AIAA Guidance, Navigation, and Control Conference and Exhibit
21 - 24 August 2006, Keystone, Colorado

AIAA 2006-6775

Flight Test Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover

Eric N. Johnson [1], Michael A. Turbe[2], Allen D. Wu[3] Suresh K. Kannan [4], and James C. Neidhoefer [5]
Georgia Institute of Technology, Atlanta, GA, 30332-0150

Fixed-wing unmanned aerial vehicles (UAVs) with the ability to hover have significant potential for applications in urban or other constrained environments where the combination of fast speed, endurance, and stable hovering flight can provide strategic advantages. This paper discusses the use of dynamic inversion with neural network adaptation to provide an adaptive controller capable of transitioning a fixed-wing UAV to and from hovering flight in a nearly stationary position. This approach allows utilization of the entire low speed flight envelope even beyond stall conditions. The method is applied to the GTEdge, an 8.75 foot wing span fixed-wing aerobatic UAV which has been fully instrumented for autonomous flight. Results from actual flight test experiments of the system where the airplane transitions from high speed steady flight into a stationary hover and then back are presented.

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
α	=	angular acceleration or angle of attack
$\boldsymbol{\alpha}(\cdot), \hat{\boldsymbol{\alpha}}(\cdot)$	=	attitude dynamics and estimate
$b_{ax,y,z}$	=	accelerometer measurement biases
$b_{\omega x,y,z}$	=	rate gyro biases
b_v, b_w	=	neural network (NN) biases
$\bar{\boldsymbol{\Delta}}(\cdot)$	=	total function approximation error
$\Delta\Phi$	=	attitude correction
$\delta, \hat{\delta}$	=	actuator deflection and estimate
ζ	=	damping ratio
f	=	the process model
F_k	=	the process model Jacobian at t_k
g	=	the gravitational constant
$\mathbf{g}$	=	gravitational acceleration, vector
$\boldsymbol{\Gamma}_{\mathbf{v}}, \boldsymbol{\Gamma}_{\mathbf{w}}$	=	NN learning rate matrices
h_t	=	altitude of ground level
h_{cg}	=	altitude of the center of gravity (c.g.)

[1] Lockheed Martin Assistant Professor of Avionics Integration, Aerospace Eng. Dept., Georgia Institute of Technology, Atlanta, GA, 30332-0150

[2] Graduate Research Assistant, Aerospace Eng. Dept., Georgia Institute of Technology, Atlanta, GA, 30332-0150

[3] Graduate Research Assistant, Aerospace Eng. Dept., Georgia Institute of Technology, Atlanta, GA, 30332-0150

[4] Senior Research Scientist, Guided Systems Technologies, 75 Fifth Street, NW, Suite 336, Atlanta, GA 30308

[5] Research Engineer, Aerospace Eng. Dept., Georgia Institute of Technology, Atlanta, GA, 30332-0150

h_k	=	the measurement model at time t_k
H_k	=	the measurement model Jacobian at t_k
h_{sensor}	=	altitude of the pressure sensor
K_k	=	Kalman Filter gain at time $\mathrm{t_k}$
$\mathbf{K}, \mathbf{R}$	=	inner-loop, outer-loop gain matrices
κ	=	E-modification parameter
$\mathbf{p}$	=	position vector
P_k^-	=	predicted error covariance matrix at time t_k
P_k^+	=	corrected error covariance matrix at time t_k
$\tilde{\mathbf{Q}}(\cdot)$	=	attitude error angle function
$\mathbf{q}$	=	attitude quaternion
$q_0\ q_1\ q_2\ q_3$	=	quaternion parameters
$\mathbf{q}(\cdot)$	=	Euler rotation to quaternion transform
$Q(t)$	=	process noise covariance matrix at time t
r_{GPS}	=	position vector of GPS relative to the c.g.
r_{sensor}	=	position vector of sensor relative to the c.g.
R_k	=	measurement noise covariance matrix at time t_k
$\boldsymbol{\sigma}, \boldsymbol{\sigma}'$	=	neuron sigmoidal function, gradient
Δt	=	integration time step
$T_{b->i}$	=	transformation from body frame to inertial frame
$T_{i->b}$	=	transformation from inertial frame to body frame
$T_{b->i}[3]$	=	third row of $T_{b\text{-}>i}$
t_k	=	the time at which the k^{th} measurement is taken
$\theta, \theta_v, \theta_w$	=	polar coordinate, NN thresholds
u ,	=	x-body axis velocity
$u(t)$	=	control vector at time t
u_k	=	process control vector at time t_k
$\mathbf{V}, \mathbf{W}, v, w$	=	NN input and output weights
$\mathbf{v}, V$	=	velocity
v_k	=	measurement noise at time t_k
v	=	y-body axis velocity
v_{cg}	=	velocity of the c.g.
v_{GPS}	=	velocity of the GPS in and inertial frame
$\mathbf{v_{ad}}, \bar{\mathbf{v}}_{\mathbf{ad}}$	=	adaptive element signals
$\mathbf{v_r}$	=	robustifying signal
w	=	z-body axis velocity
$w(t)$	=	process noise at time t
w_{sensor}	=	pressure sensor measurement noise
$x, \mathbf{x}$	=	state variable, state vector
$x_{in}, \bar{\mathbf{x}}$	=	NN input
$x(t)$	=	process model state
$x_{pos}\ x_{pos}$	=	x position
x_{posGPS}	=	position of the GPS in an inertial frame

x_{poscg}	= position of the c.g.
x_k	= process state at time t_k
$\hat{x}_k^-$	= predicted state estimate at time t_k
$\hat{x}_k^+$	= corrected state estimate at time t_k
y_{pos}	= y position
$\hat{y}_k$	= output estimate at time t_k
z_k	= observations at time t_k
z_{pos}	= z position
z_j	= input to jth hidden-layer neuron
ω	= angular velocity or natural frequency

Acronyms

A/D	= analog to digital converter
CAS	= command augmentation system
DSP	= digital signal processor
EC20	= FCS20 processor board
EKF	= Extended Kalman Filter
FCS20	= Flight Control System 20
FIFO	= first in first out
FPGA	= field programmable gate array
GCS	= ground control station
GPS	= global positioning system
GN&C	= guidance, navigation, and control
IMU	= inertial measurement unit
IO	= input/output
ISR	= intelligence, surveillance, and reconnaissance
NN	= neural network
NAC	= neural adaptive controller
MEMS	= micro electromechanical system
PCH	= pseudo control hedging
PD	= proportional/derivative
SB20	= FCS20 sensor/power board
SDRAM	= synchronous dynamic random access memory
UAV	= unmanned aerial vehicle

I. Introduction

FIXED-wing unmanned aerial vehicles (UAVs) with the ability to hover in a stationary position offer unique capabilities since they enable a stable observation platform for surveillance 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 UAVs used for border patrol, traffic monitoring, and hazardous site inspection. The capability to perform tail-sitting takeoffs and landings also increases the operational domains of such aircraft.

While the ability to hover fixed-wing UAVs has been demonstrated with remotely piloted vehicles, performing such maneuvers autonomously presents many unique challenges. Successful guidance and control schemes must be robust to uncertainties present in the available models of the vehicle dynamics in these regimes. Additionally, in ensuring robustness, it is often necessary to place significant limits on the flight envelope. However, these limits can be detrimental to autonomous vehicles operating in urban environments where aggressive maneuvering is often useful.

Adaptive control lends itself well to this challenge. The specific approach chosen to control the GTEdge both for normal operation and in hover is dynamic inversion with adaptation. A neural network is trained to account for errors in a simple vehicle model used in the dynamic inversion. Additionally, pseudo-control hedging (PCH) allows the neural network to continue adapting when actuator nonlinearities, such as saturation, occur. This type of neural network adaptive control has been used successfully in similar projects [1-3].

Work on hovering fixed-wing UAVs is currently being done by several research groups. William E. Green and Paul Y. Oh at Drexel University [4,5] have performed autonomous flight tests of a Micro-UAV hovering in an urban environment. These experiments involved the use of a Microstrain IMU, a PIC16F87 microcontroller, and a linear controller for hovering a small fixed-wing UAV in an urban environment. Transitions to and from hover were performed manually. Aerovironment's SkyTote UAV [6,7] is a UAV with the potential capabilities to hover, take off and land vertically, and also transition into conventional horizontal fixed-wing flight. The prototype SkyTote is complete and hover testing is currently underway. Aurora Flight Sciences' GoldenEye family of UAVs [8,9] have been used in numerous flight tests throughout 2005 and 2006 in which autonomous transitions were made from vertical hovering flight to horizontal fixed-wing flight, and back again. While the Goldeneye has the ability to fly in a fixed-wing configuration, its wing is not "fixed" in that it has the ability to vary the angle of incidence in flight. The University of Sydney is developing a T-Wing UAV [10-11] 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.

This paper provides a description of the GTEdge UAV and the guidance and adaptive control architectures used to successfully transition the aircraft into hover 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. Finally, the flight test results are presented, followed by a discussion of the present architecture's limitations and suggestions for improving performance.

II. Flight Test Hardware Description

The GTEdge UAV consists of four major subsystems: 1) the baseline commercially available 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.

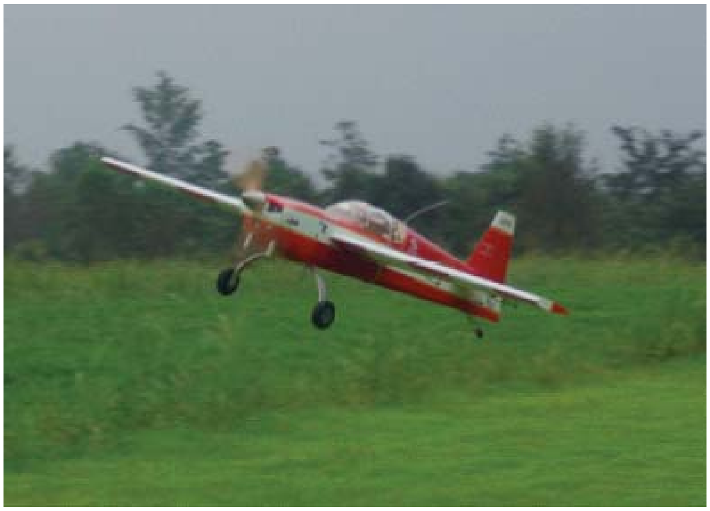


Figure 1. The GTEdge research UAV is a modified 33% scale Edge 540T. This vehicle was selected for its ability to carry moderate payloads and to perform aggressive aerobatic maneuvers. Vibration isolation and electromagnetic interference shielding protect the avionics mounted within the aircraft's canopy.

A. The GTEdge Airframe

The GTEdge (see Figure 1) is a modified Aeroworks 33% scale Edge 540T aircraft which is commercially available as an aerobatic radio-controlled airplane. 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, allowing for accelerated vertical climbs. Payload requirements governed the selection of the airplane's scale. The GTEdge has the following physical characteristics:

- Wing span: 8.75 ft; Wing area: 13 ft^2; Length 7.8 ft
- Engine: gasoline, Desert Aircraft DA100 100cc engine, 9.8hp
- Dry weight of aircraft without payload: 35 lbs
- Propeller: 27x10 wooden propeller
- Throttle, elevator, aileron, rudder actuated by JR8611A ultra torque digital servos from JR
- Lithium-ion batteries power the actuators
- Endurance of approximately 30 minutes at steady 70 ft/s flight

B. The 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) [12-13] recently developed by the Georgia Institute of Technology is the cornerstone of the GTEdge's avionics suite. This miniature computer uses its floating point Digital Signal Processor (DSP) for high level serial processing, its Field Programmable Gate Array (FPGA) for low level parallel processing, along with Micro Electro Mechanical Systems (MEMS) sensors. The FCS20 is mounted in the GTEdge's canopy for modularity since the canopy can be readily removed from the aircraft. A metal enclosure and shielded wires mitigate the effects of electromagnetic interference on the onboard electronics. Vibration isolators support the enclosure to protect inertial sensors from vibrations generated by the engine and the rotation of the propeller.

The basic modules of the FCS20 are the EC20 processor board and the SB20 sensor/power board. A minimal FCS20 system consists of one EC20 board and one SB20 board, usually set up with the boards connected back to back. The EC20 Processor board (see Figure 2) handles FCS20 processing and internal and external communications. The EC20 design uses a fast sequential processing core in the DSP supported by a parallel system of lower-level support components in the FPGA. The board was implemented with a Texas Instruments 225 MHz Floating Point TMS320C6713 DSP along with an Altera 1M gate Stratix FPGA. In addition, the EC20 includes a high-speed internal data bus and few external interrupts, allowing the DSP to spend more time on high speed processing and less time on data transfers. The EC20 contains all of the components required for fast data processing, including a total of 32Mbytes of SDRAM and 8Mbytes of configuration FLASH memory. Other EC20 features include power saving circuitry, 110 general purpose IO-pins, optional remote configuration control and dedicated board-to-board fast differential serial lines that operate at up to 840Mb/s.

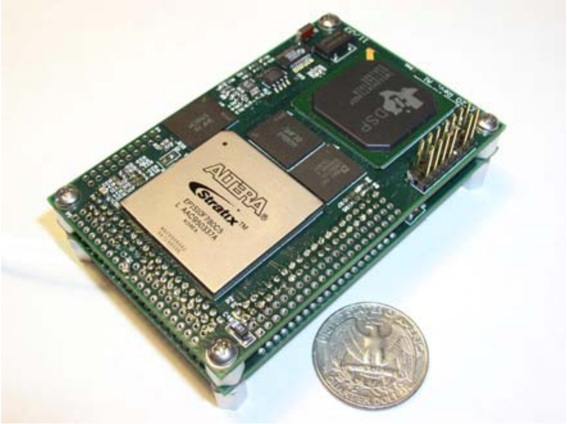


Figure 2. The FCS20 with EC20 Processor Board on top

The SB20 sensor/power board (see Figure 3) 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 2-axis 10g accelerometers, a μBlox GPS module, and Freescale absolute and differential pressure sensors. In addition, the rate gyros have temperature-sensing capabilities that can be used to detect the installed operating conditions inside the enclosure. The SB20 also has an A/D channel dedicated for sensing the main power supply voltage.

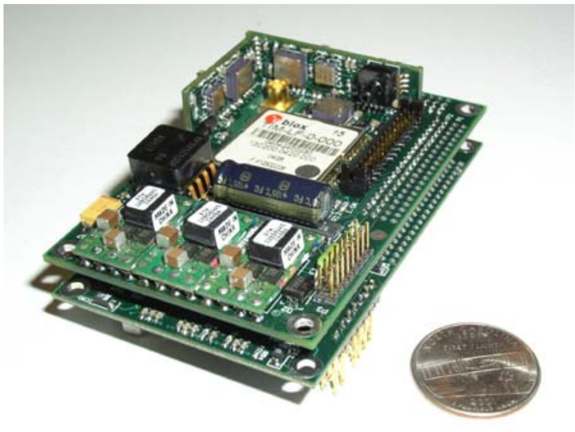


Figure 3. FCS20 with SB20 Sensor/Power Board on top

The SB20 uses the same form factor as the EC20 with a layout such that the main headers line up when the boards are assembled back-to-back. In general, the layout of the SB20 was driven by the desire to locate the high frequency switching power regulators as far as possible from the sensitive circuitry in the analog sensors and A/D converters.

Internal data flow in the FCS20 is dominated by the communication between the DSP and the FPGA that occurs through the high-speed 32-bit internal data bus. Multiple, parallel First-In First-Out (FIFO) components inside the FPGA enable communication with the DSP from dedicated components within the FPGA, avoiding information bottlenecks and reducing time delays. The FIFOs also act as buffers between the different components, which helps ensure data integrity. Figure 4 highlights the main data flow pathways in the FCS20 system. Onboard the SB20, the analog sensors are read by the A/D converters. The accelerometers have both analog and digital outputs, and the digital outputs are sent directly to counters in the FPGA. The GPS on the SB20 communicates with the FPGA via an RS232 serial line. As the FPGA receives packets of sensor data, it processes and packages the data, and sends relevant data to the DSP via the high-speed internal data bus. The data is then processed on the DSP by the guidance, navigation, and control algorithms. Control outputs are then sent back to the FPGA via the internal data bus, converted to PWM signals, and sent out to external servos.

As a measure of safety, the FPGA also hosts logic that allows for the GTEdge to be switched over to manual control by a safety pilot at the flip of a switch. Similar logic also defaults to the computer-controlled autopilot should the aircraft move beyond range of the safety pilot's transmitter.

Other significant components of the GTEdge's avionics suite include a Novatel OEM4 differential GPS and a Freewave spread spectrum transceiver for communications with the ground control station.

C. The Extended Kalman Filter Based Navigation System

An integral component of the FCS20 is the sixteen state Extended Kalman Filter (EKF) that uses data from the sensors on the SB20 processor board to generate a navigation solution that closely estimates the state of the system [14-15]. 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 the slower rate of 10 Hz.

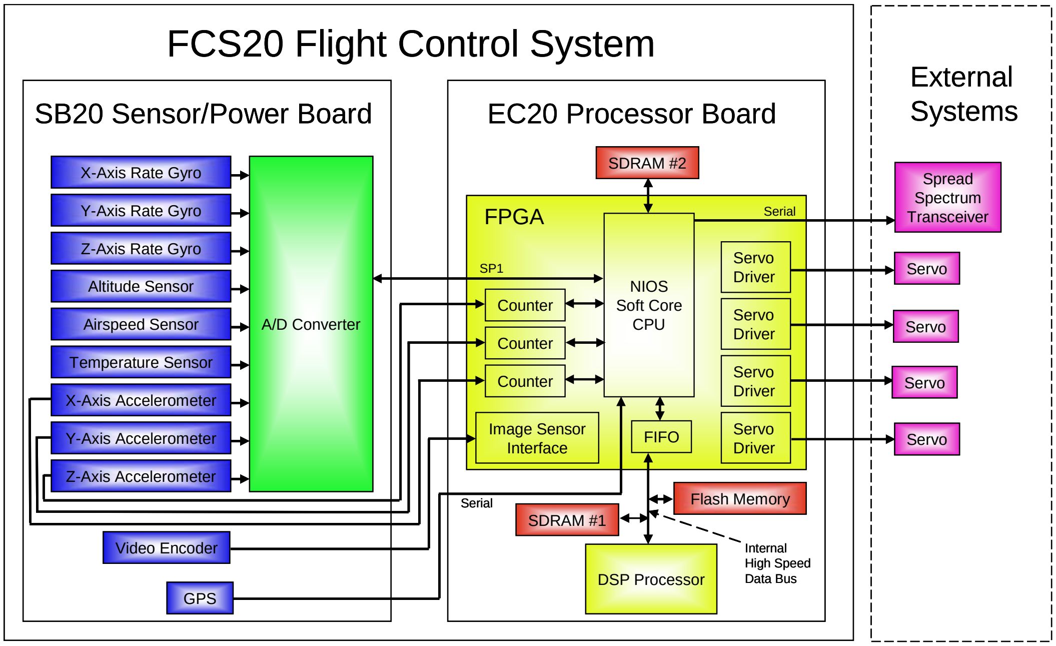


Figure 4. FCS20 Data Flow Diagram

1. *The EKF Process Model*

The EKF is described by a set of continuous and discrete equations. Equation (1) is the state vector for the EKF process, and equation (2) represents the continuous process model. The states in the navigation filter include: four quaternion components, three position states in the local inertial frame, three velocity states in the body-fixed frame, and six accelerometer and gyro biases.

$$x(t) = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & x_{pos} & y_{pos} & z_{pos} & u & v & w & b_{ax} & b_{ay} & b_{az} & b_{\omega x} & b_{\omega y} & b_{\omega z} \end{bmatrix} \tag{1}$$

$$\dot{x}(t) = f(t, x(t), u(t)) + w(t) \tag{2}$$

Since one of the main goals of the EKF is to provide values of the Euler orientation angles for pitch, roll, and yaw (or equivalently, in this case, the four quaternion parameters) based on acceleration and angular rate measurements, the process model *f* is made up of the following kinematic relationships [16]:

$$\begin{gathered}
\begin{bmatrix} \dot{\hat{q}}_0 \\ \dot{\hat{q}}_1 \\ \dot{\hat{q}}_2 \\ \dot{\hat{q}}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\left(\omega_{IMUx} - \hat{b}_p\right) & -\left(\omega_{IMUy} - \hat{b}_q\right) & -\left(\omega_{IMUz} - \hat{b}_r\right) \\ \omega_{IMUx} - \hat{b}_p & 0 & \omega_{IMUz} - \hat{b}_r & -\left(\omega_{\mathrm{IMUy}} - \hat{\mathrm{b}}_{\mathrm{q}}\right) \\ \omega_{IMUy} - \hat{b}_p & -\left(\omega_{IMUz} - \hat{b}_r\right) & 0 & \omega_{IMUx} - \hat{b}_p \\ \omega_{IMUz} - b_r & \omega_{IMUy} - b_q & -\left(\omega_{\mathrm{IMUx}} - \mathrm{b}_{\mathrm{p}}\right) & 0 \end{bmatrix} \begin{bmatrix} \hat{q}_0 \\ \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{bmatrix} \\
\dot{\hat{x}} = T_{b->i}\hat{v} \\
\dot{\hat{v}} = T_{b->i}(a_{IMU} - \hat{b}_a) + g \\
\dot{\hat{b}}_a = 0 \\
\dot{\hat{b}}_\omega = 0
\end{gathered} \tag{3}$$

The transformation from the body frame to the inertial frame is given by:

$$T_{b->i} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{4}$$

Equations (5) and (6) represent the discrete measurement observations and the discrete output estimate. In these equations, time t_k is the time at which the k^{th} measurement is taken and v_k if the measurement noise.

$$z_k = h_k(t_k, x_k, u_k) + v_k \tag{5}$$

$$\hat{y}_k = h_k(t_k, \hat{x}_k^+, u_k) \tag{6}$$

2. The EKF Measurement Model

The measurement model *h(·)* corresponds to the sensor measurements (except for the IMU) at the c.g. The set of equations in (7) represent the sensor measurements corrected for the effects of measurement noise and for their off-center location. In equation (8), z_k is the vector of corrected measurements that are available to the EKF.

$$\begin{aligned} x_{poscg} &= x_{posGPS} - T_{b->i} r_{GPS} - w_x \\ v_{cg} &= v_{GPS} - T_{b->i} \omega \times r_{GPS} - w_v \\ h_{cg} &= h_{sensor} - T_{b->i}[3] r_{sensor} + h_t - w_{sensor} \end{aligned} \tag{7}$$

$$z_k = [x_{poscg} \quad v_{cg} \quad h_{cg}] \tag{8}$$

3. The Extended Kalman Filter Implementation

Equations (9) and (10) are the state estimate time update (predictor) equations, and equations (11) and (12) are the error covariance time update equations. In these equations, the continuous derivatives (equations (9) and (11)) are used along with standard numerical integration methods to propagate forward in time. In the case of the state estimate propagation, a 2nd order modified Euler's method (equation (10)) is used, where $\dot{\hat{x}}_{k+1}^-$ is determined using a temporary value for $\hat{x}_{k+1}^-$ that comes from a 1st order Euler predictor step. In the case of the error covariance, a 1st order Euler's method (equation (12)) is used.

$$\dot{\hat{x}}(t) = f(t, \hat{x}_k^+, u) \tag{9}$$

$$\hat{x}_{k+1}^- = \hat{x}_k^+ + \frac{\Delta t}{2}(\dot{\hat{x}}_{k+1}^- + \dot{\hat{x}}_k) \tag{10}$$

$$\dot{P}(t_k) = F(t, \hat{x}_k^-, u_k)P_k^- + P_k^- F^T(t, \hat{x}_k^-, u_k) + Q(t_k) \tag{11}$$

$$P_{k+1}^- = P_k^+ + \dot{P}(t_k)\Delta t \tag{12}$$

Equation (13) is the equation for the Kalman Filter gain, and equations (14) and (15) are the state and error covariance measurement update (corrector) equations.

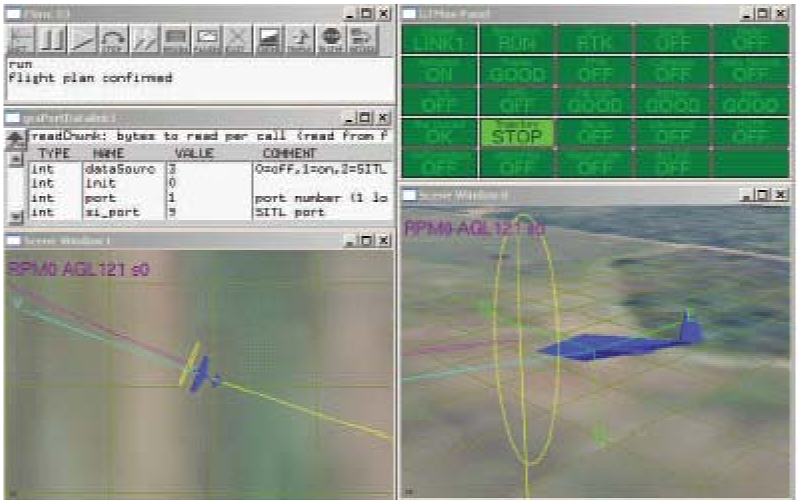
$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{13}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - h_k(t_k, \hat{x}_k^-, u_k)) \tag{14}$$

$$P_k^+ = (I - K_k H_k)P_k^- \tag{15}$$

The EKF equations (9)-(15) above are similar to those derived in [17-18].

It is interesting to note that while one of the main functions of the EKF is to estimate orientation angles based on rates and accelerations, the rates and accelerations are not included as process states, nor are they included in the EKF measurements in z_k (equation (8)). The acceleration and rate measurements are corrected for being located off the c.g., and then are used directly in the right hand side of the process equations (equations (3)). This is how the process noise is introduced into the EKF system. Another similar EKF implementation can be found in [19] for comparison.



```
run
Flight plan confirmed
```

```
readChunk: bytes to read per call (read from f
 TYPE   NAME        VALUE   COMMENT
 int    dataSourc   2       0=off,1=on,2=SITL
 int    init        0
 int    port        1       port number (1 lo
 int    si_port     5       SITL port
```

Figure 5. Image from the Ground Control Station (GCS) computer for the GTEdge. State information transmitted from the onboard system to the GCS is used to display the aircraft. The colored panel in the top-right window displays status information and quickly alerts the operator of any system malfunctions.

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 10Hz, and another message packet containing less vital information once every second. These message packets provide the GCS with information used to display vehicle information to the GCS operator and are also recorded on the GCS to provide extra data recording to supplement the data recorded by the 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. A screenshot of the GTEdge GCS computer is shown in Figure 5.

III. The Guidance and Control Systems

A neural-network adaptive controller [2] developed and flight tested on other UAVs [20] was used to control the GTEdge. An overview of the controller architecture is included in this paper, along with the major governing equations. A proof of the underlying theory and validation of the controller's capabilities is given in [25].

Many methods for autonomously guiding and controlling UAVs have been developed [21-24]. In this application, 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) controller, and a hedging block as shown in Figure 6. Each loop utilizes feedback linearization, and more specifically dynamic inversion, as the control strategy. The transformed system may be controlled by means of a linear PD compensator to track the output of a reference model. The reference model is nonlinear and selected to impose a desired closed loop response and also impose rate limits on the evolution of states. Any step changes in the external command now appear as continuous signals to the linear PD compensator.

Dynamic inversion, when used alone, requires accurate system models for all flight regimes [3] and these models can be costly and difficult to obtain. Alternatively, a simple model of the vehicle is used in the approximate dynamic inversion, and a neural network is trained online to correct for these modeling errors. However, in this approach, certain nonlinear effects such as actuator saturation can create difficulties for the adaptive element [1,2]. Thus a technique called pseudo-control 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. This control methodology reduces dependence on accurate models of the system and actuators.

In this cascaded inner and outer loop architecture [25], the inner loop appears to the outer loop as an actuator that generates translational accelerations by means of a commanded attitude. It should be noted that this configuration was initially designed for rotorcraft where the translational dynamics are strongly dependent upon the attitude dynamics since these types of vehicles accelerate primarily by manipulating the direction and magnitude of the thrust vector. Thus, in the case of a fixed-wing aircraft, the outer loop generates a throttle command and an attitude augmentation 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. Most fixed-wing vehicles, on the other hand, are equipped with aerodynamic surfaces in addition to variable magnitude fixed-direction thrust for generating maneuvering forces and moments. However, the adaptive controller implemented here is capable of compensating for these neglected effects. In order to tailor the adaptive neural network controller for use with fixed-wing aircraft, only modifications to the dynamic inversion were required.

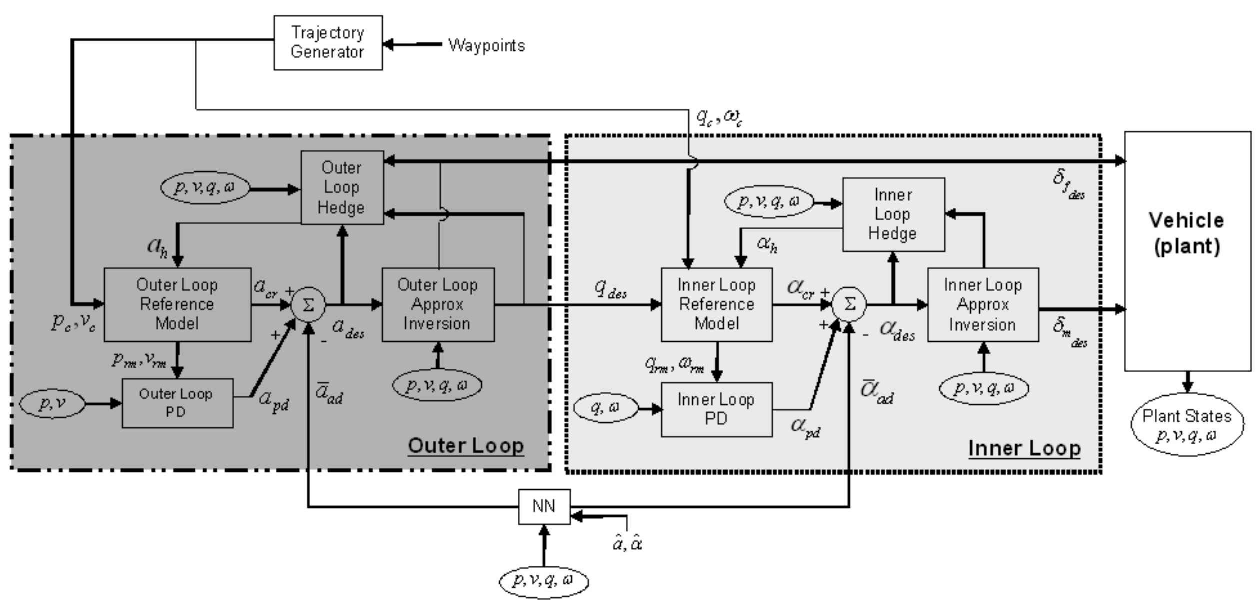


Figure 6: Overall controller architecture used for the stationary hovering of a fixed-wing UAV. The outer loop controls translational dynamics whereas the inner loop is responsible for vehicle attitude dynamics.

A. Controller Synthesis

An aircraft can generally be described with the following nonlinear equations [2]:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{16}$$

$$\dot{\mathbf{v}} = \mathbf{a}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_{\mathbf{f}}, \boldsymbol{\delta}_{\mathbf{m}}) \tag{17}$$

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}(\mathbf{q}, \boldsymbol{\omega}) \tag{18}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_{\mathbf{f}}, \boldsymbol{\delta}_{\mathbf{m}}) \tag{19}$$

in which quaternions are used to express vehicle attitude.

The control architecture is governed by nested outer and inner loops which handle the translational dynamics, equation (17), and the attitude dynamics, equation (19) respectively. The state vector of the vehicle is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T & \mathbf{q}^T & \boldsymbol{\omega}^T \end{bmatrix}^T \tag{20}$$

The vector of control signals is written as:

$$\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_{\mathbf{f}}{}^{T} & \boldsymbol{\delta}_{\mathbf{m}}{}^{T} \end{bmatrix}^{T} \tag{21}$$

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

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

$$\begin{bmatrix} \mathbf{a}_{\mathbf{des}} \\ \boldsymbol{\alpha}_{\mathbf{des}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{q}_{\mathbf{des}}, \boldsymbol{\delta}_{\mathbf{f}_{\mathbf{des}}}, \hat{\boldsymbol{\delta}}_{\mathbf{m}}\right) \\ \hat{\boldsymbol{\alpha}}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_{\mathbf{f}}, \boldsymbol{\delta}_{\mathbf{m}_{\mathbf{des}}}\right) \end{bmatrix} \tag{22}$$

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, $\mathbf{a}_{\mathbf{des}}$ and $\boldsymbol{\alpha}_{\mathbf{des}}$ are commonly referred to as the pseudo-control signals and are analogous to desired translational and angular accelerations. The control inputs and attitude needed to produce the desired pseudo-controls are $\boldsymbol{\delta}_{\mathbf{f}_{\mathbf{des}}}$, $\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{des}}}$, and $\mathbf{q}_{\mathbf{des}}$, respectively and $\hat{\boldsymbol{\delta}}_{\mathbf{f}}$ and $\hat{\boldsymbol{\delta}}_{\mathbf{m}}$ 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:

$$\begin{bmatrix} \boldsymbol{\delta}_{\mathbf{f}_{\mathbf{des}}} \\ \mathbf{q}_{\mathbf{des}} \end{bmatrix} = \hat{\mathbf{a}}^{-1}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{a}_{\mathbf{des}}, \hat{\boldsymbol{\delta}}_{\mathbf{m}}\right) \tag{23}$$

$$\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{des}}} = \hat{\boldsymbol{\alpha}}^{-1}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_{\mathbf{f}}, \boldsymbol{\alpha}_{\mathbf{des}}\right) \tag{24}$$

This approximate inversion provides the following closed-loop dynamics:

$$\dot{\mathbf{v}} = \mathbf{a}_{\mathbf{des}} + \overline{\boldsymbol{\Delta}}_{\mathbf{a}}\left(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}\right) - \mathbf{a}_{\mathbf{h}} \tag{25}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}_{\mathbf{des}} + \overline{\boldsymbol{\Delta}}_{\boldsymbol{\alpha}}\left(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}\right) - \boldsymbol{\alpha}_{\mathbf{h}} \tag{26}$$

in which

$$\overline{\boldsymbol{\Delta}}(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) = \begin{bmatrix} \overline{\boldsymbol{\Delta}}_{\mathbf{a}}(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) \\ \overline{\boldsymbol{\Delta}}_{\boldsymbol{\alpha}}(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(\mathbf{x}, \boldsymbol{\delta}) - \hat{\mathbf{a}}(\mathbf{x}, \hat{\boldsymbol{\delta}}) \\ \boldsymbol{\alpha}(\mathbf{x}, \boldsymbol{\delta}) - \hat{\boldsymbol{\alpha}}(\mathbf{x}, \hat{\boldsymbol{\delta}}) \end{bmatrix} \tag{27}$$

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 pseudo-control so that the desired pseudo-control may be unrealizable; this effect introduces the $\mathbf{a}_{\mathbf{h}}$ and $\boldsymbol{\alpha}_{\mathbf{h}}$ terms in the closed-loop dynamical equations. System stabilization can be achieved by choosing the pseudo-controls in the following manner [2]:

$$\mathbf{a}_{\mathbf{des}} = \mathbf{a}_{\mathbf{cr}} + \mathbf{a}_{\mathbf{pd}} - \overline{\mathbf{a}}_{\mathbf{ad}} \tag{28}$$

$$\boldsymbol{\alpha}_{\mathbf{des}} = \boldsymbol{\alpha}_{\mathbf{cr}} + \boldsymbol{\alpha}_{\mathbf{pd}} - \overline{\boldsymbol{\alpha}}_{\mathbf{ad}} \tag{29}$$

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

The following sections outline the determination of the signals used in the computation of the pseudo-controls. 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

$$\mathbf{e} = \begin{bmatrix} \mathbf{p_r} - \mathbf{p} \\ \mathbf{v_r} - \mathbf{v} \\ \tilde{\mathbf{Q}}(\mathbf{q_r}, \mathbf{q}) \\ \boldsymbol{\omega}_\mathbf{r} - \boldsymbol{\omega} \end{bmatrix} \tag{30}$$

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

$$\tilde{\mathbf{Q}}(\mathbf{p},\mathbf{q}) = 2\,\mathrm{sgn}(q_1 p_1 + q_2 p_2 + q_3 p_3 + q_4 p_4) \times \begin{bmatrix} -q_1 p_2 + q_2 p_1 + q_3 p_4 - q_4 p_3 \\ -q_1 p_3 - q_2 p_4 + q_3 p_1 + q_4 p_2 \\ -q_1 p_4 + q_2 p_3 - q_3 p_2 + q_4 p_1 \end{bmatrix} \tag{31}$$

The tracking error dynamics are found by directly differentiating equation (30),

$$\dot{\mathbf{e}} = \begin{bmatrix} \mathbf{v_r} - \mathbf{v} \\ \dot{\mathbf{v}}_\mathbf{r} - \dot{\mathbf{v}} \\ \boldsymbol{\omega}_\mathbf{r} - \boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}}_\mathbf{r} - \dot{\boldsymbol{\omega}} \end{bmatrix}. \tag{32}$$

Furthermore, the linear PD compensator has the following form:

$$\begin{bmatrix} \mathbf{a_{pd}} \\ \boldsymbol{\alpha}_\mathbf{pd} \end{bmatrix} = \begin{bmatrix} \mathbf{R_p} & \mathbf{R_d} & 0 & 0 \\ 0 & 0 & \mathbf{K_p} & \mathbf{K_d} \end{bmatrix} \mathbf{e} \tag{33}$$

B. The Reference Model & Pseudo Control Hedging (PCH)

Care must be taken in designing the reference model to ensure the effects which introduce $\mathbf{a_h}$ and $\boldsymbol{\alpha}_\mathbf{h}$ in equations (25) and (26) 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 the discrepancy. This can be avoided through PCH in the reference model dynamics as follows [2]:

$$\dot{\mathbf{v}}_\mathbf{r} = \mathbf{a_{cr}}(\mathbf{p_r}, \mathbf{v_r}, \mathbf{p_c}, \mathbf{v_c}) - \mathbf{a_h} \tag{34}$$

$$\dot{\boldsymbol{\omega}}_\mathbf{r} = \boldsymbol{\alpha}_\mathbf{cr}(\mathbf{q_r}, \boldsymbol{\omega}_\mathbf{r}, \mathbf{q_c} \oplus \mathbf{q_{des}}, \boldsymbol{\omega}_\mathbf{c}) - \boldsymbol{\alpha}_\mathbf{h} \tag{35}$$

where the subscript *c* refers to commands and *r* refers to the reference model and the expression $\mathbf{q_c} \oplus \mathbf{q_{des}}$ represents the combination of two quaternion vectors through quaternion multiplication. The signals $\mathbf{a_h}$ and $\boldsymbol{\alpha}_\mathbf{h}$, which represent the error between the commanded and realized pseudo-control, are given as follows:

$$\mathbf{a_h} = \hat{\mathbf{a}}(\mathbf{x}, \mathbf{q_{des,}} \boldsymbol{\delta}_{\mathbf{f_{des}}}, \hat{\boldsymbol{\delta}}_\mathbf{m}) - \hat{\mathbf{a}}(\mathbf{x}, \hat{\boldsymbol{\delta}}_\mathbf{f}, \hat{\boldsymbol{\delta}}_\mathbf{m}) = \mathbf{a_{des}} - \hat{\mathbf{a}}(\mathbf{x}, \hat{\boldsymbol{\delta}}_\mathbf{f}, \hat{\boldsymbol{\delta}}_\mathbf{m}) \tag{36}$$

$$\boldsymbol{\alpha}_\mathbf{h} = \hat{\boldsymbol{\alpha}}(\mathbf{x}, \hat{\boldsymbol{\delta}}_\mathbf{f}, \boldsymbol{\delta}_{\mathbf{m_{des}}}) - \hat{\boldsymbol{\alpha}}(\mathbf{x}, \hat{\boldsymbol{\delta}}_\mathbf{f}, \hat{\boldsymbol{\delta}}_\mathbf{m}) = \boldsymbol{\alpha}_\mathbf{des} - \hat{\boldsymbol{\alpha}}(\mathbf{x}, \hat{\boldsymbol{\delta}}_\mathbf{f}, \hat{\boldsymbol{\delta}}_\mathbf{m}) \tag{37}$$

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, in other words, when the actuators are not saturated, then the $\mathbf{a_h}$ and $\boldsymbol{\alpha}_\mathbf{h}$ terms vanish and the reference model is unaffected. In the presence of saturated actuators however, the $\mathbf{a_h}$ and $\boldsymbol{\alpha}_\mathbf{h}$ terms are nonzero and therefore are subtracted from the reference model dynamics to effectively hide the error due to saturation in the error dynamics. Substituting equations (34) – (37) into the error dynamics (32), it can be shown, as in [2], that the error dynamics become

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}[\bar{\nu}_{ad} - \bar{\boldsymbol{\Delta}}(\mathbf{x},\boldsymbol{\delta},\hat{\boldsymbol{\delta}})] \tag{38}$$

$$\bar{\nu}_{ad} = \begin{bmatrix} \bar{\mathbf{a}}_\mathbf{ad} \\ \bar{\boldsymbol{\alpha}}_\mathbf{ad} \end{bmatrix} \tag{39}$$

$$\mathbf{A} = \begin{bmatrix} 0 & I & 0 & 0 \\ -\mathbf{R_p} & -\mathbf{R_d} & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -\mathbf{K_p} & -\mathbf{K_d} \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \tag{40}$$

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

The following reference model with appropriate rate limits, as given in [2], was chosen:

$$\mathbf{a_{cr}} = \mathbf{R_p}(\mathbf{p_c} - \mathbf{p_r}) + \mathbf{R_d}(\mathbf{v_c} - \mathbf{v_r}) \tag{41}$$

$$\dot{\mathbf{v}}_\mathbf{r} = \mathbf{a_{cr}} - \mathbf{a_h} \tag{42}$$

$$\boldsymbol{\alpha}_\mathbf{cr} = \mathbf{K_p}\left\lfloor \tilde{\mathbf{Q}}(\mathbf{q_c} \oplus \mathbf{q_{des}}, \mathbf{q_r}) \right\rfloor + \mathbf{K_d}(\boldsymbol{\omega}_\mathbf{c} - \boldsymbol{\omega}_\mathbf{r}) \tag{43}$$

$$\dot{\boldsymbol{\omega}}_\mathbf{r} = \boldsymbol{\alpha}_\mathbf{cr} - \boldsymbol{\alpha}_\mathbf{h} \tag{44}$$

The four gain matrices, $\mathbf{R_{p,d}}$ and $\mathbf{K_{p,d}}$, in the above equations are chosen to result in desirable closed loop pole placement, and are the same as those in the PD compensator. Formulas for the gain values are given below [2]:

$$R_p = \frac{\omega_o^2\omega_i^2}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \tag{45}$$

$$R_d = 2\frac{\omega_o\omega_i(\zeta_o\omega_i + \omega_o\zeta_i)}{\omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2} \tag{46}$$

$$K_p = \omega_i^2 + 4\zeta_o\omega_o\zeta_i\omega_i + \omega_o^2 \tag{47}$$

$$K_d = 2\zeta_i\omega_i + 2\zeta_o\omega_o \tag{48}$$

where $\omega_{i,o}$ represent the natural frequencies for the inner and outer loops and $\zeta_{i,o}$ stand for the damping ratios. These scalar gains are placed in the diagonal gain matrices shown in equations (41) and (43).

C. The Adaptive Neural Network

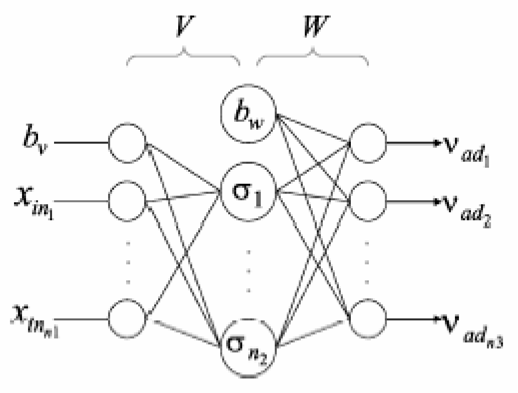


Figure 7: A neural network with a single hidden layer.

A single-hidden layer perceptron neural network (NN) was used to approximate the modeling error. As shown in Figure 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 [2]:

$$v_{ad_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j \left(z_j \right) \tag{49}$$

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

$$\sigma_j \left(z_j \right) = 1 / \left(1 + e^{-a z_j} \right) \tag{50}$$

with the neuronal activation potential given as a. For the j$^{\text{th}}$ neuron, the input is:

$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{in_i} \tag{51}$$

where b_v is the inner-layer bias, θ_{v_j} is the j$^{\text{th}}$ threshold, v_{ij} represents the weights of the inner layer, and x_{in_i} represents the NN inputs. For convenience, the following NN weight matrices are introduced:

$$\mathbf{V} = \begin{bmatrix} \theta_{v,1} & \cdots & \theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_2,1} & \cdots & v_{n_1,n_2} \end{bmatrix} \tag{52}$$

$$\mathbf{W} = \begin{bmatrix} \theta_{w,1} & \cdots & \theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{bmatrix} \tag{53}$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{V} & 0 \\ 0 & \mathbf{W} \end{bmatrix} \tag{54}$$

Furthermore, the following vectors are defined

$$\boldsymbol{\sigma}^T(\mathbf{z}) = \begin{bmatrix} b_w & \sigma(z_1) & \cdots & \sigma(z_{n_2}) \end{bmatrix} \tag{55}$$

$$\bar{\mathbf{x}}^T = \begin{bmatrix} b_v & \mathbf{x}_{in}^T \end{bmatrix}, \ \mathbf{x}_{in}^T = \begin{bmatrix} \mathbf{x}_c^T & \mathbf{e}_r^T & \mathbf{e}^T & \nu_{ad}^T & \|\mathbf{Z}\|_F \end{bmatrix} \tag{56}$$

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

Using the above notation, the output of the neural network is given by the following equations [2]:

$$\mathbf{r} = \left(\mathbf{e}^T \mathbf{P}\mathbf{B}\right)^T \tag{57}$$

$$\bar{\mathbf{v}}_{\mathbf{ad}} = \mathbf{v}_{\mathbf{ad}} + \mathbf{v}_{\mathbf{r}} \tag{58}$$

$$\mathbf{v}_{\mathbf{ad}} = \mathbf{W}^T \boldsymbol{\sigma}\left(\mathbf{V}^T \bar{\mathbf{x}}\right) \tag{59}$$

$$\mathbf{v}_{\mathbf{r}} = -K_r\left(\|\mathbf{Z}\|_F + \bar{Z}\right)\left(\|\mathbf{e}\| / \|\mathbf{r}\|\right)\mathbf{r} \tag{60}$$

where the matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \tag{61}$$

$$\mathbf{P}_1 = \begin{bmatrix} R_p^2 + \frac{1}{2}R_p R_d^2 & \frac{1}{2}R_p R_d \\ \frac{1}{2}R_p R_d & R_p \end{bmatrix} > 0 \tag{62}$$

$$\mathbf{P}_2 = \begin{bmatrix} K_p^2 + \frac{1}{2}K_p K_d^2 & \frac{1}{2}K_p K_d \\ \frac{1}{2}K_p K_d & K_p \end{bmatrix} > 0 \tag{63}$$

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

$$\dot{\mathbf{W}} = -\left[\left(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}}\right)\mathbf{r}^T + \kappa\|\mathbf{e}\|\mathbf{W}\right]\boldsymbol{\Gamma}_{\mathbf{w}} \tag{64}$$

$$\dot{\mathbf{V}} = -\boldsymbol{\Gamma}_{\mathbf{v}}\left[\bar{\mathbf{x}}\left(\mathbf{r}^T\mathbf{W}^T\boldsymbol{\sigma}'\right) + \kappa\|\mathbf{e}\|\mathbf{V}\right] \tag{65}$$

where $\boldsymbol{\Gamma}_{\mathbf{w}}$ and $\boldsymbol{\Gamma}_{\mathbf{v}}$ are positive definite matrices and κ is a positive scalar. The $\boldsymbol{\sigma}'$ matrix is the Jacobian of the $\boldsymbol{\sigma}$ vector and the **B** matrix comes from the error dynamics in equation (38). The $\bar{Z}$ value is a bound on the norm of the **Z** matrix corresponding to the ideal NN weighting matrices in accordance with the universal approximation property.

In [2], it has been proven that Lyapunov analysis carried over the error dynamics in Equation (38) using these weight adaptation rules given in (64) and (65) will result in the ultimate boundedness of the tracking error **e** and the NN weights.

D. The Approximate Dynamic Model

This method requires an invertible, approximate model of the vehicle and the neural network is used to compensate for the inaccuracies. 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 airplane linearized about a stationary hover are as follows [2]:

$$\boldsymbol{\alpha}_{\mathbf{des}} = \hat{\mathbf{A}}_{\mathbf{1}}\boldsymbol{\omega} + \hat{\mathbf{A}}_{\mathbf{2}}\mathbf{v} + \hat{\mathbf{B}}\left(\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{des}}} - \boldsymbol{\delta}_{\mathbf{m}_{\mathbf{trim}}}\right) \tag{66}$$

where $\hat{\mathbf{A}}_{\mathbf{1}}$ and $\hat{\mathbf{A}}_{\mathbf{2}}$ describe the attitude and translational dynamics and $\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{trim}}}$ is the trim control vector. The equation can be solved for the moment control vector $\boldsymbol{\delta}_{\mathbf{m}_{des}}$ if the control effectiveness $\hat{\mathbf{B}}$ is invertible. To obtain the translational dynamics, the vehicle was modeled as a point mass with a thrust vector that can be directed in any direction. The dynamics of the model are as follows:

$$\mathbf{a}_{\mathbf{des}_b} = \begin{bmatrix} X_{\delta_{thr}} \\ 0 \\ 0 \end{bmatrix} \left(\delta_{f_{des}} - \delta_{f_{trim}}\right) + T_{i->b}\mathbf{g} \tag{67}$$

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

$$f_{sf} = (\mathbf{a}_{des} - T_{i->b}\mathbf{g})_1 \tag{68}$$

$$\delta_{f_{des}} = f_{sf} / X_{\delta_{thr}} + \delta_{f_{trim}} \tag{69}$$

The attitude changes necessary to align the thrust vector in the proper direction is given by the small angle corrections as shown in Figure 8 and the following equations [2]:

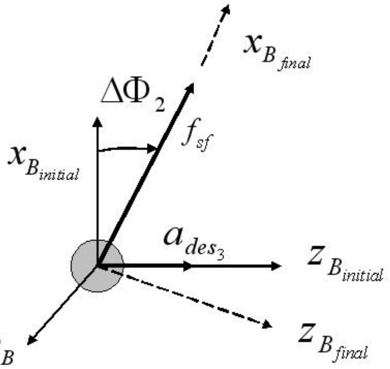


Figure 8: A point mass model is used for the outer-loop approximate inversion. This diagram illustrates a small-angle correction for a pitch attitude change.

$$\Delta\Phi_1 = 0 \tag{70}$$

$$\Delta\Phi_2 = -a_{des_3} \big/ f_{sf} \tag{71}$$

$$\Delta\Phi_3 = a_{des_2} \Big/ f_{sf} \quad (72)$$

where $\Delta\Phi_1$ is zero since changes in roll attitude have no effect on the direction of the thrust vector. These can then be converted to a quaternion correction as follows:

$$\mathbf{q_{des}} = \mathbf{q}(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3) \quad (73)$$

where $\mathbf{q}(\cdot)$ is the transformation from a 3-2-1 Euler angle rotation to a quaternion.

E. The Hover-Transition Guidance Logic

The trajectory generator in Figure 6 provides a commanded trajectory to the adaptive control system. For transitions from forward flight to stationary hover, two guidance schemes have been tested. The first scheme involved a gradual (linear) ramping of the commanded pitch angle from its forward flight trim condition, to a commanded pitch angle of 90° while simultaneously decreasing (again, linearly) the commanded velocity over a period of 10 seconds. The second scheme involved a step change in the commanded pitch angle from trim to 120° and then back to 90° as the aircraft decelerates to zero forward velocity. This faster transition also involved a linear decrease in commanded velocity over a period of 5 seconds. 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 flight test experience. Results for each of these transition schemes are shown below.

IV. Flight Test Results

To date, eight successful transitions from forward horizontal flight, to a stationary hover, and then back to forward flight under autonomous control have been recorded using the GTEdge (see Figure 9.) while operating in varying conditions. The first of these successful flights was achieved on July 29, 2005. Flight performance verification was initially carried out for this vehicle by having the airplane perform circular orbits at a moderate groundspeed of 80 ft/s, and then performing successive orbits at increasingly slower ground speeds, in decrements of 10 ft/s, to ensure proper operation of the controller throughout the flight envelope. This process was repeated until the airplane could operate at zero commanded ground speed.

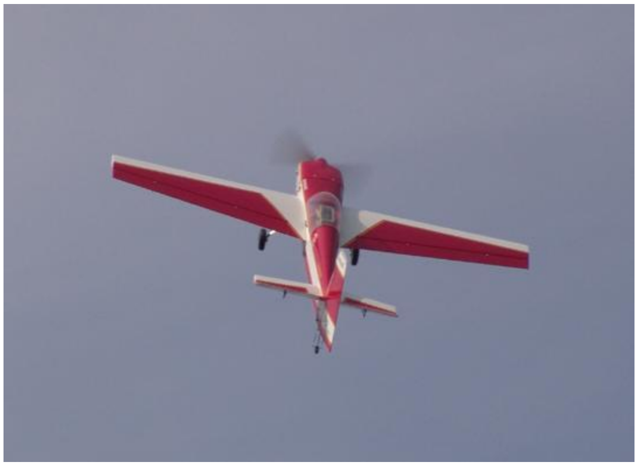
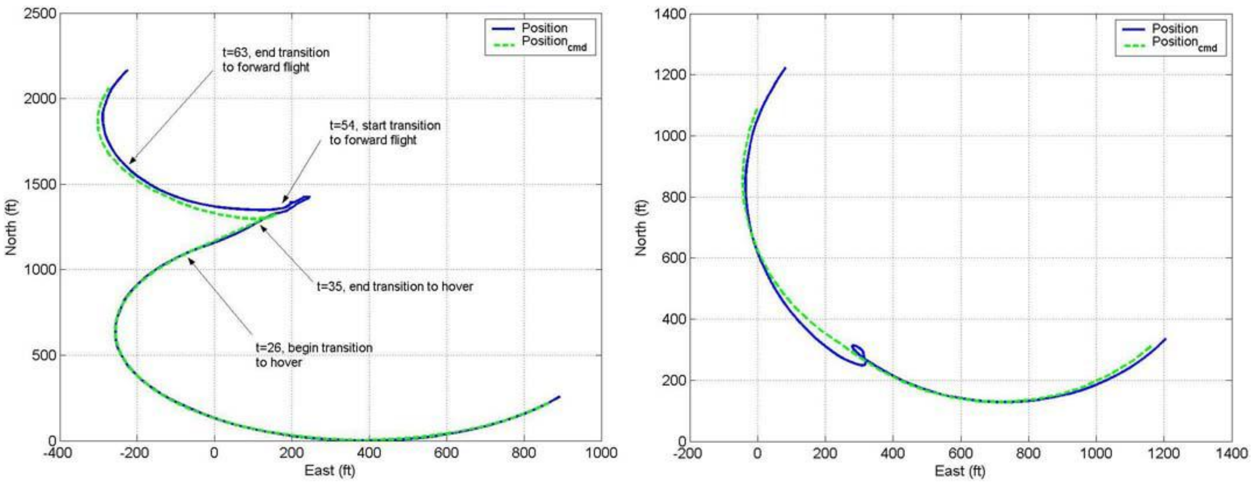


Figure 9. For a stationary hover, the GTEdge is commanded to zero ground speed and attains a near vertical attitude. In this configuration, the aircraft modulates throttle to maintain altitude, and uses the control surfaces to offset the corresponding engine torques.

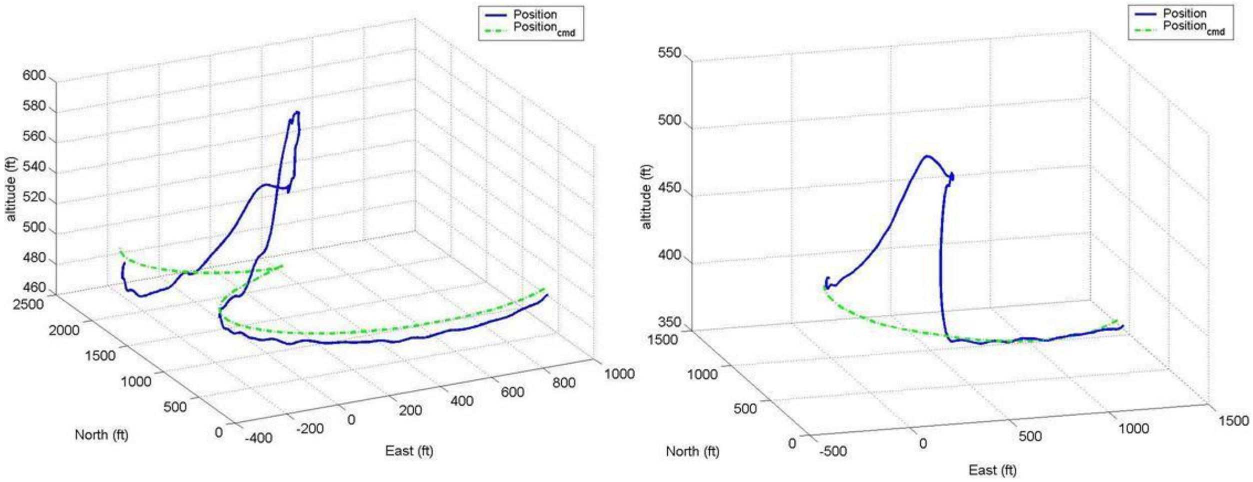
This section presents data sets from two separate instances, recorded on separate days, in which the GTEdge performed these transitions under dissimilar flight conditions. In the first set of recorded data, the flight test was initiated with the airplane performing a circular orbit at 80 ft/s, and the vehicle then transitions into a hover by ramping the ground speed down to 0 ft/s over a period of 9 seconds 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 change in commanded pitch angle). Stationary hover was sustained for 19 seconds 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 at a more rapid pace with the ground speed being ramped down to 0 ft/s over a duration of 5 seconds. Hover was held for 10 seconds, and then the aircraft transitions back to forward flight at the same slower rate as in the first data set. Air data was recorded during the second flight test, but was unavailable during the slower transition flight. Using the recorded air data, it was found that the wind during the faster transition was approximately 30 ft/s pointing in the southeast direction. Wind direction for the first data set was observed to be towards the southwest with an unknown magnitude. During both flights, the transition to hovering flight was initiated during the upwind leg of the circular orbit.



(a) slower transition

(b) faster transition

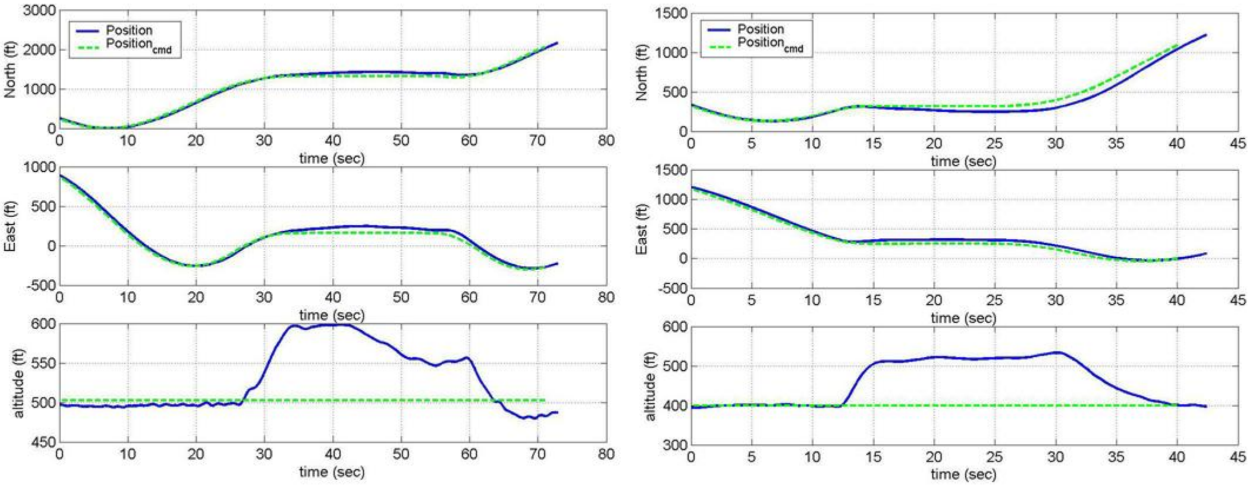
Figure 10: Ground track during transitions to and from hover. In both (a) and (b), the aircraft is flying in the clockwise direction. In the slower transition, the airplane begins a new orbit after transitioning to horizontal flight, whereas in the faster transition, it continues along the same circular orbit. This discrepancy was due to modifications in the guidance laws.



(a) slower transition

(b) faster transition

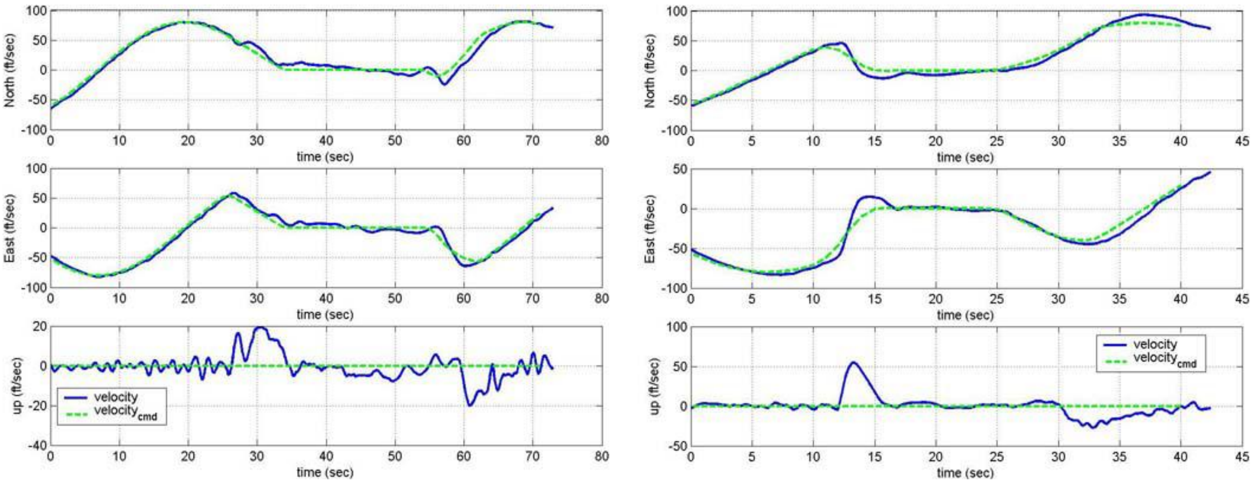
Figure 11: Three-dimensional trajectory of the GTEdge during the transitions to and from hovering flight. In both (a) and (b), the aircraft is flying in the clockwise direction.



(a) slower transition

(b) faster transition

Figure 12: Commanded and estimated actual positions of the aircraft during hovering flight The maximum position errors in (a) were [102.2 89.6 95.8] ft in the North-East-up directions and [112.5 69.0 133.4] ft in (b).

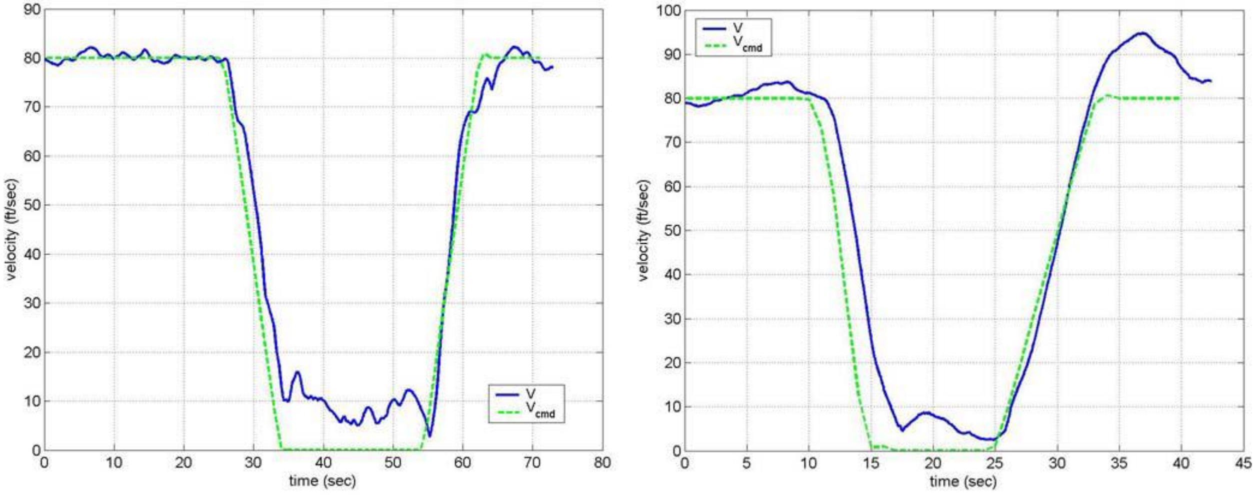


(a) slower transition

(b) faster transition

Figure 13. Commanded and estimated actual velocities of the aircraft in the North, East, and up directions. The maximum position errors in (a) were [17.9 18.4 20.1] ft/s in the North-East-up directions and [26.0 39.0 54.8] ft/s in (b).

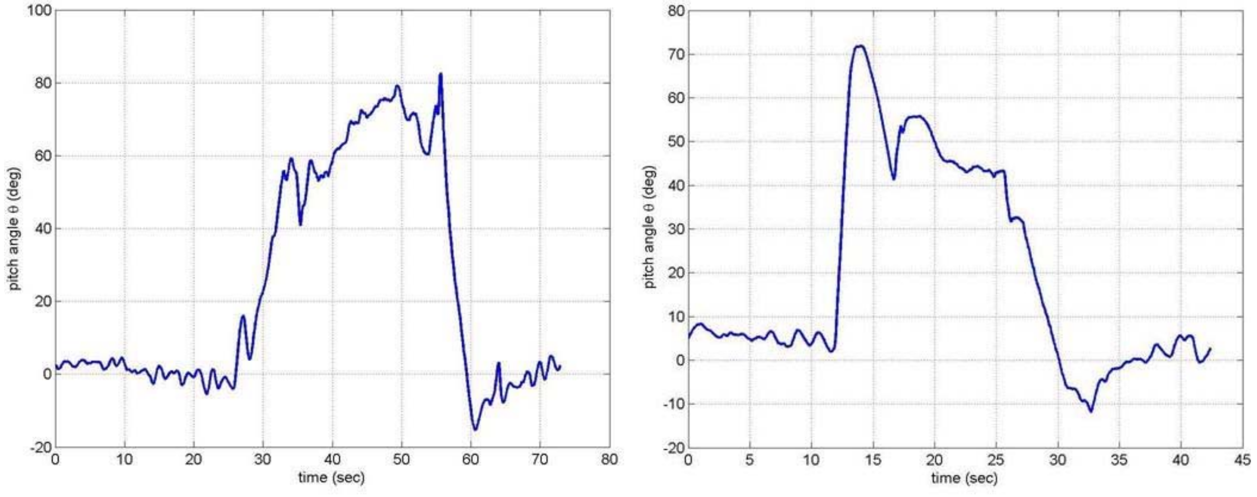
Figure 10 shows the commanded and estimated ground tracks while Figure 11 and Figure 12 show the three-dimensional trajectory of the aircraft throughout the maneuvers. By comparing the shapes of the position and velocity curves in Figure 12(a) and Figure 13(a), it can be seen that in the slower transition, the GTEdge actually overshoots the commanded hovering position and gradually corrects for this error. The North and East velocities in Figure 13(a) change sign as would be expected to correct for the overshoot. These figures also show that in the faster transition, the aircraft actually 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 more rapid transition exhibit a climb in altitude of about 100 feet during the transition to hover since the aircraft has to pull up in order to attain the near vertical attitude required for low-speed flight. The maximum altitude errors for the slower and faster transitions are 95.8 ft and 133.4 ft respectively. During the hover 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 centered at a new location as can be seen by the shape of the trajectory in Figure 10(a). The guidance law was later changed so that the airplane would continue along the same circular path upon exiting the hover as in Figure 10(b).



(a) slower transition

(b) faster transition

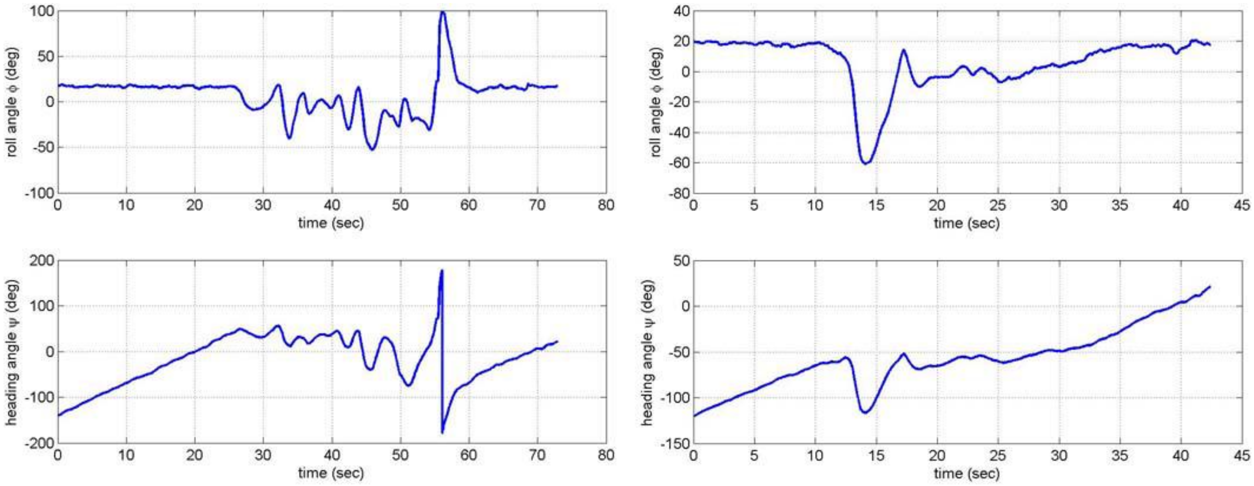
Figure 14. Commanded and estimated total speeds for the airplane relative to the ground during the transition to and from hovering flight. These values are computed by taking the norm of the overall velocity vector. The airplane is commanded to a ground velocity of 80 ft/s in steady flight, and a ground velocity of 0 ft/s during hovering flight. The slower transition between steady flight and hover in (a) take approximately 9 seconds each. In the faster transition in (b), the transition to hovering flight takes 5 seconds and the transition back to horizontal flight was performed at the rate and takes 9 seconds to complete.



(a) slower transition

(b) faster transition

Figure 15. Pitch angle for the GTEdge during the transition to and from hover. In the slower transition (a), the maximum pitch angle is 83° and the minimum is -15°. In the faster transition (b), the maximum pitch angle is 72° and the minimum angle is -12°.



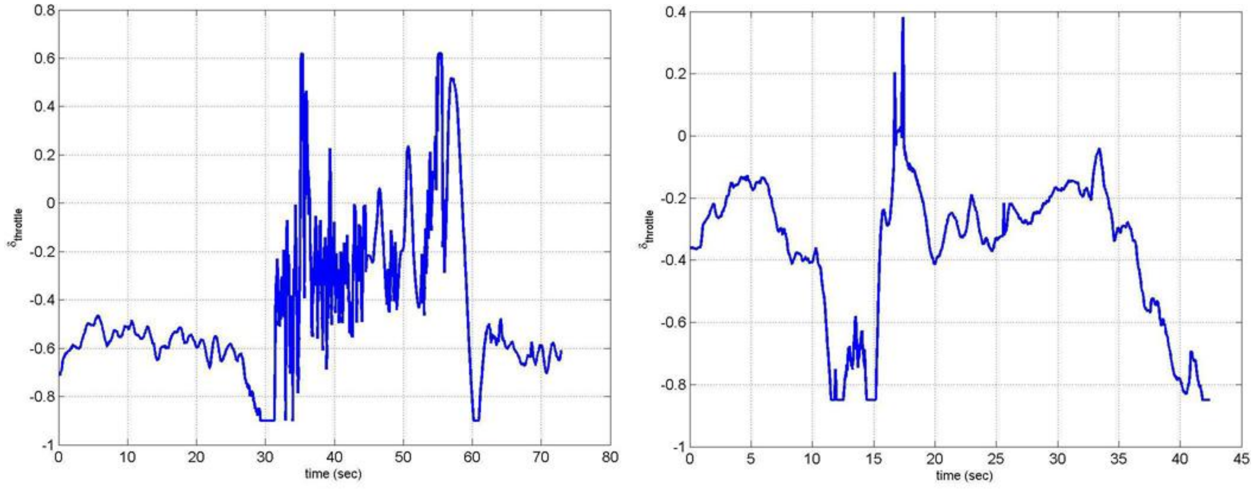
(a) slower transition

(b) faster transition

Figure 16. 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 since the aircraft is in a steady turn. During the stationary hover in the slower transition (a), the airplane performs a three-quarter revolution roll as the transition back to forward flight is initiated as can be seen from the wrap around in the heading angle.

The velocity profile for the GTEdge throughout the maneuvers in both of the described test cases is presented in Figure 14. These plots show that the GTEdge never actually achieves zero velocity relative to the ground. Furthermore, the controller is able to track the velocity profile more closely during the slower transitions than during the faster transition from forward flight into hover as shown in Figure 14. Figure 15 and Figure 16 show vehicle attitude expressed as Euler angles with Figure 15 giving the pitch angle and Figure 16 showing the roll and heading angles. These plots illustrate that the GTEdge never reaches a fully vertical attitude since the pitch angle remains below 90 degrees throughout the stationary hover portion of the maneuver. This is an expected result since the vehicle was operating in the presence of wind, so the aircraft must have its thrust vector at an angle relative to the vertical to counteract these forces. The pitch angle plots also show that the airplane attains a higher pitch angle in the slower transition to hover than when it is commanded to transition more rapidly. This phenomenon, however, may have been a result of differing wind conditions between the two flight tests.

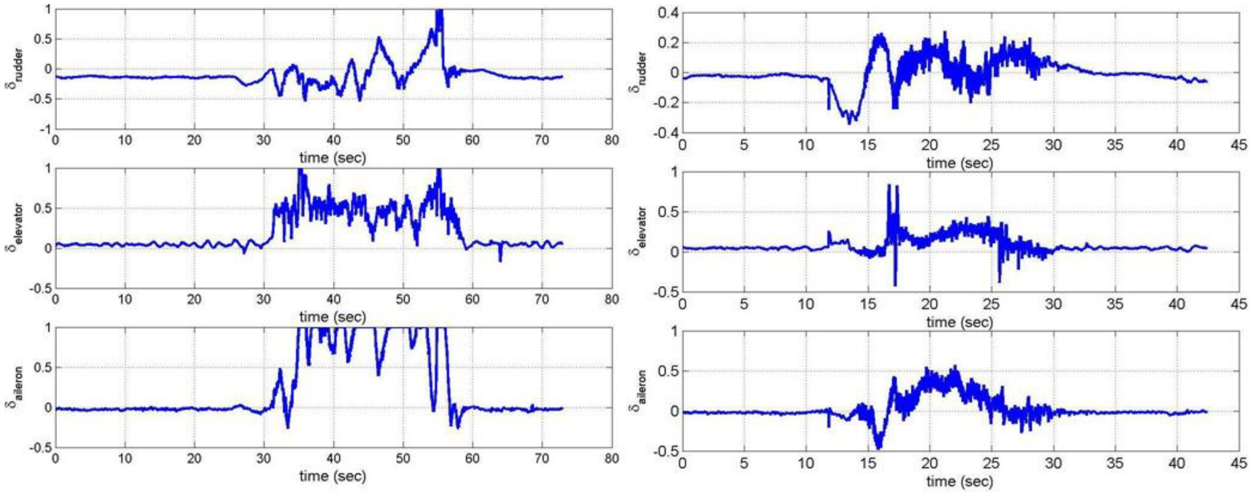
Here, we note that the returning transition from a stationary hover to forward flight is marked by the rapid descent in pitch angle in Figure 15. As the pitch angle decreases, the GTEdge descends in altitude and begins to pick up speed until it reaches steady horizontal flight at 80 ft/s. One other interesting point that is evident from the attitude information for the slower transition presented in Figure 16(a) was 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 second marker. This was due to the trajectory commanded to the aircraft by the guidance law as it exits the hover.



(a) slower transition

(b) faster transition

Figure 17. Throttle actuator commands during the transitions to and from hovering flight. The throttle settings for the two presented cases 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 mid-flight. The maximum throttle command in (a) is 0.62 and the maximum in (b) is 0.38.



(a) slower transition

(b) faster transition

Figure 18. Aerodynamic control surface deflections during steady flight and hovering flight. Control surface deflection values can range from values of -1.0 to 1.0 which represent minimum and maximum saturation values. Deflection values are relatively level during straight-and-level flight and are comparatively erratic during hovering flight. During the more rapid transition in (b), the actuators never reach their limit values. The slower transition in (a), 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.

Actuator commands are shown in Figure 17 and Figure 18 with Figure 17 showing throttle and Figure 18 showing the deflections for the rudder, elevator, and ailerons. These control deflections are expressed on a dimensionless scale ranging from -1.0 to 1.0 which represent minimum and maximum saturation values. The pitch angle plot (Figure 15) and the throttle command plot (Figure 17) illustrate that the transition to hover is described by reduction in throttle down to the lower saturation limit as the commanded velocity ramps down, and then 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 in order to maintain altitude. However, the resulting rolling moment imparted on the aircraft needs to be counteracted by the ailerons 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 often insufficient to produce the aerodynamic moments required to counteract the engine torque. This is further illustrated by the repeated saturation of the ailerons in Figure 18(a) 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 Figure 18(b). This may have been because the net airspeed due to the high winds provided sufficient airflow over the ailerons for sufficient control authority.

V. Conclusions

The results presented in this paper demonstrate the capability of a maneuverable fixed-wing UAV to autonomously transition from forward flight to stationary hover and 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. The inclusion of pseudo-control hedging allowed for appropriate adaptation and better vehicle control in aggressive maneuvers and hover where actuator saturation would otherwise become problematic. During flight testing, the controller was found to perform well in both normal operations and in the hover regime.

It has been determined that the guidance scheme used to transition between forward and hovering flight significantly affects performance in the transient regime. Both slow and fast transitions to hover have been flight tested with the faster transitions allowing slightly more control of commanded hovering position during the transient. Future work will include integrating a scheme for bleeding off excess airspeed prior to the maneuver which should significantly reduce the "ballooning" in the 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 allow for fast moving fixed-wing vehicles capable of quickly reaching a destination, hovering in place to obtain surveillance, and then quickly exiting and returning to land. More sophisticated trajectory generation techniques could also be applied to this controller to widen the performance envelope of this aircraft to include aggressive aerobatic maneuvers.

Acknowledgments

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. This work was supported in part by: AeroVironment Inc., Guided Systems Technologies, and the Active-Vision Control Systems (AVCS) Multi-University Research Initiative (MURI) Program under contract #F49620-03-1-0401.

References

[1] Johnson, E. N., "Limited Authority Adaptive Flight Control," Ph.D. Thesis, Georgia Institute of Technology, 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 99-3996, 1999, pp. 1-2.

[4] Green, W. E., Oh, P. Y., "Autonomous Hovering of a Fixed-Wing Micro Air Vehicle",
http://prism2.mem.drexel.edu/~billgreen/Publications/finalGreenIcra2006.pdf, Drexel University, Philadelphia, PA.

[5] Green, W. E., Oh, P. Y., "A Micro Air Vehicle to Fly in Caves, Tunnels, and Forests"
URL: http://www.pages.drexel.edu/~weg22/fwHovering/fixedWingHovering.html.

[6] AFRL Website: http://www.afrl.af.mil/accomprpt/may04/accompmay04.asp.

[7] Aviation Week & Space Technology "Aerospace SourceBook 2006".

[8] Aurora Flight Sciences Website: http://www.aurora.aero/GE50/index.html.

[9] International Online Defense Magazine: http://www.defense-update.com/products/g/goldeneye50.htm.

[10] Stone, H., Wong, K.C. "Preliminary Design of a Tandem-Wing Tail-Sitter UAV Using Multi-Disciplinary Design Optimisation ", *International Aerospace Congress*, Sydney, February 1997, p707-720

[11]Stone, H., Clarke, G. “Optimization of Transition Maneuvers for a Tail-Sitter Unmanned Air Vehicle (UAV)”, *Australian International Aerospace Congress*, Paper 105, Canberra, March 2001.

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

[13]Christophersen, H., Pickell, W., Koller, A., Kannan, S., Johnson, E., “Small Adaptive Flight Control System for UAVs using FPGA/DSP Technology,” *Proceedings of the AIAA Unmanned Unlimited Conference*, Chicago, IL, 2004.

[14]Dittrich, J., and Johnson, E. N., “Multi-Sensor Navigation System for an Autonomous Helicopter,” *AIAA/IEEE Digital Avionics Systems Conference*, 2002.

[15]Dittrich, J, and Johnson, E., “Design and Integration of an Unmanned Aerial Vehicle Navigation System”, A Thesis, School of Aerospace Engineering, Georgia Institute of Technology, May 2002.

[16]Stevens, B., Lewis, F., *Aircraft Control and Simulation*, Wiley and Sons, 2004.

[17]Gelb, A, et. al., *Applied Optimal Estimation*, The M.I.T. Press, 1974.

[18]Brown, G., Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering*, 2nd Edition, John Wiley & Sons, 1992.

[19]Bletzacker, et. al., “Kalman filter design for integration of Phase III GPS with an inertial navigation system, “ Computing Applications Software Technology Technical Papers, Los Alamitos, CA, 1988.

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

[21]Avanzini, G., Boserman, F., Ciniglio, U., and de Matteis, G., “Design of a Rate Commanded μ-Controller for a Shrouded-Fan Uninhabited Aerial Vehicle,” *AIAA Guidance, Navigation, and Control Conf.*, AIAA 2003-5521, Aug. 2003.

[22]Avanzini, G., de Matteis, G., and Fresta, F., “Robust Multivariable Control of a Shrouded-Fan Uninhabited Aerial Vehicle,” *AIAA Atmospheric Flight Mechanics Conf.*, AIAA 2002-4703, Aug. 2003.

[23]Franz, R., Milam, M., and Hauser, J., “Applied Receding Horizon Control of the Caltech Ducted Fan,” *American Control Conf.*, May 2002.

[24]Hess, R. A. and Ussery, T. M., “Sliding Mode Techniques Applied to the Control of a Micro-Air Vehicle,” *AIAA Guidance, Navigation, and Control Conf.*, AIAA 2003-5408, Aug. 2003.

[25]Kannan, S. K., “Adaptive Control of Systems in Cascade with Saturation,” Ph.D. Thesis, Georgia Institute of Technology, 2005.