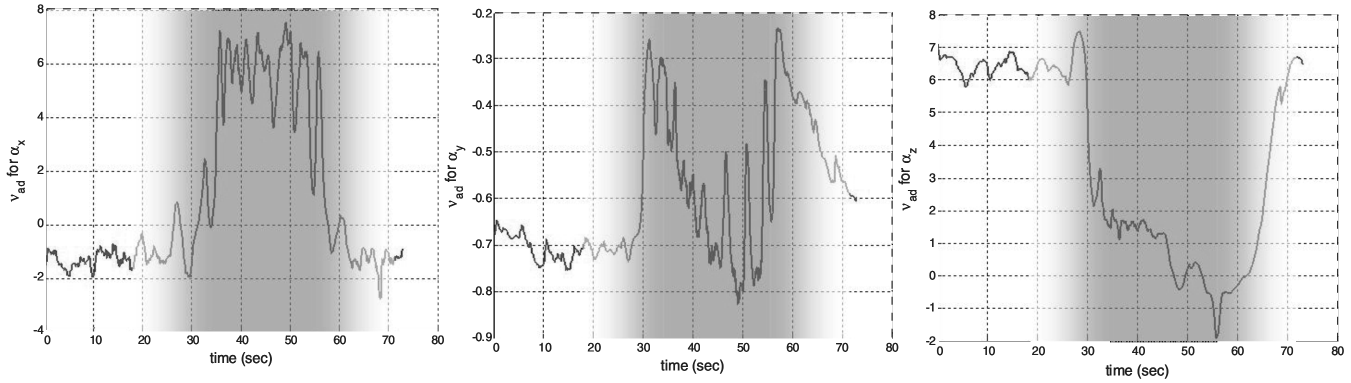
The aircraft dynamics are described with the following nonlinear equations [2]:

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



a) x axis b) y axis c) z axis

Fig. 5 Output from the neural network adaptive element for translational accelerations of the outer loop in the slow transition, expressed as components of the body-fixed axes; vertical axis is in units of $\mathrm{ft/s^2}$.



a) x axis **b) y axis** **c) z axis**

Fig. 6 Output from the neural network adaptive element for angular accelerations of the inner loop in the slow transition, expressed as components of the body-fixed axes; vertical axis is in units of rad/s^2.

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

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

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

in which quaternions are used to express vehicle attitude.

The control architecture is governed by nested outer and inner loops that handle the translational dynamics [Eq. (2)] and the attitude dynamics [Eq. (4)], respectively. The state vector of the vehicle is

$$\boldsymbol{x} = [\boldsymbol{p}^T \;\; \mathbf{v}^T \;\; \mathbf{q}^T \;\; \boldsymbol{\omega}^T]^T \tag{5}$$

The vector of control signals is written as:

$$\boldsymbol{\delta} = \left[\boldsymbol{\delta}_f^T \;\; \boldsymbol{\delta}_m^T\right]^T \tag{6}$$

where $\boldsymbol{\delta}_f$ represents the propeller thrust, and $\boldsymbol{\delta}_m$ represents the elevator, aileron, and rudder aerodynamic moment actuators. The actuator dynamics are unknown but are assumed to be asymptotically stable.

A transformation is introduced to provide approximate feedback linearization. This transformation is [2]

$$\begin{bmatrix} \mathbf{a_{des}} \\ \boldsymbol{\alpha}_{\mathbf{des}} \end{bmatrix} = \begin{bmatrix} \hat{a}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{q_{des}}, \boldsymbol{\delta}_{f_{\mathrm{des}}}, \hat{\boldsymbol{\delta}}_m) \\ \hat{\alpha}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{m_{\mathrm{des}}}) \end{bmatrix} \tag{7}$$

where $\hat{\mathbf{a}}(\cdot)$ and $\hat{\boldsymbol{\alpha}}(\cdot)$ are mappings selected to approximate the actual translational and rotational accelerations $\mathbf{a}(\cdot)$ and $\boldsymbol{\alpha}(\cdot)$ of the vehicle. Here, $\mathbf{a_{des}}$ and $\boldsymbol{\alpha}_{\mathbf{des}}$ are commonly referred to as the pseudocontrol signals and are analogous to desired translational and angular accelerations. The control inputs and attitude needed to produce the desired pseudocontrols are $\boldsymbol{\delta}_{f_{\mathrm{des}}}$, $\boldsymbol{\delta}_{m_{\mathrm{des}}}$, and $\mathbf{q_{des}}$, respectively, and $\hat{\boldsymbol{\delta}}_f$ and $\hat{\boldsymbol{\delta}}_m$ are estimated actuator positions. If the acceleration approximations are chosen to be invertible, expressions for the desired control inputs and attitude can be calculated as

$$\begin{bmatrix} \boldsymbol{\delta}_{f_{\mathrm{des}}} \\ \mathbf{q_{des}} \end{bmatrix} = \hat{\mathbf{a}}^{-1}(\boldsymbol{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{a_{des}}, \hat{\boldsymbol{\delta}}_m) \tag{8}$$

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

This approximate inversion provides the following closed-loop dynamics:

$$\dot{\mathbf{v}} = \mathbf{a_{des}} + \bar{\boldsymbol{\Delta}}_{\mathbf{a}}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) - \mathbf{a_h} \tag{10}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}_{\mathbf{des}} + \bar{\boldsymbol{\Delta}}_{\alpha}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) - \boldsymbol{\alpha}_{\mathbf{h}} \tag{11}$$

in which

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

represents the acceleration that is not cancelled out exactly in the feedback linearization due to errors in the model inversion. Furthermore, actuator saturation can introduce limitations on the achievable pseudocontrol, and so the desired pseudocontrol may be unrealizable; this effect introduces the $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ terms in the closed-loop dynamical equations. System stabilization can be achieved by choosing the pseudocontrols in the following manner [2]:

$$\mathbf{a}_{\mathbf{des}} = \mathbf{a_{cr}} + \mathbf{a_{pd}} - \bar{\mathbf{a}}_{\mathbf{ad}} \tag{12}$$

$$\boldsymbol{\alpha}_{\mathbf{des}} = \boldsymbol{\alpha}_{\mathbf{cr}} + \boldsymbol{\alpha}_{\mathbf{pd}} - \bar{\boldsymbol{\alpha}}_{\mathbf{ad}} \tag{13}$$

in which $\mathbf{a_{cr}}$ and $\boldsymbol{\alpha}_{\mathbf{cr}}$ are output from the vehicle dynamic reference models, $\mathbf{a_{pd}}$ and $\boldsymbol{\alpha}_{\mathbf{pd}}$ are output from the PD compensator, and $\bar{\mathbf{a}}_{\mathbf{ad}}$ and $\bar{\boldsymbol{\alpha}}_{\mathbf{ad}}$ are output from the adaptive element designed to cancel the modeling error $\bar{\Delta}$.

The following sections outline the determination of the signals used in the computation of the pseudocontrols. The proper selection of these signals guarantees the boundedness of the error between the plant output and the reference model output given by the vector

$$\boldsymbol{e} = \begin{bmatrix} \mathbf{p}_r - \mathbf{p} \\ \mathbf{v}_r - \mathbf{v} \\ \tilde{\mathbf{Q}}(\mathbf{q}_r, \mathbf{q}) \\ \boldsymbol{\omega}_r - \boldsymbol{\omega} \end{bmatrix} \tag{14}$$

where $\tilde{\mathbf{Q}}$ is a function that, given two quaternions, results in an error angle vector with three components, as follows:

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

The tracking error dynamics can be found by directly differentiating Eq. (14) to get

$$\dot{\mathbf{e}} = \begin{bmatrix} \mathbf{v}_r - \mathbf{v} \\ \dot{\mathbf{v}}_r - \dot{\mathbf{v}} \\ \boldsymbol{\omega}_r - \boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}}_r - \dot{\boldsymbol{\omega}} \end{bmatrix} \tag{15}$$

Furthermore, the linear PD compensator has the following form:

$$\begin{bmatrix}\mathbf{a}_{\mathbf{pd}}\\ \boldsymbol{\alpha}_{\mathbf{pd}}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_p & \mathbf{R}_d & 0 & 0\\ 0 & 0 & \mathbf{K}_p & \mathbf{K}_d\end{bmatrix}\boldsymbol{e}$$

C. Reference Model and Pseudocontrol Hedging

Care must be taken in designing the reference model to ensure that the effects which introduce $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ in Eqs. (10) and (11) are not present in the tracking error dynamics. Otherwise, the reference model will continue to generate commands as if there were no actuator saturation, and the adaptive element would try to correct for this discrepancy. This can be avoided through the use of PCH in the reference model dynamics as follows [2]:

$$\dot{\mathbf{v}}_r = \mathbf{a}_{\mathbf{cr}}(\mathbf{p}_r, \mathbf{v}_r, \mathbf{p}_c, \mathbf{v}_c) - \mathbf{a}_h \tag{16}$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{\mathbf{cr}}(\mathbf{q}_r, \boldsymbol{\omega}_r, \mathbf{q}_c \oplus \mathbf{q}_{\mathbf{des}}, \boldsymbol{\omega}_c) - \boldsymbol{\alpha}_h \tag{17}$$

where the subscript c refers to commands and r to the reference model, and the expression $\mathbf{q}_c \oplus \mathbf{q}_{\mathbf{des}}$ represents the combination of two quaternion vectors through quaternion multiplication. The signals $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$, which represent the error between the commanded and realized pseudocontrol, are given as follows:

$$\mathbf{a}_h = \hat{\mathbf{a}}(\boldsymbol{x}, \mathbf{q}_{\mathbf{des}}, \delta_{f_{\text{des}}}, \hat{\delta}_m) - \hat{\mathbf{a}}(\boldsymbol{x}, \hat{\delta}_f, \hat{\delta}_m) = \mathbf{a}_{\mathbf{des}} - \hat{\mathbf{a}}(\boldsymbol{x}, \hat{\delta}_f, \hat{\delta}_m) \tag{18}$$

$$\boldsymbol{\alpha}_h = \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\delta}_f) - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\delta}_f, \hat{\delta}_m) = \boldsymbol{\alpha}_{\mathbf{des}} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}, \hat{\delta}_f, \hat{\delta}_m) \tag{19}$$

As can be seen from the reference model dynamics, the hedging signals shift the reference models by an estimate of the amount the plant did not move due to the saturation of any actuators. This means that when the estimated actual position of the actuators is the same as the desired positions of the actuator (i.e., the actuator is not saturated), then the $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ terms vanish, and the reference model is unaffected. In the presence of saturated actuators, however, the $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ terms are nonzero and therefore are subtracted from the reference model dynamics to effectively hide the error due to saturation in the error dynamics. The pseudocontrol hedging concept is relevant to this application as the actuators often saturate during the hover-transition maneuvers and during hovering flight. The utility of PCH for enabling valid adaptation in these types of conditions has been demonstrated [19–21].

Substituting Eqs. (16–19) into the error dynamics of Eq. (15), it can be shown that the error dynamics become [2]

$$\dot{\mathbf{e}} = \mathbf{A}\boldsymbol{e} + \mathbf{B}[\bar{\nu}_{\text{ad}} - \bar{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}})] \qquad \bar{\nu}_{\text{ad}} = \begin{bmatrix}\bar{\mathbf{a}}_{\mathbf{ad}}\\ \bar{\boldsymbol{\alpha}}_{\mathbf{ad}}\end{bmatrix}$$
$$\mathbf{A} = \begin{bmatrix} 0 & I & 0 & 0\\ -\mathbf{R}_p & -\mathbf{R}_d & 0 & 0\\ 0 & 0 & 0 & I\\ 0 & 0 & -\mathbf{K}_p & -\mathbf{K}_d\end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix}0 & 0\\ I & 0\\ 0 & 0\\ 0 & I\end{bmatrix} \tag{20}$$

Thus, the linear gain matrices must be chosen such that $\mathbf{A}$ is Hurwitz, and $\bar{\nu}_{\text{ad}}$ needs to be chosen to cancel the effect of $\bar{\boldsymbol{\Delta}}$.

In this implementation, the following reference model was used:

$$\mathbf{a}_{\mathbf{cr}} = \mathbf{R}_p(\mathbf{p}_c - \mathbf{p}_r) + \mathbf{R}_d(\mathbf{v}_c - \mathbf{v}_r) \tag{21}$$

$$\dot{\mathbf{v}}_r = \mathbf{a}_{\mathbf{cr}} - \mathbf{a}_h \tag{22}$$

$$\boldsymbol{\alpha}_{\mathbf{cr}} = \mathbf{K}_p[\tilde{\mathbf{Q}}(\mathbf{q}_c \oplus \mathbf{q}_{\mathbf{des}}, \mathbf{q}_r)] + \mathbf{K}_d(\boldsymbol{\omega}_c - \boldsymbol{\omega}_r) \tag{23}$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{\mathbf{cr}} - \boldsymbol{\alpha}_h \tag{24}$$

The four gain matrices ($\mathbf{R}_{p,d}$ and $\mathbf{K}_{p,d}$) in the preceding equations were chosen to result in desirable closed-loop pole placement and are the same as those in the PD compensator (specific values of $\mathbf{R}_{p,d}$ and $\mathbf{K}_{p,d}$ are given in Table 1). Formulas for the gain values are given as [2]

$$R_p = \frac{\omega_o^2\omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \tag{25}$$

$$R_d = 2\frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \tag{26}$$

$$K_p = \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2 \tag{27}$$

$$K_d = 2\zeta_i\omega_i + 2\zeta_o\omega_o \tag{28}$$

where $\omega_{i,o}$ represent the natural frequencies for the inner and outer loops, and $\zeta_{i,o}$ represent the damping ratios. These scalar gains are placed in the diagonal gain matrices shown in Eqs. (21) and (23).

Table 1 Controller design parameters used in this implementation

Parameter	Units	Value	Description
$R_{p_{x,y,z}}$	$1/(\text{s}^2)$	0.49, 0.3623, 0.24	inner-loop proportional gain
$R_{d_{x,y,z}}$	1/s	1.4, 1.0145, 0.8	inner-loop derivative gain
$K_{p_{x,y,z}}$	$1/(\text{s}^2)$	6.25, 8.64, 17.25	outer-loop proportional gain
$K_{d_{x,y,z}}$	1/s	5.0, 4.8, 7.0	outer-loop proportional gain
K_r	N/A	0.0	gain for robustifying term
n_1	N/A	18	number neural network inputs
n_2	N/A	5	number of neurons
n_3	N/A	6	number of neural network outputs
b_v	N/A	0.5	neural network inner bias
b_w	N/A	0.5	neural network outer bias
a_j	N/A	0.2, 0.4, 0.6, 0.8, 1.0 for $j = 1, \ldots, 5$	activation potentials
$\Gamma_{\mathbf{w}}$	N/A	Diag(1.0, 1.0, 1.0, 1.0, 1.0, 1.0)	neural network outer learning rate
$\Gamma_{\mathbf{v}}$	N/A	5.0	neural network inner learning rate
κ	N/A	0.1	weight for damping term in adaptation law
$\hat{\mathbf{A}}_{\mathbf{1}}$	1/s	diag(0.0, 0.0, 0.0)	matrix for inner-loop reference model
$\hat{\mathbf{A}}_{\mathbf{2}}$	rad/ft · s	diag(0.0, 0.0, 0.0)	matrix for inner-loop reference model
$\hat{\mathbf{B}}$	$\text{rad}/(\text{s}^2)$	diag(50.0, 15.0, 50.0)	control effectiveness; inner-loop reference model
$X_{\delta_{\text{thr}}}$	$\text{ft}/(\text{s}^2)$	45.0	throttle control derivative
$\delta_{f_{\text{trim}}}$	%maximum/100	0.2344	throttle trim value for hover

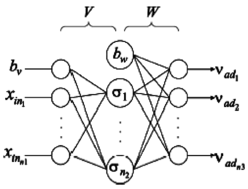


Fig. 7 Neural network with a single hidden layer.

Specific values for parameters used in this implementation are given in Table 1.

D. Adaptive Neural Network

A single-hidden-layer perceptron neural network (NN) is used to approximate the modeling error. As shown in Fig. 7, the number of inputs, neurons, and outputs are given by n_1, n_2, and n_3, respectively. In the NN, the input-output relationship is given by

$$\nu_{\mathrm{ad}_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j(z_j) \tag{29}$$

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

$$\sigma_j(z_j) = 1/(1 + e^{-az_j}) \tag{30}$$

with the neuronal activation potential given as a. For the jth neuron, the input is

$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{\mathrm{in}_i} \tag{31}$$

where b_v is the inner-layer bias, θ_{v_j} is the jth threshold, v_{ij} represents the weights of the inner layer, and x_{in_i} represents the NN inputs. Both the inner- and outer-layer biases are chosen based on experience. For convenience, the following NN weight matrices are introduced:

$$\mathbf{V} = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_2,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{V} & 0 \\ 0 & \mathbf{W} \end{bmatrix}$$

Furthermore, the following vectors are defined

$$\boldsymbol{\sigma}^T(\mathbf{z}) = [b_w \ \ \sigma(z_1) \ \ \cdots \ \ \sigma(z_{n_2})] \qquad \bar{\mathbf{x}}^T = \begin{bmatrix} b_v & \mathbf{x}_{\mathrm{in}}^T \end{bmatrix}$$

$$\mathbf{x}_{\mathrm{in}}^T = \begin{bmatrix} \mathbf{x}_c^T & \mathbf{e}_r^T & \mathbf{e}^T & \nu_{\mathrm{ad}}^T & \|\mathbf{Z}\|_F \end{bmatrix}$$

where $\mathbf{e}_r$ is the error between the output of the reference model and the command.

Using the aforementioned notation, the output of the neural network is given by the following equations:

$$\mathbf{r} = (\mathbf{e}^T \mathbf{P} \mathbf{B})^T \qquad \bar{\boldsymbol{\nu}}_{\mathbf{ad}} = \boldsymbol{\nu}_{\mathbf{ad}} + \boldsymbol{\nu}_r \qquad \boldsymbol{\nu}_{\mathbf{ad}} = \mathbf{W}^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}})$$

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

where K_r is the gain on the robustifying term. It is included here for completeness of the equations, though for this application no robustifying term was needed and K_r was set to zero. The matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \frac{1}{\frac{1}{4} n_2 + b_w^2}$$

$$\mathbf{P}_1 = \begin{bmatrix} R_p^2 + \frac{1}{2} R_p R_d^2 & \frac{1}{2} R_p R_d \\ \frac{1}{2} R_p R_d & R_p \end{bmatrix} > 0$$

$$\mathbf{P}_2 = \begin{bmatrix} K_p^2 + \frac{1}{2} K_p K_d^2 & \frac{1}{2} K_p K_d \\ \frac{1}{2} K_p K_d & K_p \end{bmatrix} > 0$$

In the preceding equations, the NN weight matrices are updated according to the adaptation laws

$$\dot{\mathbf{W}} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}^T \bar{\mathbf{x}}) \mathbf{r}^T + \kappa \|\mathbf{e}\| \mathbf{W}] \Gamma_{\mathbf{w}} \tag{32}$$

$$\dot{\mathbf{V}} = -\Gamma_{\mathbf{v}}[\bar{\mathbf{x}}(\mathbf{r}^T \mathbf{W}^T \boldsymbol{\sigma}') + \kappa \|\mathbf{e}\| \mathbf{V}] \tag{33}$$

where $\Gamma_{\mathbf{w}}$ and $\Gamma_{\mathbf{v}}$ are positive definite matrices, and κ is a positive scalar. The $\boldsymbol{\sigma}'$ matrix is the Jacobian of the $\boldsymbol{\sigma}$ vector, and the $\mathbf{B}$ matrix comes from the error dynamics in Eq. (20). The $\bar{Z}$ value is a bound on the norm of the $\mathbf{Z}$ matrix corresponding to the ideal NN weighting matrices in accordance with the universal approximation property.

It has been proven [2] that Lyapunov analysis carried over the error dynamics in Eq. (20) using these weight adaptation rules given in Eqs. (32) and (33) will result in the ultimate boundedness of the tracking error $\mathbf{e}$ and the NN weights with the bounds being explicitly defined in terms of system dynamics, and the Lyapunov equation matrices [2].

E. Approximate Dynamic Model

The method of dynamic inversion used herein requires an invertible, approximate linear model of the vehicle. This approximate model is split into an inner-loop reference model and an outer-loop reference model. The adaptive neural network is then used to compensate for the inaccuracies in this approximation. The external commands provided to the control algorithm contain commanded pitch angle as a function of speed, and the control effectiveness $\hat{\mathbf{B}}$ is scaled based on speed to reflect the reduced control authority of the control surfaces in hover. Neglecting coupling between the attitude and translational dynamics, the attitude dynamics for the inner-loop reference model linearized about a stationary hover are as follows:

$$\boldsymbol{\alpha}_{\mathbf{des}} = \hat{\mathbf{A}}_1 \boldsymbol{\omega} + \hat{\mathbf{A}}_2 \mathbf{v} + \hat{\mathbf{B}}(\boldsymbol{\delta}_{m_{\mathrm{des}}} - \boldsymbol{\delta}_{m_{\mathrm{trim}}}) \tag{34}$$

where $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$ describe the attitude and translational dynamics, and $\boldsymbol{\delta}_{m_{\mathrm{trim}}}$ is the trim control vector [2]. For this application, the state Jacobian of the inner-loop reference model ($\hat{\mathbf{A}}_1$, $\hat{\mathbf{A}}_2$) was set to zero, which implies that moments come directly from control actuations (i.e., elevator results in pitching moment, etc.). $\hat{\mathbf{B}}$ was tuned manually during flight tests to match the control effectiveness of the approximate model with that of the actual system. This equation can be solved for the moment control vector $\boldsymbol{\delta}_{m_{\mathrm{des}}}$ if the control effectiveness $\hat{\mathbf{B}}$ is invertible. To obtain the translational dynamics, the vehicle is modeled as a point mass with a thrust vector that can be directed in any direction. The dynamics of the model are as follows:

$$\mathbf{a}_{\mathbf{des}_b} = \begin{bmatrix} X_{\delta_{\mathrm{thr}}} \\ 0 \\ 0 \end{bmatrix} (\delta_{f_{\mathrm{des}}} - \delta_{f_{\mathrm{trim}}}) + T_{i \to b} \boldsymbol{g} \tag{35}$$

where $X_{\delta_{\mathrm{thr}}}$ is the control derivative for horizontal axis acceleration and $\boldsymbol{g}$ is the gravity vector. The desired specific force along the x-body axis is

$$f_{\mathrm{sf}} = (\mathbf{a}_{\mathrm{des}} - T_{i \to b} \boldsymbol{g})_1 \qquad \delta_{f_{\mathrm{des}}} = f_{\mathrm{sf}} / X_{\delta_{\mathrm{thr}}} + \delta_{f_{\mathrm{trim}}} \tag{36}$$

The attitude changes necessary to align the thrust vector in the proper direction are given by the small-angle corrections, as shown in Fig. 8 and the following equations [2]:

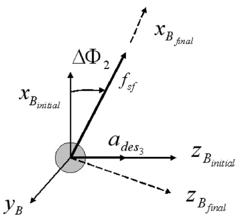


Fig. 8 Point mass model is used for the outer-loop approximate inversion. This diagram illustrates a small-angle correction for a pitch attitude change.

$$\Delta\Phi_1 = 0 \tag{37}$$

$$\Delta\Phi_2 = -a_{\mathrm{des}_3}/f_{\mathrm{sf}} \tag{38}$$

$$\Delta\Phi_3 = a_{\mathrm{des}_2}/f_{\mathrm{sf}} \tag{39}$$

Here, $\Delta\Phi_1$ is zero, because changes in roll attitude have no effect on the direction of the thrust vector.

F. Hover-Transition Guidance Logic

For transitions between steady-level and hovering flight, two guidance systems (the trajectory generator in Fig. 4) have been tested. The first scheme involved a linear ramping of the commanded pitch angle from its value in trim forward flight to a commanded pitch angle of 90 deg while simultaneously decreasing (again, linearly) the commanded velocity over a period of 9 s. The second scheme involved a step change in the commanded pitch angle from trim to 120 deg and then back to 90 deg as the aircraft decelerated to zero forward velocity. This faster transition also involved a linear decrease in commanded velocity over a period of 5 s. The transitions from stationary hover back to forward flight were essentially the inverse of these maneuvers. In both of these guidance schemes, the lower throttle limit was increased as rudder and/or elevator neared saturation; this change was made based on previous flight-test experience. Results for each of these transition schemes are shown in Sec. IV.

IV. Flight-Test Results

To date, eight successful transitions from forward horizontal flight to hovering flight and then back to forward flight under autonomous control have been recorded using the GTEdge while operating in varying conditions. The first of these successful flights was achieved on 29 July 2005. Flight performance verification was initially carried out for this vehicle by having the airplane first perform circular orbits at a moderate groundspeed of 80 ft/s and then perform successive orbits at increasingly slower groundspeeds, in decrements of 10 ft/s, to ensure proper operation of the controller throughout the flight envelope.

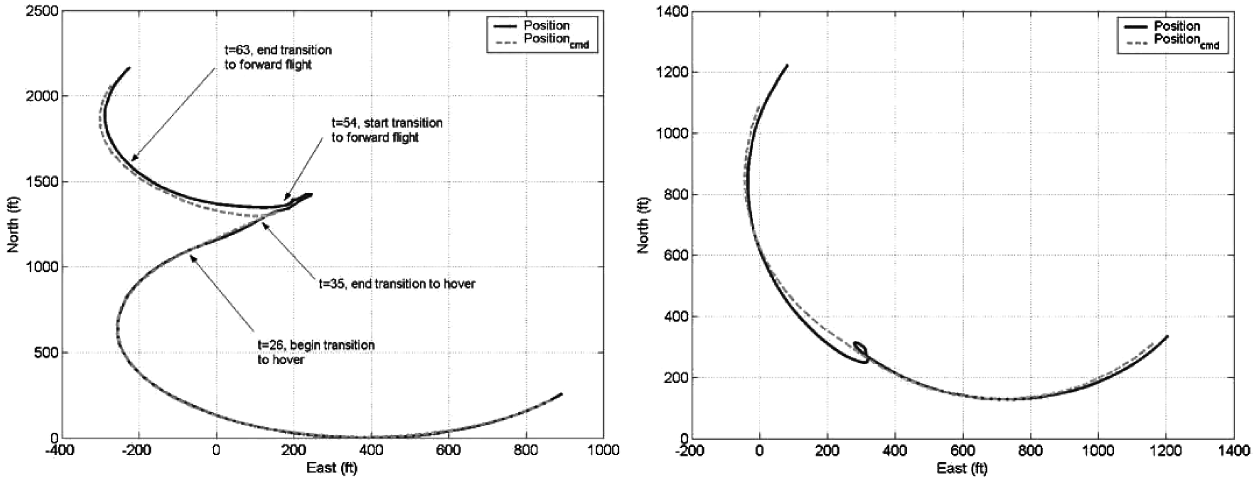
This section presents data sets from two separate flight experiments, recorded on separate days, in which the GTEdge performed these transitions under dissimilar flight conditions. In the first set, the flight test was initiated with the airplane performing a circular orbit at 80 ft/s and then transitioning into a hover by ramping the groundspeed down to 0 ft/s over a period of 9 s while simultaneously ramping up the commanded pitch angle. The reverse, transition to forward flight, was accomplished by a similar ramp up in speed to 80 ft/s and a corresponding decrease in commanded pitch angle. Hovering flight was sustained for 19 s in this particular instance.

In the second set of recorded data, the flight test was similarly initiated with the airplane performing a circular orbit at 80 ft/s. However, the transition to hovering flight was performed much more rapidly, with the groundspeed ramped down to 0 ft/s over a duration of 5 s instead of 9 s. Hover was held for 10 s, and then the aircraft transitioned back to forward flight at the same slower rate as in the first data set. We will refer to the first data set as the "slower transition" and the second data set as the "faster transition."

Air data was recorded during the faster transition but was unavailable during the slower-transition flight. Thus, although wind direction for the first data set was observed to be toward the southwest, the magnitude was unknown. Using the air data from the faster transition, it was found that the wind velocity during the second test was approximately 30 ft/s pointing in the southeast direction. During both flights, the transition to hovering flight was initiated during the upwind leg of the circular orbit.

Figure 9 shows the commanded and estimated ground tracks, whereas Figs. 10 and 11 show the three-dimensional trajectory of the aircraft throughout the maneuvers. By comparing the shapes of the position and velocity curves in Figs. 11a and 12a, it can be seen that in the slower transition, the GTEdge overshoots the commanded hovering position and gradually corrects for this error. The north and east velocities in Fig. 12a change sign, as would be expected to correct for the overshoot. These figures also show that, in the faster transition, the aircraft undershoots the commanded position. In this scenario, however, the command to return to forward flight was issued before the controller could begin compensating for the position error.

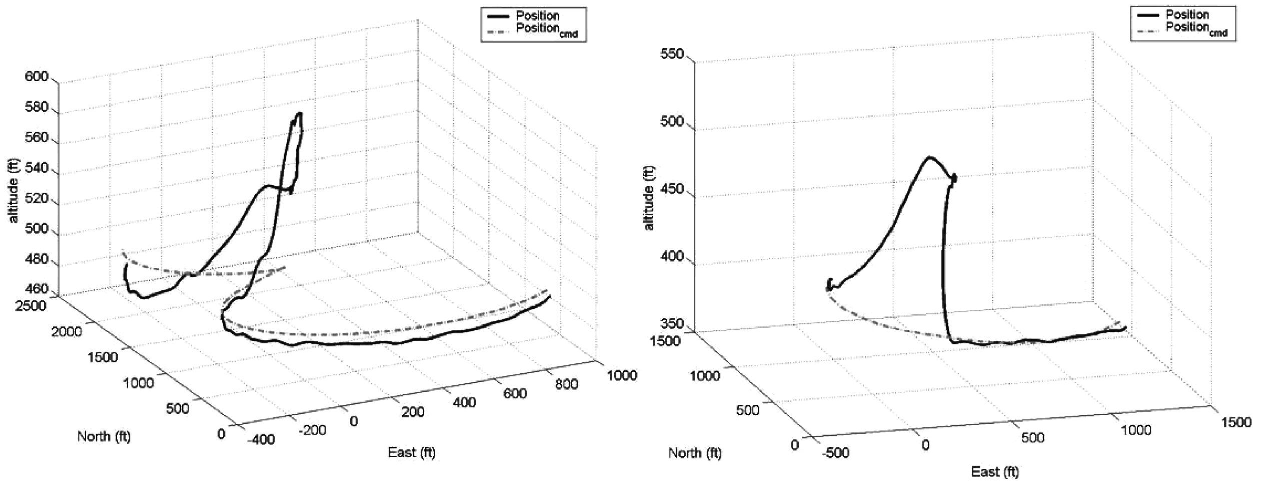
Both the slower transition and the faster transition exhibit a climb in altitude of about 100 ft during the transition to hover, because the aircraft has to pull up to attain the near vertical attitude required for low-speed flight. The maximum altitude errors for the slower and faster transitions are 95.8 and 133.4 ft, respectively. During the hover



a) Slower transition

b) Faster transition

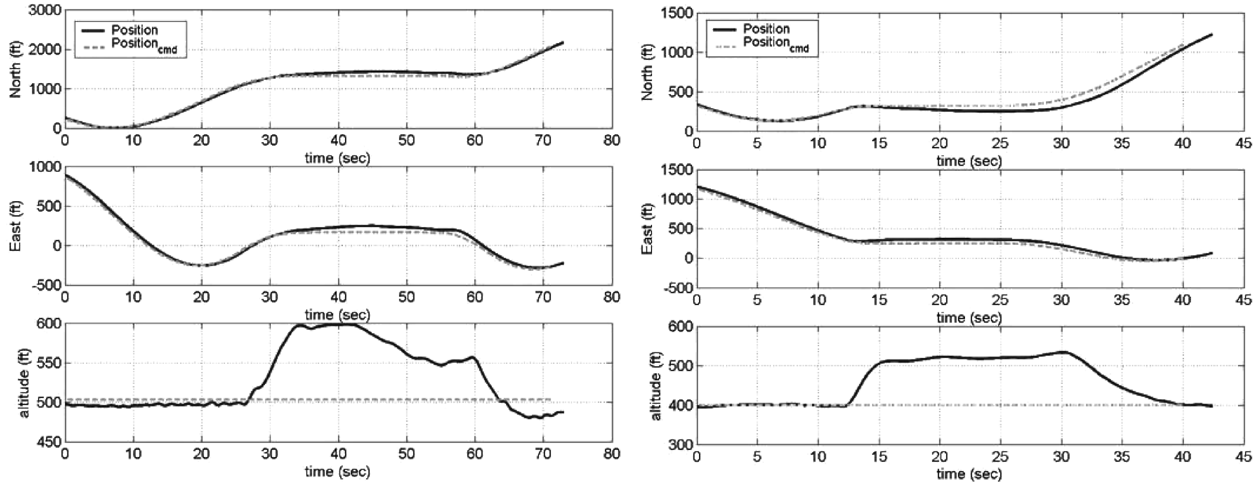
Fig. 9 Ground track during transitions to and from hover: a–b) both aircraft are flying in the clockwise direction.



a) Slower transition

b) Faster transition

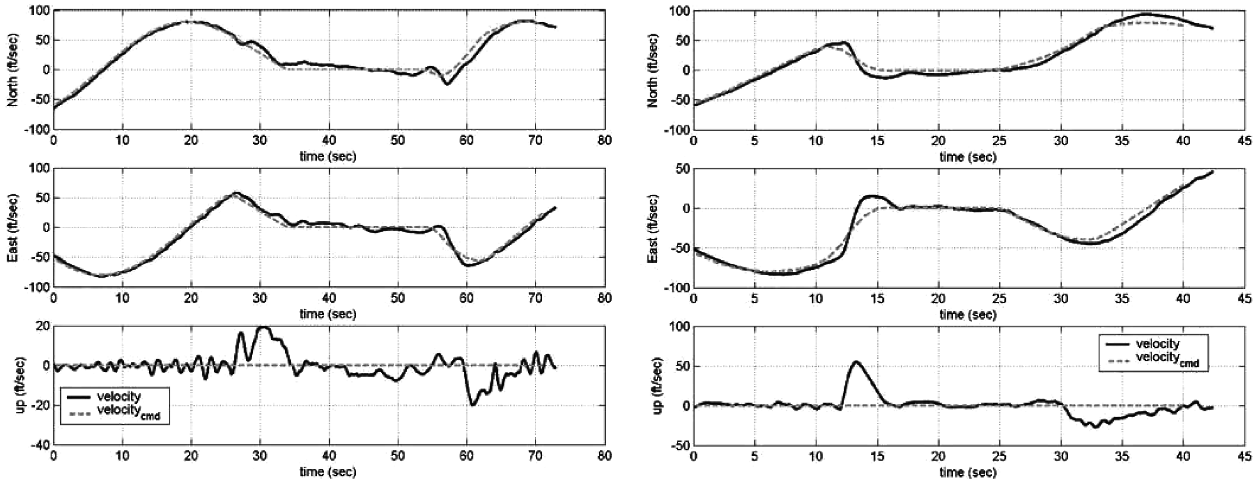
Fig. 10 Three-dimensional trajectory of the GTEdge during transitions to and from hovering flight: a–b) both aircraft are flying in the clockwise direction.



a) Slower transition

b) Faster transition

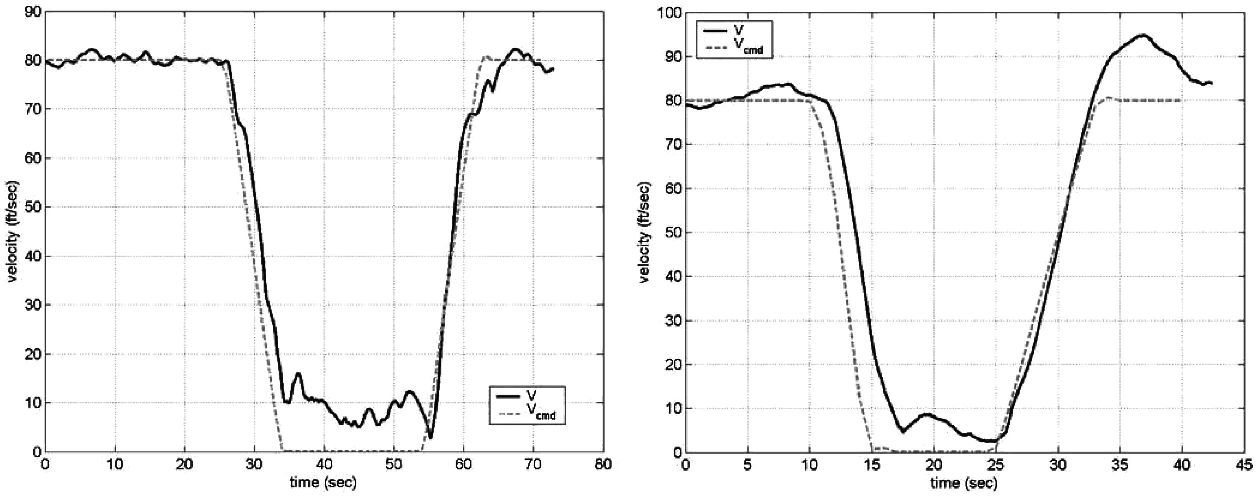
Fig. 11 Commanded and estimated actual positions of the aircraft during hover-transition maneuvers in windy conditions: maximum position errors in a) [102.2 89.6 95.8] ft in the north-east-up directions, and b) [112.5 69.0 133.4] ft.



a) Slower transition

b) Faster transition

Fig. 12 Commanded and estimated actual velocities of the aircraft during hovering flight: maximum velocity errors in a) [17.918.420.1] ft/s in the north-east-up directions, and b) [26.039.054.8] ft/s.



a) Slower transition

b) Faster transition

Fig. 13 Commanded and estimated total groundspeeds for the airplane.

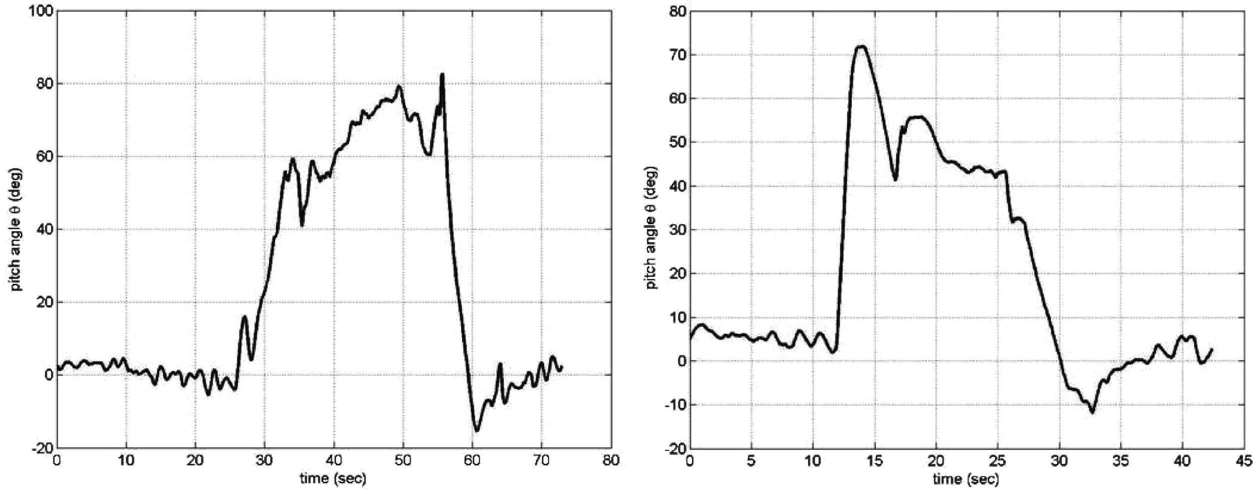
resulting from the slower transition, the controller begins to correct for the altitude error. However, in the hover resulting from the more rapid transition, the vertical velocity settles in at 0 ft/s, thus maintaining an offset in altitude. It should also be noted that when the aircraft exits the hover maneuver in the slower transition, it begins a new orbit with the center of the circle at a new location, as can be seen by the shape of the trajectory in Fig. 9a. The guidance law was later changed so that the airplane would continue along the same circular path upon exiting the hover, as in Fig. 9b.

The velocity profile for the GTEdge throughout the maneuvers in both of the test cases described here is presented in Fig. 13. Figures 14 and 15 show vehicle attitude expressed as Euler angles, with Fig. 14 giving the pitch angle and Fig. 15 showing the roll and heading angles. These plots show that the GTEdge never actually achieves zero velocity relative to the ground, nor does it reach a fully vertical attitude, because the pitch angle remains below 90 deg throughout the hover portion of the maneuver. These apparent discrepancies between the vehicle's commanded performance and its actual behavior are due to the presence of real-world effects such as wind, turbulence, etc. In particular, when the UAS is attempting to maintain near-zero groundspeed in the presence of wind, the aircraft must have its thrust vector at an angle relative to the vertical to counteract the resulting wind forces.

Figure 13 shows that the controller is able to track the velocity profile more closely during the slower transitions than during the faster transition. Also, the pitch angle plots show that the airplane attains a higher pitch angle in the slower transition to hover than when it is commanded to transition more rapidly. These phenomena, however, although possibly a function of the controller's capabilities at different transition rates, may in fact have been the result of differing wind conditions between the two flight tests.

Here, we note that the returning transition from hovering to forward flight is marked by the rapid descent in pitch angle in Fig. 14. As the pitch angle decreases, the GTEdge descends in altitude and begins to pick up speed until it reaches steady horizontal flight at 80 ft/s. One other interesting point that is evident from the attitude information for the slower transition presented in Fig. 15a is that the aircraft actually performed a three-quarter revolution roll at the end of the hover as the commanded velocity was being ramped back up to 80 ft/s, as can be seen by the wraparound in the heading angle at the 56 s marker. This behavior was due to particulars of the trajectory commanded to the aircraft by the guidance law as it exited the hover.

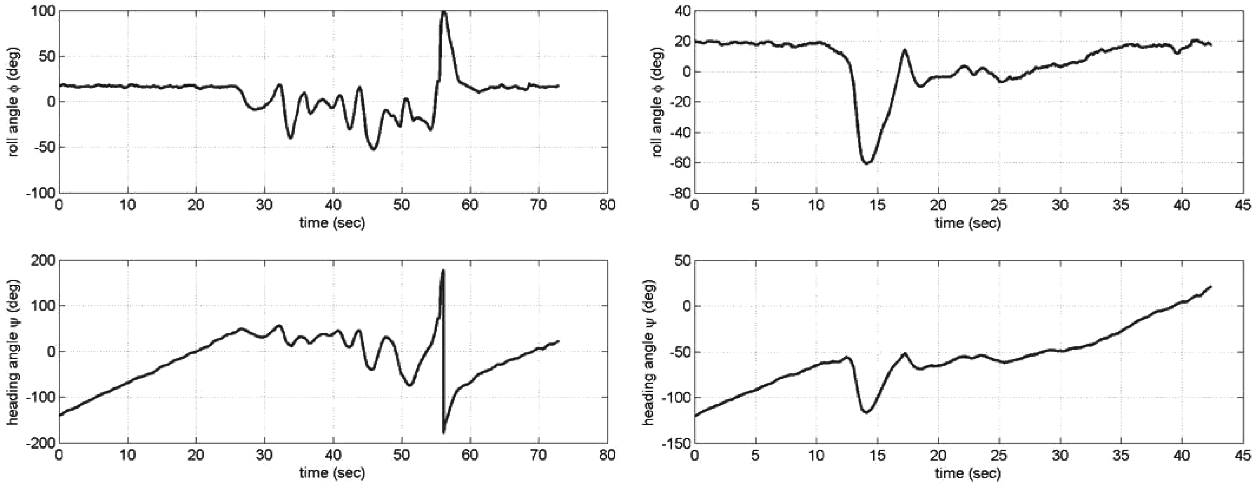
Actuator commands are shown in Figs. 16 and 17, with Fig. 16 showing throttle and Fig. 17 showing the deflections for the rudder, elevator, and ailerons. The throttle settings in Fig. 16 for the two transition rates differ in that the slower transition has an idle throttle setting of −0.90, whereas the faster transition has −0.85 as the minimum. These lower bounds on the throttle prevent engine shutdown midflight. The maximum throttle command in Fig. 16a is 0.62 and the maximum in 16b is 0.38. These control deflections are



a) Slower transition

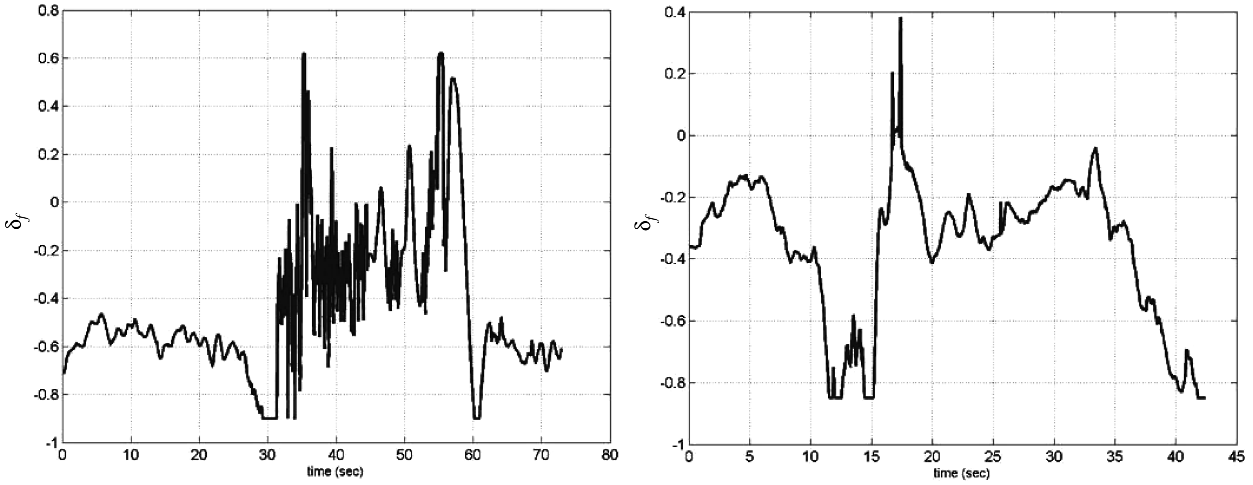
b) Faster transition

Fig. 14 Pitch angle for the GTEdge during the transitions to and from hover. In the slower transition a) the maximum pitch angle is 83 deg and the minimum is −15 deg. In the faster transition b) the maximum pitch angle is 72 deg and the minimum angle is −12 deg.



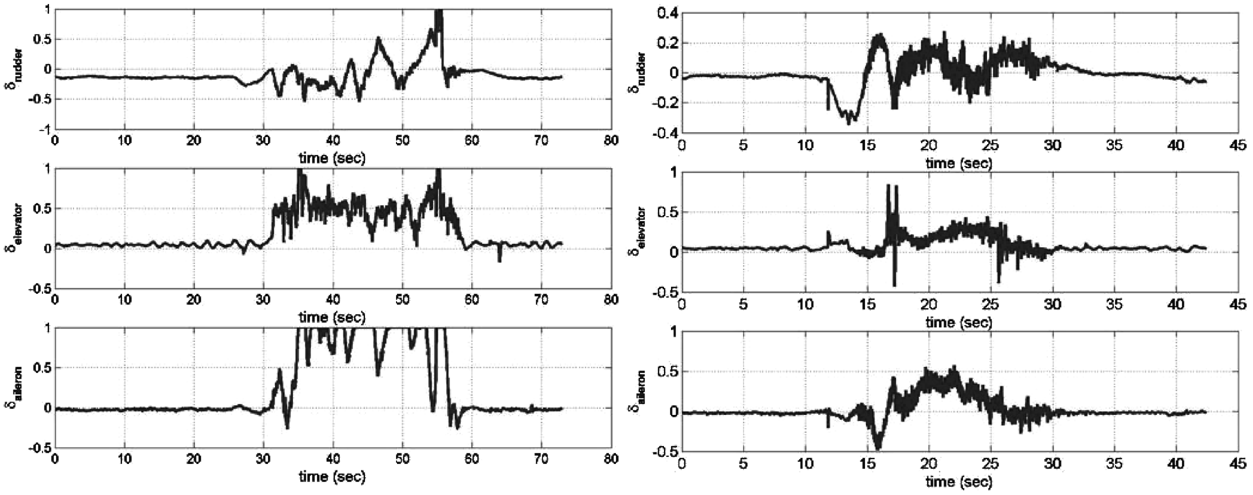
a) Slower transition b) Faster transition

Fig. 15 Roll and heading angles during the transition maneuvers. During steady forward flight, the roll angle remains relatively constant while the heading angle ramps up at a constant rate, because the aircraft is in a steady turn.



a) Slower transition b) Faster transition

Fig. 16 Throttle actuator commands during the transitions to and from hovering flight.



a) Slower transition b) Faster transition

Fig. 17 Aerodynamic control surface deflections during steady and hovering flight.

expressed on a dimensionless scale ranging from -1.0 to 1.0, which represent minimum and maximum saturation values.

The control surface deflection values in Fig. 17 can range from -1.0 to 1.0, which represent minimum and maximum saturation values. Actual deflection values for the elevator ranged between ±32 deg, whereas the aileron and rudder deflections ranged between ±29 and ±25 deg, respectively. The "center" location of each control surface varied slightly from flight to flight based on the aircraft's trim settings. The deflection values are relatively level during steady-level flight, becoming comparatively more erratic during hovering flight. During the faster transition in Fig. 17b, the actuators never reach their limit values. The slower transition in 17a, however, has all the aerodynamic control surfaces reaching their saturation values at some point. The aileron in particular is saturated for the majority of the hover in the slower-transition case.

The pitch angle plot (Fig. 14) and the throttle command plot (Fig. 16) illustrate that the transition to hover is described by a reduction in throttle down to the lower saturation limit as the commanded velocity ramps down, followed by a drastic jump in both throttle and elevator deflections to force the aircraft into a vertical attitude.

During hovering flight, the GTEdge modulates thrust by manipulating the propeller rotation rate to maintain altitude. However, the resulting rolling moment imparted on the aircraft needs to be counteracted by the ailerons in order for a constant attitude to be maintained. With little or no airspeed during a stationary hover, the prop wash from the airplane's propeller can often be the primary source of airflow over the control surfaces. The airflow over the ailerons is generally insufficient to produce the aerodynamic moments required to counteract the engine torque. This is illustrated by the repeated saturation of the ailerons in Fig. 17a in the slower-transition scenario. On the other hand, with the faster transition to hover, the ailerons never reached their saturation values, as can be seen from Fig. 17b. This may have been because the net airspeed due to the high winds during the faster transition provided enough airflow over the ailerons for sufficient control authority.

Future work will include integrating a scheme for bleeding off excess airspeed before the maneuver, which should significantly reduce the "ballooning" in vehicle altitude upon execution of the commanded pitch increase. Future work also includes the possible application of this control architecture to vertical takeoffs and landings for fixed-wing vehicles to allow deployment in cluttered areas where runways might not be readily available. Cameras could also potentially be mounted on such platforms to create fast-moving, fixed-wing vehicles capable of quickly reaching a destination, hovering in place to obtain surveillance, and then quickly exiting. More sophisticated trajectory generation techniques could also be applied to widen the performance envelope of this aircraft to include more aggressive and more accurate transition maneuvers.

V. Conclusions

The results presented in this paper demonstrate the capability of a maneuverable fixed-wing unmanned aircraft system to autonomously transition from forward flight to hovering flight and then back again. A neural network adaptive controller was used to correct for modeling errors present in a simple vehicle model used for feedback linearization via dynamic inversion. Based on the recorded output of the neural network along with the fact that the weights were initialized to zero, it is concluded that, in each experiment, significant and meaningful adaptation of the weights occurred. The inclusion of pseudocontrol hedging allowed for appropriate adaptation and better vehicle control in maneuvers where actuator saturation would otherwise be problematic.

Justification for the use of adaptive control for this application is based on the fact that, although linear systems can be used to adequately model and control an aircraft either in steady-level flight or in ideal hovering flight, the same cannot be said for the highly nonlinear transition between these two regimes, and that accurately modeling or controlling an aircraft during this transition requires the use of systems capable of dealing with these nonlinearities.

During flight testing, the guidance and control systems were found to successfully enact an autonomous transition from forward, steady-level flight to hovering flight and then back again, and also to maintain stable flight during each of these flight regimes. Both slow and faster transitions to hover were flight tested, and it was found that the faster transitions allow slightly more control of commanded hovering position.

It is noted that the term hovering has been used throughout to indicate a flight regime in which groundspeed is very small (or even zero). Thus, hovering usually indicates a flight regime beyond wing stall, where engine thrust is the primary force keeping the aircraft aloft. It is recognized that it is possible for a fixed-wing airplane to fly at zero groundspeed while its wings are not stalled; however, in most cases, the high-wind conditions required to achieve this would preclude safe flight.

Acknowledgments

This work was supported in part by AeroVironment, Inc., Guided Systems Technologies, and the Active-Vision Control Systems Multi-University Research Initiative Program under contract #F49620-03-1-0401. The authors would like to thank the following people for their contributions to the work presented in this paper: Michael Cancienne, J. Eric Corban, Claus Christmann, Henrik Christophersen, Jason Fine, Stewart Geyer, Jeong Hur, Wayne Pickell, Nimrod Rooz, and Brent Yates.

References

[1] Johnson, E. N., "Limited Authority Adaptive Flight Control," Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, 2000, pp. 15–16, 42.

[2] Johnson, E. N., and Kannan, S. K., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 524–538.

[3] Nardi, F., Rysdyk, R. T., and Calise, A. J., "Neural Network Based Adaptive Control of a Thrust Vectored Ducted Fan," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 99-3996, Aug. 1999.

[4] *Aerospace Source Book 2006*, Aviation Week and Space Technology, Vol. 164, No. 23, Jan. 2006.

[5] Stone, H., and Wong, K. C., "Preliminary Design of a Tandem-Wing Tail-Sitter UAV Using Multi-Disciplinary Design Optimisation," *Second International Aerospace Congress*, Feb. 1997, pp. 707–720.

[6] Stone, H., and Clarke, G., "Optimization of Transition Maneuvers for a Tail-Sitter Unmanned Air Vehicle (UAV)," *Ninth Australian International Aerospace Congress*, Australian International Aerospace Congress Paper 105, 2001.

[7] Avanzini, G., Ciniglio, U., and De Matteis, G., "Full-Envelope Robust Control of a Shrouded-Fan Unmanned Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, March–April 2006, pp. 435–443.

[8] Avanzini, G., de Matteis, G., and Fresta, F., "Robust Multivariable Control of a Shrouded-Fan Uninhabited Aerial Vehicle," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2002-4703, Aug. 2002.

[9] Franz, R., Milam, M., and Hauser, J., "Applied Receding Horizon Control of the Caltech Ducted Fan," *American Control Conference* Inst. of Electrical and Electronics Engineers Paper No. 105, 2002.

[10] Hess, R. A., and Ussery, T. M., "Sliding Mode Techniques Applied to the Control of a Micro-Air Vehicle," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2003-5408, Aug. 2003.

[11] Christophersen, H., Pickell, W., Neidhoefer, J., Koller, A., Kannan, S., and Johnson, E., "Compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, No. 5, May 2006, pp. 187–213.

[12] Christophersen, H., Pickell, W., Koller, A., Kannan, S., and Johnson, E., "Small Adaptive Flight Control Systems for UAVs using FPGA/DSP Technology," *Proceedings of the AIAA Unmanned Unlimited Conference*, AIAA, Reston, VA, 2004.

[13] Dittrich, J., and Johnson, E. N., "Multi-Sensor Navigation System for an Autonomous Helicopter," *Proceedings of the 21st AIAA/IEEE Digital Avionics Systems Conference*, AIAA Paper No. 2003-5408, 2003.

[14] Dittrich, J., "Design and Integration of an Unmanned Aerial Vehicle

Navigation System," M.S. Thesis, School of Aerospace Engineering, Georgia Inst. of Technology, May 2002.

[15] Bletzacker, F. R., Eller, D. H., Forgette, T. M., Seibert, G. L., Vavrus, J. L., "Kalman Filter Design for Integration of Phase 3 GPS with an Inertial Navigation System," *Inst. of Navigation, National Technical Meeting, Proceedings (A88-51701 22-04)*, 1988.

[16] Johnson, E., Schrage, D., Prasad, J., and Vachtsevanos, G., "UAV Flight Test Programs at Georgia Tech," *Proceedings of the AIAA Unmanned Unlimited Technical Conference, Workshop, and Exhibit*, AIAA, Reston, VA, 2004.

[17] Kannan, S. K., "Adaptive Control of Systems in Cascade with Saturation," Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, 2005.

[18] Slotine, J., and Li, W., *Applied Nonlinear Control*, Prentice–Hall, Upper Saddle River, NJ, 1991, p. 94.

[19] 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, Dec. 2003, pp. 906–913.

[20] Calise, A. J., Johnson, M. D., and Yoonghyun, S., "Adaptive Flight Control of Advanced Fighter Aircraft at High Angles of Attack," *Proceedings of the Midwest Symposium on Circuits and Systems*, Georgia Inst. of Technology, Atlanta, 2005.

[21] Idan, M., Johnson, M. D., and Calise, A. J., "Hierarchical Approach to Adaptive Control for Improved Flight Safety," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, 2001.