Modeling, Control, and Flight Testing of a

Small Ducted Fan Aircraft

Eric N. Johnson[*] and Michael A. Turbe[†]

Georgia Institute of Technology, Atlanta, GA, 30332

Small ducted fan autonomous vehicles have potential for several applications, especially for missions in urban environments. This paper discusses the use of dynamic inversion with neural network adaptation to provide an adaptive controller for the GTSpy, a small ducted fan autonomous vehicle based on the Micro Autonomous Systems' Helispy. This approach allows utilization of the entire low speed flight envelope with a relatively poorly understood vehicle. A simulator model is constructed from a force and moment analysis of the vehicle, allowing for a validation of the controller in preparation for flight testing. Data from flight testing of the system is provided.

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
a_0	=	rotor blade lift curve slope
b	=	number of rotor blades
b_v, b_w	=	neural network (NN) biases
C_D, C_d	=	drag coefficients
C_{D_o}	=	rotor drag coefficient
C_{duct}	=	duct moment coefficient
C_L, C_l	=	lift coefficients
C_{l_α}	=	airfoil lift curve slope
c	=	chord
D, L	=	drag, lift

[*] Lockheed Martin Assistant Professor of Avionics Integration, AIAA member.

[†] Graduate student, AIAA student member.

d = drag per unit span

$\mathbf{e}$ = reference model tracking error

$\hat{\mathbf{e}}_\mathbf{r}$ = radial unit vector

$\mathbf{F}, \mathbf{M}$ = force, moment vectors

f_r = rotor friction term

f_{sf} = specific force along vehicle z axis

$g, \mathbf{g}$ = gravitational acceleration, vector

$\mathbf{H}$ = angular momentum vector

$\mathbf{I}$ = vehicle inertia matrix

$\hat{\mathbf{i}}, \hat{\mathbf{j}}$ = unit vectors along body x, y axes

i_b = rotor blade moment of inertia

$\mathbf{K}, \mathbf{R}$ = inner-loop, outer-loop gain matrices

K_{bhp} = engine horsepower

K_{dr} = rotor gear reduction ratio

$K_{rps,\max}$ = max engine revolutions per second

K_{time} = engine time constant

K_{twist} = rotor blade twist

$\mathbf{L_{bv}}$ = local-to-body direction cosine matrix

l = lift per unit span or moment arm

m = vehicle mass

$\dot{m}$ = mass flow through rotor

n_1, n_2, n_3 = number of NN inputs, neurons, outputs

P = power

$\mathbf{p}$ = position vector

$\widetilde{\mathbf{Q}}(\,\cdot\,)$ = attitude error angle function

$\mathbf{q}$ = attitude quaternion

$\mathbf{q}(\,\cdot\,)$ = Euler rotation to quaternion transform

q = dynamic pressure

S = planform area

$r, \mathbf{r}$ = rotor radius, filtered tracking error

$\mathbf{V}, \mathbf{W}, v, w$ = NN input and output weights

v, V = velocity

v' = farfield velocity

v_b = flow velocity past rotor

v_i = flow velocity induced through rotor

$x, \mathbf{x}$ = state variable, state vector

$x_{in}, \overline{\mathbf{x}}$ = NN input

Z = control derivative

z_{aero} = aerodynamic drag moment arm

z_j = input to j[th] hidden-layer neuron

α = angular acceleration or angle of attack

$\boldsymbol{\alpha}(\,\cdot\,), \hat{\boldsymbol{\alpha}}(\,\cdot\,)$ = attitude dynamics and estimate

$\boldsymbol{\Gamma}_\mathbf{v}, \boldsymbol{\Gamma}_\mathbf{w}$ = NN learning rate matrices

γ = downwash

$\overline{\boldsymbol{\Delta}}(\,\cdot\,)$ = total function approximation error

$\Delta\Phi$ = attitude correction

$\delta, \hat{\delta}$ = actuator deflection and estimate

ζ	=	damping ratio	c	=	commanded
η_e	=	engine efficiency	cr	=	reference model dynamics
$\theta, \theta_v, \theta_w$	=	polar coordinate, NN thresholds	d	=	derivative, duct
			des	=	desired
κ	=	E-modification parameter	e	=	elevator, engine
$\mathbf{v_{ad}}, \mathbf{\bar{v}_{ad}}$	=	adaptive element signals	f	=	force
$\mathbf{v_r}$	=	robustifying signal	h	=	hedge
ρ_∞	=	freestream density	i,o	=	inner loop, outer loop
			$ind, prof$	=	induced, profile
$\mathbf{\sigma}, \mathbf{\sigma'}$	=	neuron sigmoidal function, gradient	lat, lon	=	lateral, longitudinal
τ	=	rotor thrust	m	=	moment
ω	=	angular velocity or natural frequency			
			off	=	offset
			p	=	proportional
Subscripts			r	=	reference model, rotor, rudder
a	=	aileron			
			thr	=	throttle
ad	=	adaptive			

I. Introduction

Ducted fan vehicles offer several advantages as autonomous uninhabited aerial vehicles (UAVs). They can be very small with a compact layout. Many are capable of high-speed flight in addition to the normal hover and vertical takeoff and landing capabilities. These features make them well-suited for a variety of missions, especially in urban environments. This demand has led to the development of several of these vehicles, such as the GoldenEye, Kestrel, and iSTAR[1,2]. The GTSpy is one of the smallest entries in the rapidly growing field of small ducted fan vehicles.

The control of a small, autonomous ducted fan aircraft presents many unique challenges. Ducted fan vehicles are usually highly unstable with complex aerodynamics. Therefore, successful controllers must be very robust to deal with the uncertainty present in the available models of the vehicle dynamics. 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 GTSpy 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 allows the GTSpy to make full use of its low speed flight envelope. This type of neural network adaptive control has been used successfully in similar projects[3-5].

Another challenge facing the development of successful UAVs of this type is modeling. A reasonably accurate model is helpful for simulation, especially in the early phases of design and flight testing. An accurate simulator allows designers to estimate the performance of their design or examine the effect of various layout configurations. It also allows the safety pilots to begin familiarizing themselves with vehicle handling preceding flight testing. The performance of the controller during various maneuvers can also be evaluated in order to mitigate the risks to the actual vehicle during flight tests. The simulation model for the GTSpy was developed by analyzing the various forces and moments acting on the vehicle. Such an approach forms external force and moment vectors by determining the contribution of various effects, such as drag, gravity, etc. Each of these effects is analyzed in turn to determine the contributing forces and moments, which are then added to the total force and moment vectors. This offers the benefit of modularity because any force or moment component can be readily added, removed, or changed.

This paper provides a basic model for a ducted fan vehicle and the adaptive controller architecture used to successfully fly the GTSpy. A description of the GTSpy is given first, followed by the formulation of the vehicle model used in simulation. A detailed discussion of the GTSpy controller follows, as well as a brief description of the simulation environment. Finally, the flight and simulation results are presented, followed by a discussion of the present architecture's limitations and suggestions for improving controller performance.

II. Vehicle Description

The GTSpy is a modified version of the Helispy, which was designed and manufactured by Micro Autonomous Systems. The changes involved the onboard electronics, so the external appearance of both vehicles is nearly identical. The GTSpy, shown in Fig. 1, is 27 inches high, with a maximum takeoff weight of 5.5 pounds. It is powered by a small engine driving a two-bladed, fixed propeller. The propeller is enclosed within an annular wing

duct with an outer diameter of 11 inches. The GTSpy features both helicopter and airplane flight modes. It takes off, lands, and hovers in the helicopter mode in which the propeller provides direct thrust to support the vehicle. The vehicle can translate at low speeds in this configuration by tilting the thrust vector. In airplane mode, the vehicle is tilted over so that the centerline is almost horizontal. The propeller creates forward thrust while the duct creates lift, allowing the GTSpy to move at up to 60 mph. Six vanes located radially at the end of the duct are immersed in the outflow. These vanes move together in the same sense to rotate the vehicle about its centerline. Two pairs of control surfaces are also located at the end of the

Figure 1: The GTSpy Small Ducted Fan Aircraft

fuselage. These lift-generating surfaces work in pairs to create moments that rotate the vehicle in orthogonal planes. In the helicopter mode, the vanes provide yaw control, and the control surface pairs provide pitch and roll control. In the following sections, the body axes were chosen based on the helicopter mode. Therefore, the z axis is aligned with the fuselage, pointed downwards. The x and y axes form a plane parallel to the duct face. Since the vehicle is axisymmetric, the x axis was arbitrarily fixed, with the y axis then being defined dextrally.

The GTSpy is equipped with a processor board and a sensor board. The processor board features a Texas Instruments DSP chip and an Altera Field Programmable Gate Array chip. The sensor board has 3 Microelectromechanical Systems (MEMS) rate gyros and 4 MEMS +/- 10 g accelerometers. A GPS system is also included, along with two wireless data links, one for communication with the ground station and one for the safety pilot. A 380 line color CCD board camera with a 1200 MHz analog video transmitter/receiver is also installed.

III. Modeling

A model of the GTSpy was created for use in simulation to verify appropriate controller behavior. The following sections detail the formulation of this model, with the numerical values for the GTSpy listed in the Appendix. Since the main focus for the simulation was controller verification, it was of primary importance to model the major forces and moments acting on the vehicle. Exactly predicting these forces and moments was of secondary importance for this model, as long as the forces and moments exhibited the proper functional dependence

and were of the correct sense and order of magnitude. Therefore, simplifying assumptions were made in certain areas because the required level of fidelity did not justify the added complexity.

The basic dynamical equations are given as:

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

$$\dot{\mathbf{v}} = \frac{\mathbf{F}}{m} \tag{2}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{bmatrix} \boldsymbol{\omega} \tag{3}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\mathbf{M} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\right) \tag{4}$$

in which $\mathbf{p} \in \mathfrak{R}^3$ represents the position vector, $\mathbf{v} \in \mathfrak{R}^3$ is the velocity vector, $\mathbf{q} \in \mathfrak{R}^4$ is the quaternion vector, and $\boldsymbol{\omega} \in \mathfrak{R}^3$ represents the angular velocity vector. The vehicle mass is represented by m, q_i represents the four components of the quaternion vector[3], $\mathbf{I}$ is the vehicle inertia matrix, and the $\mathbf{F}$ and $\mathbf{M}$ terms represent the sum of external forces and moment vectors acting on the vehicle. The force and moment vectors can be expressed as:

$$\mathbf{F} = \mathbf{F}_{\text{gear}} + \mathbf{F}_{\text{aero}} + \mathbf{F}_{\text{rotor}} + \mathbf{F}_{\text{duct}} + \mathbf{F}_{\text{cs}} + \mathbf{F}_{\text{grav}} \tag{5}$$

$$\mathbf{M} = \mathbf{M}_{\text{gear}} + \mathbf{M}_{\text{aero}} + \mathbf{M}_{\text{rotor}} + \mathbf{M}_{\text{duct}} + \mathbf{M}_{\text{cs}} + \mathbf{M}_{\text{gyro}} \tag{6}$$

where all the component forces and moments are discussed below.

A. Landing Gear & Fuselage Aerodynamics

The landing gear force and moment vectors are due to the interaction of the landing gear with the simulated terrain. The forces and moments due to fuselage aerodynamic drag can be written as:

$$\mathbf{F}_{\text{aero}} = -\frac{1}{2}\rho_\infty \begin{bmatrix} C_{D,x} v_x |v_x| \\ C_{D,y} v_y |v_y| \\ C_{D,z} v_z |v_z| \end{bmatrix} S \tag{7}$$

$$\mathbf{M_{aero}} = \frac{1}{2}\rho_\infty \begin{bmatrix} C_{D,y}v_y|v_y| \\ -C_{D,x}v_x|v_x| \\ 0 \end{bmatrix} S z_{aero} \tag{8}$$

where ρ_∞ represents the freestream density, the C_D terms represent the x, y, z drag coefficients of the vehicle, the S term represents the vehicle planform area, the v terms represent the components of the velocity vector of the vehicle with respect to the air, expressed in the body-fixed frame, and z_{aero} represents the distance between the vehicle aerodynamic center and the center of gravity. Note that Eq. (8) implies that distance between the vehicle aerodynamic center and the center of gravity along the y and x axes is zero. This is a result of the vehicle being axisymmetric.

B. Rotor

The equations governing the rotor model are based on basic momentum theory and blade element theory. While these theories were developed for larger aspect ratio helicopter blades and do not account for the unsteady flow in the duct, they do provide a sufficiently accurate prediction of the rotor behavior for this application. More advanced methods would be necessary if more accurate performance predictions were required of the model.

The flow through the rotor is caused by the airflow along the z axis (v_z) and the motion and geometry of the blades. It can be expressed as[6]:

$$v_b = v_z + \frac{2}{3}\omega_r r\left(\frac{3}{4}K_{twist}\right) \tag{9}$$

where ω_r represents the angular velocity of the rotor, r is the radius of the rotor, and K_{twist} is the twist of the blades. The thrust can now be written as[6]:

$$\tau = \frac{1}{4}(v_b - v_i)\omega_r r^2 \rho_\infty a_0 bc_r \tag{10}$$

where v_i denotes the induced velocity through the rotor, a_0 represents the rotor lift curve slope, b represents the number of rotor blades, and c_r is the rotor blade chord. The farfield velocity can be expressed as[7]:

$$v' = \sqrt{v_x^2 + v_y^2 + (v_z - v_i)^2} \tag{11}$$

The induced velocity can now be expressed as follows[7]:

$$v_i = \frac{\tau}{2\rho_\infty \pi r^2 v'}$$

(12)

This set of equations was solved iteratively. The force on the vehicle due to the rotor can now be expressed as:

$$\mathbf{F}_{\mathbf{rotor}} = \begin{bmatrix} 0 \\ 0 \\ -\tau \end{bmatrix}$$

(13)

Expressions for power are needed to calculate the aerodynamic moment exerted on the rotor. The induced and profile power[6] are given as:

$$P_{ind} = \tau(v_i - v_z)$$

(14)

$$P_{prof} = \frac{1}{8}\rho_\infty f_r r\omega_r \left[(r\omega_r)^2 + 4.6(v_x^2 + v_y^2)\right]$$

(15)

with f_r representing a rotor friction term, given as[6]:

$$f_r = C_{D_o} rbc_r$$

(16)

The moment exerted by the air on the rotor can be expressed as:

$$M_r = \frac{P_{ind} + P_{prof}}{\omega_r}$$

(17)

A simple expression relating the power of the engine to the rotor angular velocity and the throttle position was chosen as:

$$P_e = \frac{x_{thr}K_{bhp}\eta_e \min(\omega_r K_{dr}, K_{rps,\max})}{K_{rps,\max}}$$

(18)

where x_{thr} represents the state of the throttle, K_{bhp} is the horsepower of the engine, η_e is the engine efficiency, K_{dr} is the main rotor gear ratio, and $K_{rps,\max}$ is the maximum revolutions per second of the engine. The differential equation governing the throttle state is given as:

$$\dot{x}_{thr} = \frac{\delta_{thr} - x_{thr}}{K_{time}}$$

(19)

where δ_{thr} represents the throttle input and K_{time} is the engine time constant. The moment exerted by the engine and the moment applied to the vehicle can now be written as:

$$M_e = \frac{P_e}{\omega_r K_{dr}} \tag{20}$$

$$\mathbf{M_{rotor}} = \begin{bmatrix} 0 \\ 0 \\ -M_e K_{dr} \end{bmatrix} \tag{21}$$

Note that Eq. (21) holds for a rotor that rotates clockwise, when viewed from the top. The sign would be switched if the rotor spun counter-clockwise. Finally, the differential equation governing the rotor angular speed is given as:

$$\dot{\omega}_r = \frac{M_e K_{dr} - M_r}{b i_b} \tag{22}$$

where i_b represents the moment of inertia of a single blade about the spin axis.

C. Duct

A simple lift-drag model was developed to estimate the aerodynamic forces acting on the annular wing that forms the duct. This model uses incompressible, steady flow theory, which is a broad simplification of the highly unsteady flow around the duct. As was the case with the rotor model, this simple aerodynamic model is sufficient to introduce duct aerodynamic forces and moments with appropriate magnitudes.

Take θ to be a measure of angular position around the duct, and $\hat{\mathbf{i}}, \hat{\mathbf{j}}$ to be unit vectors aligned with the vehicle's x and y axes. The following expressions for the radial unit vector and flow velocity vector in the xy-plane of the duct can be written:

$$\widehat{\mathbf{e}}_\mathbf{r} = \widehat{\mathbf{i}} \cos\theta + \hat{\mathbf{j}} \sin\theta \tag{23}$$

$$\mathbf{V_{xy}} = -v_x \widehat{\mathbf{i}} - v_y \widehat{\mathbf{j}} \tag{24}$$

Note that the velocity terms in Eq. (24) represent the velocity of the vehicle relative to the air, so the negative signs provide the air velocities with respect to the body. The radial flow velocity as a function of the angular position around the duct and the z-component of the flow velocity are now given as:

$$V_r(\theta) = \mathbf{V_{xy}} \cdot \hat{\mathbf{e}}_\mathbf{r} = -v_x \cos\theta - v_y \sin\theta \tag{25}$$

$$V_z(\theta) = v_i - v_z \tag{26}$$

Note that this model assumes the z-component to be constant around the duct. This assumption can be relaxed and replaced with a more advanced model if necessary. The dynamic pressure and angle of attack can now be written for every position around the duct:

$$q_d(\theta) = \frac{1}{2}\rho_\infty \left(V_r^2 + V_z^2\right) \tag{27}$$

$$\alpha_d(\theta) = \tan^{-1}\left(\frac{V_r}{V_z}\right) \tag{28}$$

This allows us to find the lift and drag per unit span around the duct:

$$l(\theta) = C_{l,d}(\alpha)q_d c_d \qquad d(\theta) = C_{d,d}(\alpha)q_d c_d \tag{29}$$

where $C_{l,d}(\alpha)$ is the duct airfoil lift curve, $C_{d,d}(\alpha)$ is the duct airfoil drag curve, and c_d is the duct chord. Since the airfoil comprising the duct is symmetric, the pitching moment was neglected. The components of the lift and drag per unit span can be written as follows:

$$l_x(\theta) = l\cos\alpha\cos\theta \qquad d_x(\theta) = d\sin\alpha\cos\theta \tag{30}$$

$$l_y(\theta) = l\cos\alpha\sin\theta \qquad d_y(\theta) = d\sin\alpha\sin\theta \tag{31}$$

$$l_z(\theta) = -l\sin\alpha \qquad d_z(\theta) = d\cos\alpha \tag{32}$$

These quantities can now be integrated around the duct; the ensuing expression for the x–component of lift is given as an example:

$$L_x = \int_{duct} l_x(x,y)dxdy = r\int_0^{2\pi} l_x(\theta)d\theta \tag{33}$$

The following expressions were used to describe the lift and drag curves of the duct airfoil:

$$C_{l,d}(\alpha) = \min\left(C_{l,\max}, \quad \max\left(\frac{1}{2}C_{l_\alpha}\sin(2\alpha), \quad C_{l,\min}\right)\right) \tag{34}$$

$$C_{d,d}(\alpha) = C_{d,offset} - C_{d,gain}\cos(2\alpha) \tag{35}$$

where C_{l_α} represents the duct airfoil lift curve slope, $C_{l,\min}$ and $C_{l,\max}$ represent limits on the lift coefficient, and $C_{d,offset}$ and $C_{d,gain}$ represent empirical constants used to fit the drag curve. Fig. 2 compares the estimated lift and drag curves with data for a NACA 0015 airfoil[8-10] at a Reynolds number of 80000. The GTSpy Reynolds number is much lower than 80000, but the data still gives an approximation for the shape of the curves. There are also some discrepancies between the estimated and experimental lift curves; however, for the sake of simplicity, a more accurate lift curve estimation was not pursued.
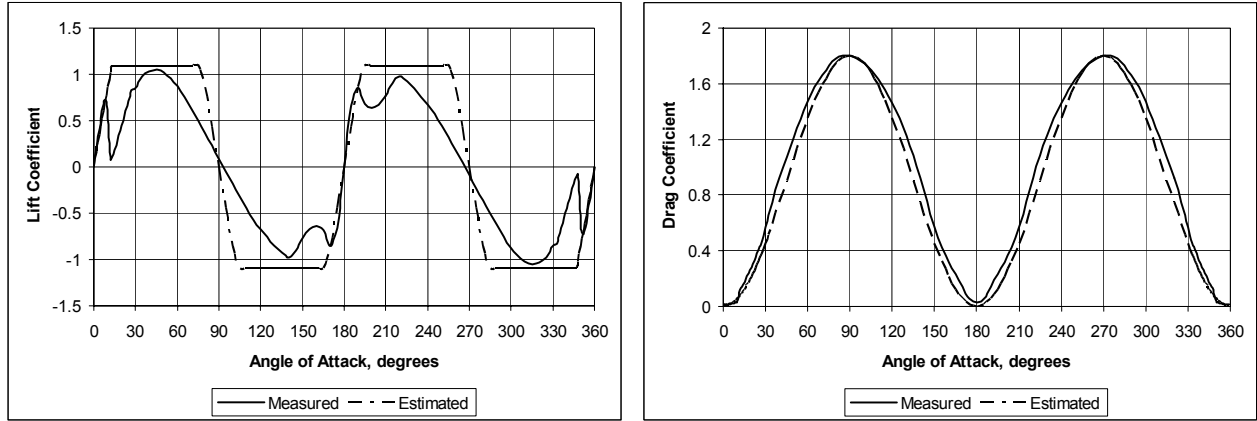
a) Lift Coefficient   **b) Drag Coefficient**
Figure 2: Measured and Estimated NACA 0015 Aerodynamics

In the presence of a crosswind, the vehicle must supply a force to the incoming flow to align it with the duct. This creates a reaction force, known as momentum drag, on the vehicle that tries to move the vehicle downwind. The force is equal to the mass flow through the duct multiplied by the crosswind velocity[2]. This force vector can be expressed as:

$$\mathbf{F_{mdrag}} = -\dot{m} \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} = -v_i \rho_\infty \pi r^2 \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \tag{36}$$

Crosswinds also create a region of higher velocity over the near lip of the duct as the surrounding air is pulled into the duct by the rotor. This creates higher lift on this lip, resulting in a moment that turns the vehicle away from the crosswind[2]. This moment is expressed as follows:

$$\mathbf{M_{lip}} = C_{duct} \rho_\infty r \begin{bmatrix} v_y \left| v_y \right| \\ -v_x \left| v_x \right| \\ 0 \end{bmatrix} \tag{37}$$

11

where C_{duct} represents a proportionality constant. This expression for the duct moment is a very simple approximation in which the moment is assumed to be proportional to the dynamic pressure caused by the crosswind. The proportionality constant was initially estimated from experimental data provided in Ref. 2 and later adjusted based on vehicle flight tests.

The force and moments caused by the duct can now be expressed as:

$$\mathbf{F_{duct}} = \begin{bmatrix} L_x + D_x \\ L_y + D_y \\ L_z + D_z \end{bmatrix} + \mathbf{F_{mdrag}} \tag{38}$$

$$\mathbf{M_{duct}} = \begin{bmatrix} L_x l_d \\ L_y l_d \\ 0 \end{bmatrix} + \mathbf{M_{lip}} \tag{39}$$

where l_d represents the distance between the vehicle center of gravity and the duct aerodynamic center. In this model, the moment due to drag forces was neglected.

The downwash angle induced by the duct can be calculated from the lift force components. The lift is generally related to the downwash angle by the following expression:

$$L = \dot{m}v\gamma \tag{40}$$

where $\dot{m}$ and v represent the mass flow and flow velocity. This equation yields expressions for the downwash angles along the x and y axes:

$$\gamma_x = \frac{L_x}{\rho_\infty \pi r^2 \left[(v_i - v_z)^2 + v_x^2 \right]} \tag{41}$$

$$\gamma_y = \frac{L_y}{\rho_\infty \pi r^2 \left[(v_i - v_z)^2 + v_y^2 \right]} \tag{42}$$

An alternate approach used in modeling duct aerodynamics, especially when more accurate performance evaluation is required, is using computational fluid dynamics to form a database of aerodynamic coefficients that is accessed during simulation to formulate the aerodynamic forces and moments[11,12]. If such a database is available or a higher degree of aerodynamic fidelity is desired, the duct model given above could be replaced.

D. Control Surfaces

Since the control surfaces are moveable flat-plate lifting surfaces, the lift forces they generate will depend on their effective angles of attack. The angle of attack is a function of the control surface deflection, the local airflow velocities, and the downwash induced by the duct. Since the control surfaces are located away from the center of gravity, terms are included to account for induced airflow due to the angular velocity of the vehicle. The following expressions specify how these effects affect the elevator, aileron, and rudder angles of attack:

$$\alpha_e = \delta_e + \tan^{-1}\left(-v_x - \omega_y l_e, v_i - v_z\right) - \gamma_x \tag{43}$$

$$\alpha_a = -\delta_a + \tan^{-1}\left(-v_y + \omega_x l_a, v_i - v_z\right) - \gamma_y \tag{44}$$

$$\alpha_r = \delta_r + \tan^{-1}\left(-\omega_z l_r, v_i - v_z\right) \tag{45}$$

In these expressions, the δ terms represent the control surface deflections, and the l terms represent the distances between the control surfaces' aerodynamic centers and the vehicle center of gravity. The terms elevator, aileron, and rudder are applied to the control surfaces in the helicopter mode. Therefore, the rudder denotes the vanes in the duct outflow, and the elevators and ailerons are the control surfaces located at the end of the fuselage.

Similarly, expressions for the dynamic pressure experienced by each control surface can be found by including the appropriate velocity components for each surface:

$$q_e = \frac{1}{2}\rho_\infty\left[(v_i - v_z)^2 + (v_x + \omega_y l_e)^2\right] \tag{46}$$

$$q_a = \frac{1}{2}\rho_\infty\left[(v_i - v_z)^2 + (v_y - \omega_x l_a)^2\right] \tag{47}$$

$$q_r = \frac{1}{2}\rho_\infty\left[(v_i - v_z)^2 + (\omega_z l_r)^2\right] \tag{48}$$

Using Eq. (34) with the appropriate parameters for the various control surfaces provides the control surface lift coefficients based on the control surface angles of attack. This allows us to find the components of the lift forces that are normal to the vehicle's body axes and the moments created by these forces.

$$\mathbf{F_{cs}} = \begin{bmatrix} \text{sgn}(v_i - v_z)C_{L,e}q_e\cos(\alpha_e - \delta_e)S_e \\ \text{sgn}(v_i - v_z)C_{L,a}q_a\cos(\alpha_a + \delta_a)S_a \\ 0 \end{bmatrix} \tag{49}$$

13

$$\mathbf{M_{cs}} = \begin{bmatrix} -\operatorname{sgn}(v_i - v_z)C_{L,a}q_a\cos(\alpha_a + \delta_a)S_a l_a \\ \operatorname{sgn}(v_i - v_z)C_{L,e}q_e\cos(\alpha_e - \delta_e)S_e l_e \\ \operatorname{sgn}(v_i - v_z)C_{L,r}q_r\cos(\alpha_r - \delta_r)S_r l_r \end{bmatrix} \tag{50}$$

The *S* terms represent the total area of each control surface. Note that for the control surfaces, the effect of the drag forces and moments has been neglected.

E. Gravity

Gravitational forces are accounted for as follows:

$$\mathbf{F_{grav}} = \mathbf{L_{bv}}\begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{51}$$

in which g represents the local gravitational acceleration and $\mathbf{L_{bv}}$ represents the direction cosine matrix that transforms vectors in the local frame to the body frame.

F. Gyroscopic Moment

The rotation of the rotor creates gyroscopic torques. During hover, the rotor speed is nearly constant. It is only in aggressive maneuvering that the rate of change becomes appreciable. It is assumed, for simplicity, that the rate of change of the rotor's angular velocity is negligible. The expression for this moment vector, as well as the equation from which it was derived, is therefore given as:

$$\mathbf{M_{gyro}} = -\boldsymbol{\omega} \times \mathbf{H_r} = bi_b\omega_r\begin{bmatrix} -\omega_y \\ \omega_x \\ 0 \end{bmatrix} \tag{52}$$

IV. Control Architecture

A neural-network adaptive controller developed to operate other UAVs was used to control the GTSpy. 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 Ref. 4.

14

Numerous methods for controlling autonomous UAVs have been developed[13-16]. Feedback linearization, specifically dynamic inversion, is used to control the GTSpy. Use of dynamic inversion by itself would require an accurate system model for all flight regimes[5]. Such a model is difficult to obtain, especially for a ducted fan aircraft, which is a highly coupled system with complex flow fields. Instead, a very simple model of the vehicle is used in the dynamic inversion. A neural network is then trained online to adapt for the modeling errors. In this approach, certain effects, such as actuator dynamics and nonlinearities, can create problems for the adaptive element[3,4]. Pseudo-control hedging (PCH) is used to prevent the neural network from continuously trying to adapt to these effects. This design eliminates the need for accurate models of the system and actuators. Additionally, since PCH allows adaptation in the face of actuator limitations (saturation), it allows the vehicle to make use of its entire flight envelope. This is an essential capability for UAVs in urban environments or combat situations, where aggressive maneuvering is important for obstacle avoidance and vehicle survival.

A. Vehicle Dynamics

An aircraft can be generally described with the following nonlinear equations[4]; Eqs. (1) – (4) represent specific forms of these equations:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{53}$$

$$\dot{\mathbf{v}} = \mathbf{a}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_f, \boldsymbol{\delta}_m) \tag{54}$$

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}(\mathbf{q}, \boldsymbol{\omega}) \tag{55}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \boldsymbol{\delta}_f, \boldsymbol{\delta}_m) \tag{56}$$

The system is governed by two nested loops. The outer loop features the translational dynamics, Eq. (54), and the inner loop features the attitude dynamics, Eq. (56). The vehicle attitude is tracked by a quaternion to avoid singularities present in kinematics based in Euler angles. The state vector of the vehicle is given by:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T & \mathbf{q}^T & \boldsymbol{\omega}^T \end{bmatrix}^T \tag{57}$$

The vector of control signals can be written as:

$$\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_f^T & \boldsymbol{\delta}_m^T \end{bmatrix}^T \tag{58}$$

with $\boldsymbol{\delta}_f$ representing the main force actuators and $\boldsymbol{\delta}_m$ representing the main moment actuators. For the GTSpy, the main force actuator is the rotor thrust, which is controlled by the engine RPM. The main moment actuators are the vanes and control surface pairs. The actuator dynamics may not be known, but they are assumed to be asymptotically stable.

A transformation is introduced to provide approximate feedback linearization for the system. This transformation is represented as[4]:

$$\begin{bmatrix} \mathbf{a}_{des} \\ \boldsymbol{\alpha}_{des} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{q}_{des}, \boldsymbol{\delta}_{f_{des}}, \hat{\boldsymbol{\delta}}_m\right) \\ \hat{\boldsymbol{\alpha}}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{m_{des}}\right) \end{bmatrix} \tag{59}$$

where $\mathbf{a}_{des}$ and $\boldsymbol{\alpha}_{des}$ are known as the pseudo-control signals, which can be though of as desired translational and angular accelerations. The approximation used for $\mathbf{a}(\cdot)$ and $\boldsymbol{\alpha}(\cdot)$ is given by $\hat{\mathbf{a}}(\cdot)$ and $\hat{\boldsymbol{\alpha}}(\cdot)$. The control inputs and attitude that are needed to produce the desired pseudo-control are given by $\boldsymbol{\delta}_{f_{des}}$, $\boldsymbol{\delta}_{m_{des}}$, and $\mathbf{q}_{des}$, respectively. Finally, $\hat{\boldsymbol{\delta}}_f$ and $\hat{\boldsymbol{\delta}}_m$ represent the estimated actuator positions. If the approximation is chosen so that it is invertible, expressions for the desired control inputs and attitude can be obtained:

$$\begin{bmatrix} \boldsymbol{\delta}_{f_{des}} \\ \mathbf{q}_{des} \end{bmatrix} = \hat{\mathbf{a}}^{-1}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \mathbf{a}_{des}, \hat{\boldsymbol{\delta}}_m\right) \tag{60}$$

$$\boldsymbol{\delta}_{m_{des}} = \hat{\boldsymbol{\alpha}}^{-1}\left(\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\alpha}_{des}\right) \tag{61}$$

This approximate inversion provides the closed-loop dynamics given below:

$$\dot{\mathbf{v}} = \mathbf{a}_{des} + \overline{\boldsymbol{\Delta}}_a\left(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}\right) - \mathbf{a}_h \tag{62}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}_{des} + \overline{\boldsymbol{\Delta}}_\alpha\left(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}\right) - \boldsymbol{\alpha}_h \tag{63}$$

where the $\overline{\boldsymbol{\Delta}}_{a,\alpha}\left(\mathbf{x}, \boldsymbol{\delta}, \hat{\boldsymbol{\delta}}\right)$ terms represent the nonlinear model error that results from inaccuracies in the inverted vehicle model and actuator models. Certain actuator behavior, such as saturation, will introduce pseudo-control that cannot be reached; this effect introduces the signals $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$. Stabilization of the system can be achieved by choosing the pseudo-controls in the following way[4]:

$$\mathbf{a}_{des} = \mathbf{a}_{cr} + \mathbf{a}_{pd} - \overline{\mathbf{a}}_{ad} \tag{64}$$

$$\boldsymbol{\alpha}_{des} = \boldsymbol{\alpha}_{cr} + \boldsymbol{\alpha}_{pd} - \overline{\boldsymbol{\alpha}}_{ad} \qquad (65)$$

in which $\mathbf{a}_{cr}$ and $\boldsymbol{\alpha}_{cr}$ are the output signals from the vehicle dynamics reference models, $\mathbf{a}_{pd}$ and $\boldsymbol{\alpha}_{pd}$ are the output signals of proportional-derivative compensators, and $\overline{\mathbf{a}}_{ad}$ and $\overline{\boldsymbol{\alpha}}_{ad}$ are the output signals from the adaptive element.

B. Reference Model & PCH

In designing the reference model, care must be taken to ensure that the effects which introduce the terms $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ in Eqs. (62) and (63) are not present in the tracking error dynamics. If that happens, the reference model will continue to issue commands as if there were no attitude limits or actuator saturation. The adaptive element would then try to correct for the discrepancy. This can be avoided through PCH as follows[4]:

$$\dot{\mathbf{v}}_r = \mathbf{a}_{cr}\left(\mathbf{p}_r, \mathbf{v}_r, \mathbf{p}_c, \mathbf{v}_c\right) - \mathbf{a}_h \qquad (66)$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{cr}\left(\mathbf{q}_r, \boldsymbol{\omega}_r, \mathbf{q}_c \oplus \mathbf{q}_{des}, \boldsymbol{\omega}_c\right) - \boldsymbol{\alpha}_h \qquad (67)$$

where the subscript c denotes commands and the subscript r denotes the reference model. The signals $\mathbf{a}_h$ and $\boldsymbol{\alpha}_h$ represent the difference between the commanded and realized pseudo-control. They are given as follows:

$$\mathbf{a}_h = \hat{\mathbf{a}}\left(\mathbf{x}, \mathbf{q}_{des}, \delta_{f_{des}}, \hat{\boldsymbol{\delta}}_m\right) - \hat{\mathbf{a}}\left(\mathbf{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m\right) = \mathbf{a}_{des} - \hat{\mathbf{a}}\left(\mathbf{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m\right) \qquad (68)$$

$$\boldsymbol{\alpha}_h = \hat{\boldsymbol{\alpha}}\left(\mathbf{x}, \hat{\boldsymbol{\delta}}_f, \boldsymbol{\delta}_{m_{des}}\right) - \hat{\boldsymbol{\alpha}}\left(\mathbf{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m\right) = \boldsymbol{\alpha}_{des} - \hat{\boldsymbol{\alpha}}\left(\mathbf{x}, \hat{\boldsymbol{\delta}}_f, \hat{\boldsymbol{\delta}}_m\right) \qquad (69)$$

For the specific class of aircraft that this controller was developed, the following general reference model, as given in Ref. 4, was chosen:

$$\mathbf{a}_{cr} = \mathbf{R}_p\left(\mathbf{p}_c - \mathbf{p}_r\right) + \mathbf{R}_d\left(\mathbf{v}_c - \mathbf{v}_r\right) \qquad (70)$$

$$\dot{\mathbf{v}}_r = \mathbf{a}_{cr} - \mathbf{a}_h \qquad (71)$$

$$\boldsymbol{\alpha}_{cr} = \mathbf{K}_p\left[\widetilde{\mathbf{Q}}\left(\mathbf{q}_c \oplus \mathbf{q}_{des}, \mathbf{q}_r\right)\right] + \mathbf{K}_d\left(\boldsymbol{\omega}_c - \boldsymbol{\omega}_r\right) \qquad (72)$$

$$\dot{\boldsymbol{\omega}}_r = \boldsymbol{\alpha}_{cr} - \boldsymbol{\alpha}_h \qquad (73)$$

The attitude error angle function, $\widetilde{\mathbf{Q}}(\,\cdot\,)$, takes two quaternions and forms a vector containing three error angles. It is explicitly provided in Ref. 4. Note that the four gain matrices in the above equation are identical to those used in

the proportional-derivative compensator whose signals are included in Eqs. (64) and (65). The reference model dynamics can be modified if the reference model is to be rate-limited; the details are provided in Ref. 4. These gains must be chosen so that the closed-loop poles are well-placed. Explicit formulas for the gain values can be found through an analysis detailed in Ref. 4 and are given below:

$$R_p = \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \tag{74}$$

$$R_d = 2\frac{\omega_o \omega_i (\zeta_o \omega_i + \omega_o \zeta_i)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \tag{75}$$

$$K_p = \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2 \tag{76}$$

$$K_d = 2\zeta_i \omega_i + 2\zeta_o \omega_o \tag{77}$$

where the gains are functions of the outer-loop and inner-loop natural frequencies and damping ratios of each pole in turn. These scalar gains are subsequently placed in the diagonal gain matrices shown in Eqs. (70) and (72).

C. Adaptive Element

The adaptive element consists of a single-hidden layer perceptron neural network (NN) used to approximate the model error. The input-output relationship of the NN is given as[4]:

$$v_{ad_k} = b_w \theta_{w_k} + \sum_{j=1}^{n_2} w_{jk} \sigma_j (z_j) \tag{78}$$

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

$$\sigma_j (z_j) = 1/(1 + e^{-az_j}) \tag{79}$$

with the neuron activation potential given as a. The input to the jth neuron is given as follows:

$$z_j = b_v \theta_{v_j} + \sum_{i=1}^{n_1} v_{ij} x_{in_i} \tag{80}$$

where b_v represents the inner-layer bias, θ_{v_j} represents the jth threshold, and x_{in_i} represents the NN inputs. The number of inputs, neurons, and outputs is given by n_1, n_2, and n_3, respectively.

18

The NN output can now be written as[4]:

$$\overline{\mathbf{v}}_{\mathbf{ad}} = \mathbf{W}^T \boldsymbol{\sigma}\!\left(\mathbf{V}^T \overline{\mathbf{x}}\right) + \mathbf{v}_{\mathbf{r}} = \begin{bmatrix} \mathbf{a}_{\mathbf{ad}} + \mathbf{a}_{\mathbf{r}} \\ \boldsymbol{\alpha}_{\mathbf{ad}} + \boldsymbol{\alpha}_{\mathbf{r}} \end{bmatrix} \tag{81}$$

in which $\mathbf{W}$ and $\mathbf{V}$ are weight matrices constructed from the weight elements and thresholds described above. The input element $\overline{\mathbf{x}}$ is constructed by concatenating the inner-layer bias with the NN input vector. A robustifying signal, $\mathbf{v}_{\mathbf{r}}$, is introduced to ensure the boundedness of the NN response. The learning laws for the NN can be expressed as time derivatives of the weight matrices, which are given as follows:

$$\dot{\mathbf{W}} = -\left[\left(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}^T \overline{\mathbf{x}}\right)\mathbf{r}^T + \kappa \|\mathbf{e}\| \mathbf{W}\right]\boldsymbol{\Gamma}_{\mathbf{w}} \tag{82}$$

$$\dot{\mathbf{V}} = -\boldsymbol{\Gamma}_{\mathbf{v}}\left[\overline{\mathbf{x}}\left(\mathbf{r}^T \mathbf{W}^T \boldsymbol{\sigma}'\right) + \kappa \|\mathbf{e}\| \mathbf{V}\right] \tag{83}$$

in which the positive definite matrices $\boldsymbol{\Gamma}_{\mathbf{w}}$ and $\boldsymbol{\Gamma}_{\mathbf{v}}$ and the positive scalar κ ensure that the reference model tracking error $\mathbf{e}$ and the NN weights are ultimately uniformly bounded. The outer-layer bias is concatenated with the sigmoidal function outputs to form the $\boldsymbol{\sigma}$ vector, and the outputs of the derivatives of the sigmoidal function are used to create the $\boldsymbol{\sigma}'$ matrix. Finally, the $\mathbf{r}$ signal is a function of the reference model tracking error, the appropriate $\mathbf{P}$ matrix from the Lyapunov equation that addresses the stability of the tracking error dynamics, and the $\mathbf{B}$ matrix from the tracking error dynamics. A proof of the boundedness of the NN response is given in Ref. 4.

D. Approximate Model

As mentioned above, an invertible, approximate model of the vehicle is needed. This model can be exceedingly simple, and the neural network will still be able to compensate for the inaccuracies. Here, the attitude dynamics are obtained through linearization about hover, with all coupling between the attitude and translational dynamics ignored. This is an appropriate choice for low speed flight, which constituted the entirety of the GTSpy flight testing. This model yields the following expression[4]:

$$\boldsymbol{\alpha}_{\mathbf{des}} = \hat{\mathbf{A}}_1 \boldsymbol{\omega} + \hat{\mathbf{A}}_2 \mathbf{v} + \hat{\mathbf{B}}\!\left(\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{des}}} - \boldsymbol{\delta}_{\mathbf{m}_{\mathbf{trim}}}\right) \tag{84}$$

in which the $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$ matrices describe the attitude and translational dynamics and $\boldsymbol{\delta}_{\mathbf{m}_{\mathbf{trim}}}$ represents the trim control vector. This equation can now be rearranged to solve for the moment control vector if the $\hat{\mathbf{B}}$ matrix is

chosen to be invertible. In order to obtain the translational dynamics, the vehicle was modeled as a point mass with

a thrust vector that can be pointed in a certain direction. The dynamics of this model can be expressed as follows:

$$\mathbf{a_{des}} = \begin{bmatrix} 0 \\ 0 \\ Z_{\delta_{thr}} \end{bmatrix} \left(\delta_{thr_{des}} - \delta_{thr_{trim}} \right) + \mathbf{L_{bv}g} \tag{85}$$

where $Z_{\delta_{thr}}$ is the control derivative for vertical axis acceleration and $\mathbf{g}$ represents the gravity vector. The specific

force along the body z axis can be expressed as follows:

$$f_{sf} = \left(\mathbf{a_{des}} - \mathbf{L_{bv}g} \right)_3 \tag{86}$$

This expression can now be rearranged, with Eq. (85), to provide an expression for the required throttle input. The

attitude changes necessary to point the thrust vector in the proper direction to attain necessary translational

accelerations can be expressed as follows[4]:

$$\Delta\Phi_1 = a_{des_2} / f_{sf} \tag{87}$$

$$\Delta\Phi_2 = a_{des_1} / f_{sf} \tag{88}$$

$$\Delta\Phi_3 = 0 \tag{89}$$

These quantities represent small corrections to reference body attitude and the attitude commands. Note that the

third quantity is zero because heading plays no part in the translational acceleration in this model. Implicit in this

method of calculating the attitude correction is the assumption that the thrust vector is pointed through small angles.

These three angle corrections can be converted to a quaternion correction:

$$\mathbf{q_{des}} = \mathbf{q}\left(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3 \right) \tag{90}$$

where $\mathbf{q}(\cdot)$ represents the transformation from an Euler angle rotation to a quaternion.

Note that this approximate model is not developed from the analysis used to create the simulation model. The

model described above is easily inverted; inverting the simulation model would be very difficult. While the above

model is less accurate than the simulation model, the neural network can sufficiently adapt for its deficiencies in the

low speed regime. Therefore, the more complex simulation model is not used in the dynamic inversion.

20

V. Results

A. Simulation

The environment used to simulate the GTSpy was implemented using C-code. The environment features wind and gust models, sensor models, and hardware simulation, such as serial data output and quantization errors[17]. The simulator also has a desired trajectory editor and 3D graphical rendering of the vehicle and environment. A screenshot from the simulation environment is provided in Fig. 3. There are two images of the GTSpy in Fig. 3; the light image represents the actual vehicle, while the dark image represents the onboard navigation system estimate of vehicle position and attitude.
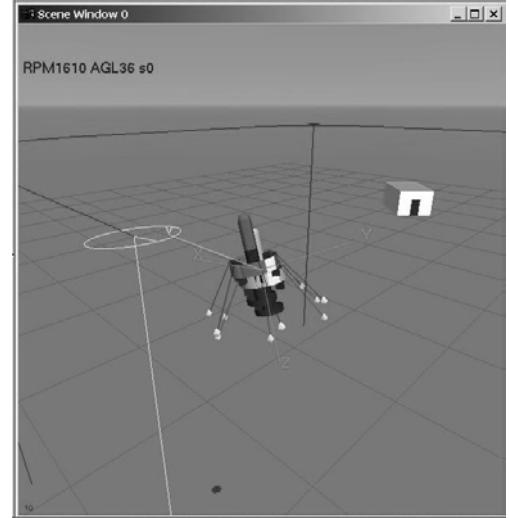
Figure 3: GTSpy Simulation Environment

The simulated position and velocity response to a 50 foot change in position, with a velocity limit of 10 feet per second and an acceleration limit of 5 feet per second squared, is shown in Fig. 4. The controller tracks the generated position trajectory very well, with little overshoot. The tracking of the velocity commands is a little coarser, as expected, but still acceptable. The position error on all three axes during the maneuver is shown in Fig. 5. As the vehicle moves north, position errors also occur in the east-west direction. However, these errors, which are roughly the same magnitude as the errors in the direction of travel, are within acceptable bounds.
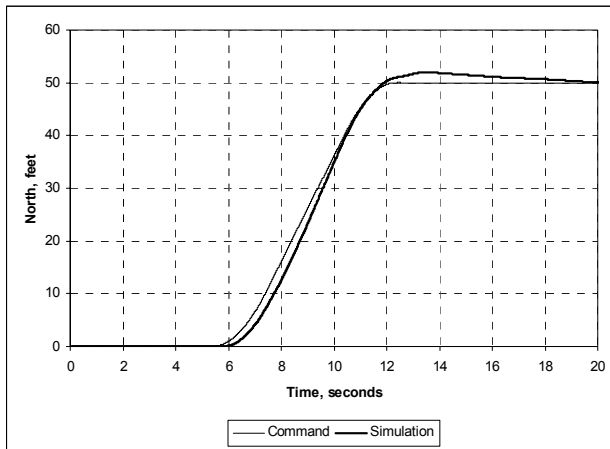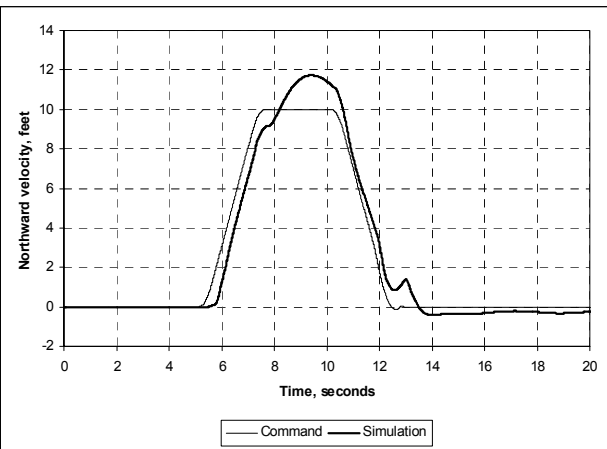
a) Position   b) Velocity
Figure 4: Results from Position Maneuver
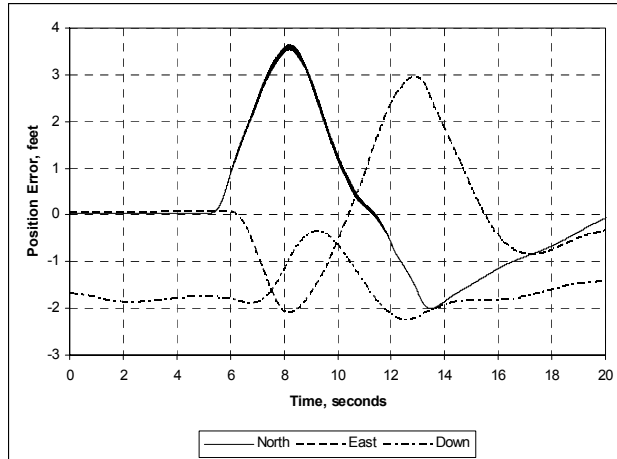
Figure 5: Position Maneuver Error
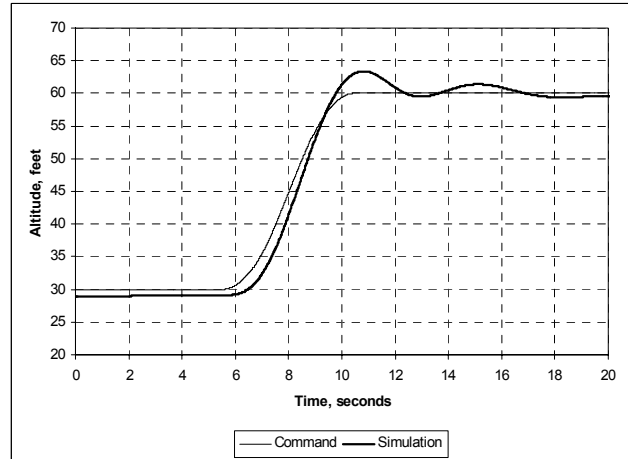
Figure 6: Results from Altitude Maneuver

The simulated response to a commanded thirty foot increase in altitude, using the same velocity and acceleration limits, is shown in Fig. 6. Again, the controller tracks the generated trajectory well. There is greater overshoot observed during this maneuver. This is a result of the inherent time delay present in using the rotor RPM as an actuator, since changes in rotor speed take a finite time to occur.

The simulated results from a box maneuver, in which the commanded trajectory is a square with sides measuring 50 feet, is provided in Fig. 7. During this maneuver, both altitude and heading are maintained. A velocity limit of 8 feet per second and an acceleration limit of 4 feet per second squared are also observed. As expected from the previously discussed simulations, the controller performs well. The trajectory is tracked closely, with little overshoot. Based on these types of simulations of low speed maneuvering, it was determined that the controller could be expected to perform well during flight testing in the near-hover regime.

Once it was determined that the controller worked well in the near-hover regime, simulations were performed to examine the performance in more aggressive maneuvers. Figure 8 shows the simulated results from a position change of 100 feet, with a velocity limit of 20 feet per second and an acceleration limit of 5 feet per second squared. Figure 9 shows the simulated results of a 160 foot position change, with a velocity limit

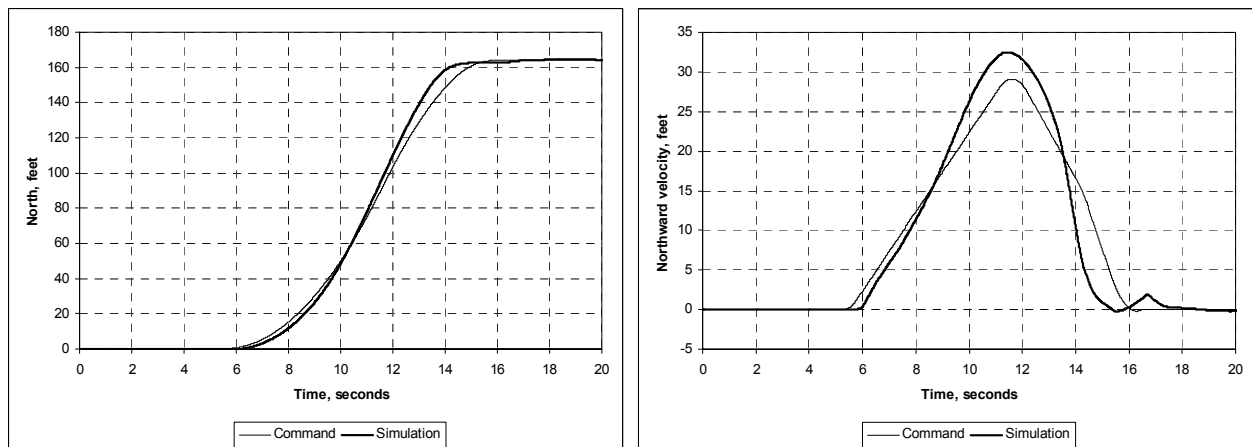**Figure 7: Results from Box Maneuver**

22

of 30 feet per second and an acceleration limit of 6 feet per second squared. In both of these cases, the controller still tracks the commanded trajectory rather well. At these speeds and the pitch angles required to achieve them, the duct forces, which are not modeled in the inverse model, begin growing in magnitude. The neural network is still capable of tracking the desired trajectory in spite of substantial unmodeled effects and the fact that the small angle assumptions implicit in the inverted approximate model no longer hold. However, these effects cause the tracking error to increase with the aggressiveness of the maneuver. The position errors from these two cases are shown in Figs. 10 and 11, respectively. While the errors are large, they are reasonable, given the nature of the maneuvers. These results suggest that the controller can be expected to perform sufficiently well in aggressive maneuvers that push the limits of the near-hover regime. However, it can also be expected that very aggressive maneuvers, such as transition to forward flight, will generate unacceptably large tracking errors.
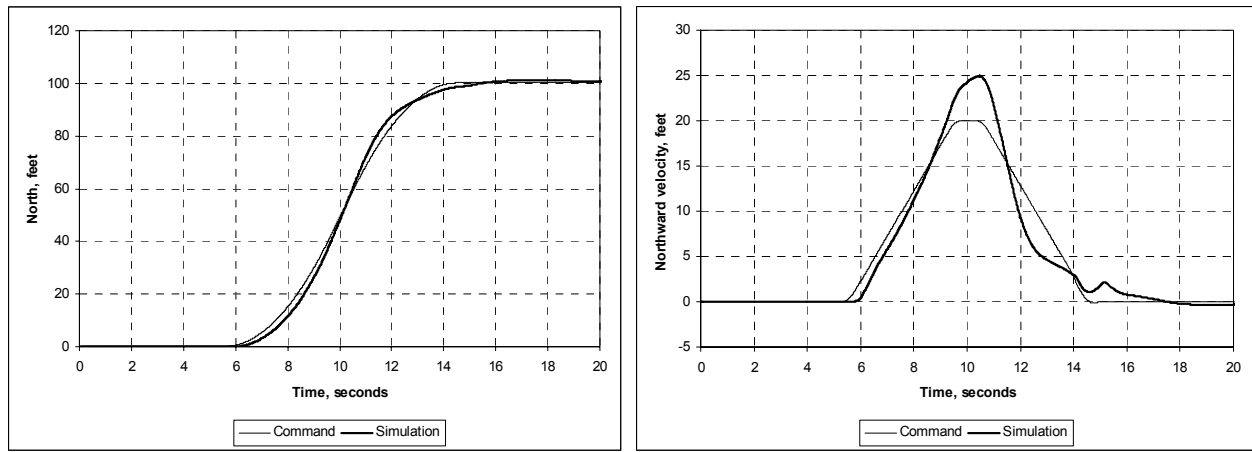
a) Position   b) Velocity
Figure 8: Results from Position Maneuver with 20 ft/s Limit

a) Position   b) Velocity
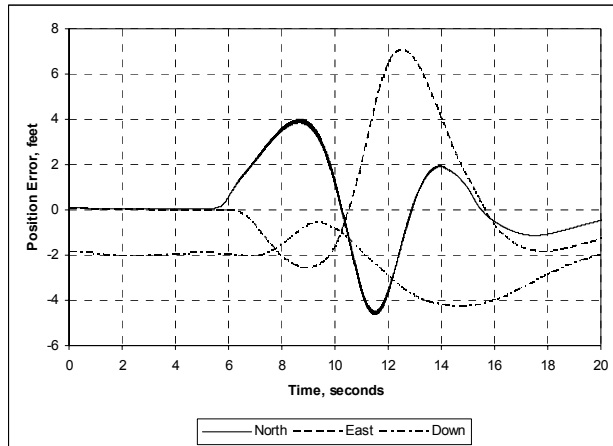**Figure 9: Results from Position Maneuver with 30 ft/s Limit**

23

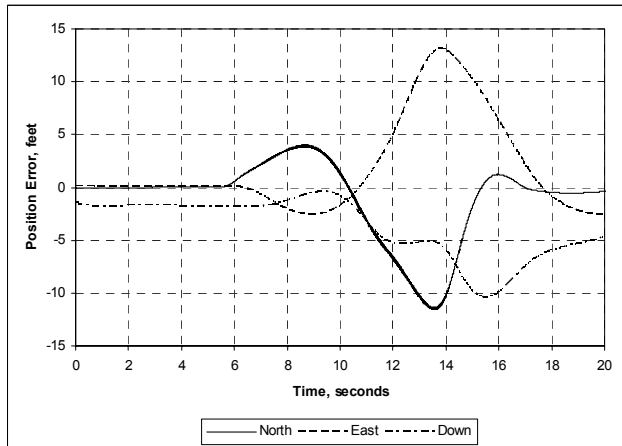**Figure 10: Position Maneuver Error, 20 ft/s Limit**

Figure 11: Position Maneuver Error, 30 ft/s Limit

B. Flight Testing

During flight testing, vehicle positions and velocities were measured using the onboard GPS. Angular rates and linear accelerations were also measured via rate gyros and accelerometers. The flight testing of the GTSpy began with hovering and simple maneuvers around hover. Successive loops were closed, incrementally giving the flight control system autonomous control over attitude, horizontal translation, and finally altitude. These initial tests were performed with the GTSpy tethered to a gantry for safety. After several tethered flights were successfully completed, the GTSpy was flown without the tether. The untethered flights began with simple hovers, and proceeded to takeoffs and landings, followed by more involved maneuvers. The culmination of

Figure 12: Deployment of GTSpy from Helicopter; First Air Deployment of a Hovering Aircraft

the flight testing involved a successful airborne deployment of the GTSpy from the GTMax, shown in Fig. 12. The GTMax is a modified Yamaha R-Max helicopter that is also flown as an autonomous UAV and is the subject of the research detailed in Ref. 17. During the deployment, the GTSpy was dropped from the hovering GTMax, and it autonomously maneuvered itself into a hover. This may be the first time a hovering aircraft has been air launched. Results from a flight test where the GTSpy autonomously executes a box maneuver similar to the one described above are shown in Figs. 13 - 15. The same velocity and acceleration limits observed in the simulated case were

observed in the flight test. As expected from the simulation results, the controller performs well during the maneuver. However, the controller does not track the trajectory as closely as in the simulated case. This is the result of two effects that were not included in the above simulation runs: quantization error on the sensor measurements and atmospheric disturbances. The flight test was performed with light, intermittent wind, and its presence contributes to the decreased performance. Even though the tracking performance is slightly less than expected, it is still good. There is one large discrepancy, occurring on the third leg of the maneuver. This is probably due to either a larger wind gust or a large, erroneous GPS position measurement. Figure 14 shows the heading tracking performance. In the simulation box maneuver, the heading was kept very close to zero; in the
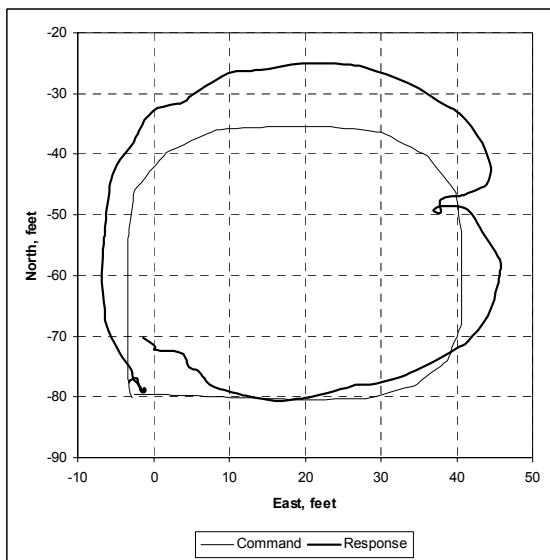

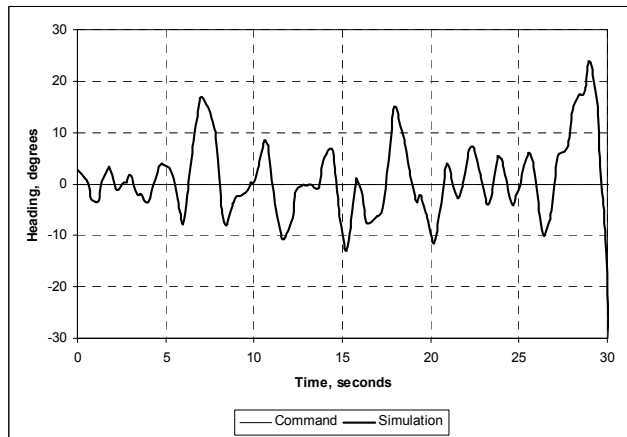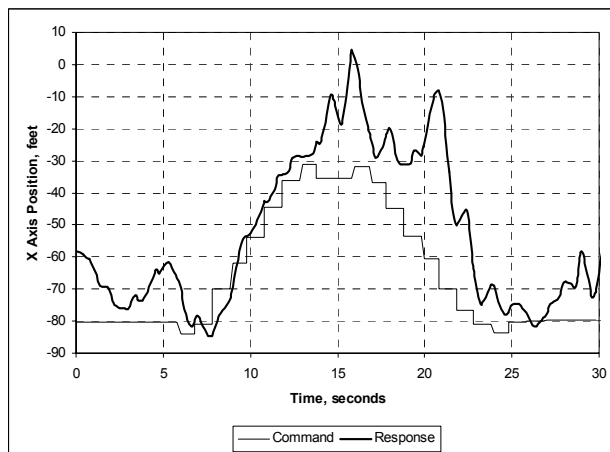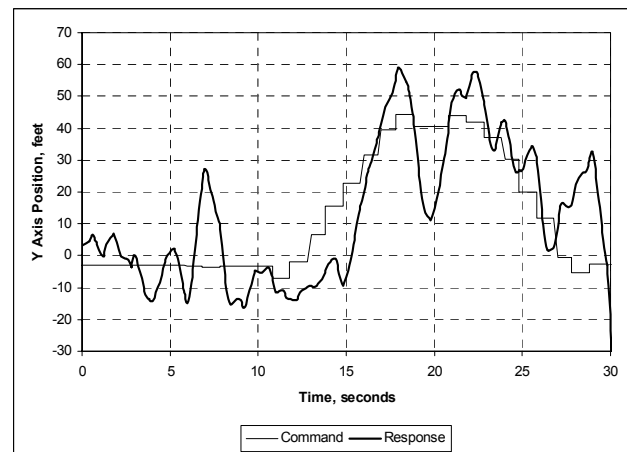**Figure 13: Position Results from Box Maneuver**

Figure 14: Heading Results from Box Maneuver

a) X Axis

b) Y Axis

Figure 15: Body Axis Position Results from Box Maneuver

flight test, the heading varies more than in the simulation, but remains within reasonable bounds. Note that Fig. 15 shows the position tracking with respect to the body frame axes. Therefore, some of the errors observed are due to differences between the commanded and actual attitude. Additionally, some of the large errors noted at times close to 20 seconds are due to the discrepancy observed in the third leg.

C. Limitations

The results provided in the Simulation subsection suggest that the controller would continue to perform well in low speed aggressive maneuvering. However, as predicted at the end of the Simulation subsection, the controller has difficulty tracking the commanded trajectory if transition to high speed flight (in airplane mode) is attempted. This results from the errors in the inverted approximate model becoming large enough to overcome the neural network's ability to adapt and compensate. The initial approximate model was linearized about hover and based on the concept of thrust pointing, both of which are not appropriate for horizontal flight. However, it is believed that the use of alternative models in the dynamic inversion which capture both hover and horizontal flight characteristics will allow the control architecture developed for low speed flight to satisfactorily control the GTSpy in all flight regimes. Minor changes to the approximate model may also provide better performance in the low speed regime. For example, the attitude corrections performed in Eqs. (87) – (89) could be modified to avoid making small angle assumptions. This would allow more aggressive maneuvers by reducing modeling error present during large attitude deviations from hover.

VI. Conclusions

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. This approach, which has been used successfully on other vehicles, was found to be well-suited for use on the small ducted fan GTSpy. The inclusion of pseudo-control hedging allowed for appropriate adaptation and fine vehicle control in aggressive maneuvers where actuator saturation would otherwise become problematic. A more involved vehicle model was formulated for use in simulation. This model was synthesized from an analysis of the external forces and moments affecting the GTSpy. Additionally, it was recognized that the vehicle model could be rapidly and easily modified through a re-analysis of the contributing forces and moments, as well as the individual models used to predict these contributions. During

flight testing, the controller was found to perform well in the near-hover regime, with the flight performance successfully predicted by simulation. Simulation results suggesting that the controller would continue to perform well in more aggressive low speed maneuvers than those performed during flight testing were also provided.

Appendix: Numerical Values for the GTSpy

Mass & Inertia

Mass	0.155	slugs
Inertia		
Ixx	0.025	slug-ft^2
Iyy	0.025	slug-ft^2
Izz	0.006	slug-ft^2
Ixz	0.000	slug-ft^2

Fuselage Aerodynamics

Drag Coefficient		
X	0.5	
Y	0.5	
Z	0.1	
Reference Area	0.5	ft^2
Moment Arm	-0.4	ft

Rotor

Rotor Radius	0.454	ft
Rotor Twist	0.2618	rad
Rotor Lift Curve Slope	5.9	/rad
Number of Blades	2	
Rotor Chord	0.083	ft
Rotor Drag Coefficient	0.01	
Engine Horsepower	550	ft-lb/s
Engine Efficiency	0.9	
Gear Ratio	1	
Max Radians per Second	1360	rad/s
Engine Time Constant	0.1	s
Blade Moment of Inertia	0.0001	slug-ft^2

Duct

Chord	0.4167	ft
Duct		
Lift Curve Slope	4.712	/rad
Min Lift Coefficient	-1.1	
Max Lift Coefficient	1.1	
Drag Coefficient Gain	0.9	
Drag Coefficient Offset	0.9	
Duct Moment Coefficient	0.8	

Control Surfaces

Control Surface		
Lift Curve Slope	5.341	/rad
Min Lift Coefficient	-1.4	
Max Lift Coefficient	1.4	
Moment Arm		
Elevator	1.156	ft
Aileron	1.156	ft
Rudder	0.371	ft
Planform Area		
Elevator	0.208	ft^2
Aileron	0.208	ft^2
Rudder	0.250	ft^2

References

[1]Schaefer, C. G., Jr. and Baskett, L. J., "GoldenEye: The Clandestine UAV," AIAA Unmanned Unlimited Systems, Technologies, and Operations, AIAA 2003-6634, 2003.

[2]Fleming, J., Jones, T., Ng, W., Gelhausen, P., and Enns, D., "Improving Control System Effectiveness for Ducted Fan VTOL UAVs Operating in Crosswinds," AIAA Unmanned Unlimited Systems, Technologies, and Operations, AIAA 2003-6514, 2003, pp. 1-3.

[3]Johnson, E. N., "Limited Authority Adaptive Flight Control," Ph.D. Thesis, Georgia Institute of Technology, 2000, pp. 15-16, 42.

[4]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.

[5]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.

[6]Heffley, R. K. and Mnich M. A., "Minimum-Complexity Helicopter Simulation Math Model," NASA Contractor Report 177476, Apr. 1988, pp. 19, 24.

[7]W. Z. Stepniewski and C. N. Keys, *Rotary-Wing Aerodynamics*, Dover Publications, New York, 1984, pp. 63.

[8]Lazauskas, L., "Airfoil Data: NACA0015 CL," URL: http://www.cyberiad.net/foildata.htm.

[9]Lazauskas, L., "Airfoil Data: NACA0015 CD," URL: http://www.cyberiad.net/foildata.htm.

[10]Sheldahl, R. E. and Klimas, P.C., "Aerodynamic Characteristics of Seven Airfoil Sections Through 180 Degrees Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines," Sandia National Laboratory, SAND80-2114, Mar. 1981.

[11]Avanzini, G., D'Angelo, S., and de Matteis, G., "Performance and Stability of a Ducted-Fan Uninhabited Aerial Vehicle," AIAA Aerospace Sciences Conf., AIAA 2001-0844, Jan. 2001, pp. 5.

[12]Chang, I. and Rajagopalan, R. G., "CFD Analysis for Ducted Fans with Validation," AIAA Applied Aerodynamics Conf., AIAA 2003-4079, Jun. 2003.

[13]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.

[14]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.

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

[16]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.

[17]Johnson, E. N. and Schrage, D. P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," AIAA Journal of Aerospace Computing, Information, and Communication, Vol. 1, No. 1, Jan. 2004, pp. 8-10.