Simulation and Control Design of a Gliding Autogyro for Precision Airdrop

by

Joshua F. Torgerson

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author .... [illegible]
Department of Aeronautics and Astronautics
May 6, 2005

Certified by .... [illegible]
John J. Deyst
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .... [illegible]
Sean George
Member Technical Staff, Draper Laboratory
Thesis Supervisor

Accepted by ....
Jaime Peraire
Chairman, Department Committee on Graduate Students

[This page intentionally left blank.]

Simulation and Control Design of a Gliding Autogyro for Precision Airdrop

by

Joshua F. Torgerson

Submitted to the Department of Aeronautics and Astronautics
on May 6, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Precision airdrop is a technology whose required capabilities have become more exacting as combat situations necessitate greater degrees of accuracy. Ballistic and parafoil type delivery vehicles do not have the capacity to consistently deliver a payload on, for example, a particular rooftop in an urban combat situation.

A gliding autogyro delivery platform has been investigated as a means of achieving greater airdrop performance. The autogyro has similar gliding characteristics to the parafoil, but has improved wind resilience and control authority. An initial simulation, based on momentum and blade element helicopter theory, has been constructed. A classical controller using a multiple loop closure strategy has been developed that uses a new nonlinear guidance law to follow paths generated by an algorithm considering initial conditions. An extended Kalman filter is used for state estimation. Results from simulations show consistent accuracy of about 5 feet, with the final position error rarely exceeding 10 feet.

Thesis Supervisor: John J. Deyst
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Sean George
Title: Member Technical Staff, Draper Laboratory

[This page intentionally left blank.]

Acknowledgments

Acknowledgment

May 6, 2005

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research in the Autonomous Vehicles Laboratory, Project Number 5042.

Publication of this thesis does not constitute approval by Draper of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Joshua F. Torgerson

[This page intentionally left blank.]

Contents

[This page intentionally left blank.]

List of Figures

List of Tables

[This page intentionally left blank.]

List of Symbols

$\mathbf{C}$	direction cosine transformation matrix, body to local frames
C_{D_0}	zero lift drag coefficient of blades
C_T	coefficient of thrust $\frac{T}{\rho\pi r^2 V_{tip}^2}$
$\mathbf{F}$	force vector $[X\ Y\ Z]^T$
F_r	equivalent flat plate area of blades
G	controller transfer function
$\mathbf{H}$	angular momentum vector
$\mathbf{I}$	inertia matrix
I_b	inertia of rotor blade
I_{xx}, I_{yy}, I_{zz}	inertia
K_{GE}	ground effect constant
L, M, N	torque around x, y, z axes
L_1	reference point distance in nonlinear lateral guidance law
P, Q, R	body rotation rates about x, y, z axes
P_i, P_p	induced and profile power of rotor
$\mathbf{T}$	torque vector $[L\ M\ N]^T$
T	thrust from rotor
U, V, W	linear velocities in x, y, z directions
V_i	induced velocity
V_h	induced velocity in hover
V_{tip}	blade tip speed Ωr
W_b	aggregate velocity normal to blade
W_r	aggregate velocity normal to rotor disk
X, Y, Z	forces in x, y, z directions
X_{uu}, Y_{vv}, Z_{ww}	dimensional drag/lift coefficients
Y_{uv}, Z_{uw}	dimensional lift coefficient due to angle of attack

a	lift curve slope of rotor blade
a_0, a_1, b_1	coning, forward, and side flapping angles of the rotor disk
b	number of blades
c	chord of blades
d_{dw}	horizontal tail location with respect to downwash stream
g	acceleration due to gravity
h_r	height of rotor above the ground
k	controller gain
m	vehicle mass
${}_pN, {}_pE, {}_pD$	north, east, and down positions in the local frame
q_0, q_1, q_2, q_3	quaternions
r	rotor radius
$\mathbf{u}$	input vector
$\mathbf{v}$	velocity vector $[U\ V\ W]^T$
$\mathbf{x}$	estimation state vector
$\tilde{\mathbf{x}}$	simulation state vector
x, y, z	center of pressure location relative to center of gravity
Ω	rotor rotation rate
$\mathbf{\Omega}_3$, $\mathbf{\Omega}_4$	matrices of rotation rates (Equations (2.3) and (2.12)
α_{tip}	blade tip angle of attack
γ	glide slope ratio $\frac{\text{forward speed}}{\text{descent speed}}$
γ_L	Lock number
δ_{col}	rotor collective
δ_f	front/back rotor tilt
δ_s	side rotor tilt
η	angle error for nonlinear path guidance
η_{GE}	ground effect factor
λ	sensor bias
μ	advance ratio
ν	noise

ρ	air density
σ	standard deviation
σ_r	rotor solidity
τ	rotor torque
ϕ, θ, ψ	Euler angles (roll, pitch, yaw)
$\boldsymbol{\omega}$	body rotation rate vector $[P\ Q\ R]^T$
$(\cdot)_A$	aerodynamic
$(\cdot)^L$	local (inertial) frame
$(\cdot)_a$	air relative
$(\cdot)_c$	command
$(\cdot)_{fus}$	fuselage
$(\cdot)_{ht}$	horizontal tail
$(\cdot)^p$	path frame
$(\cdot)_r$	rotor
$(\cdot)_t$	trim
$(\cdot)_{vt}$	vertical tail or fin

[This page intentionally left blank.]

Chapter 1

Introduction

There is a significant need for precision airdrop capability, especially in the case of military operations. Precision airdrop is currently used for missions such as humanitarian aid, combat resupply, and even reentry with the X-38 Crew Return Vehicle parafoil system. The overall strategy is to use a parent vehicle to transport a payload contained in a drop vehicle and deploy it near the desired target. The drop vehicle safely guides the payload to the target and lands. The advent of close quarter urban combat scenarios has increased required capabilities to include higher precision deliveries of small and large payloads. Applicable missions include urban combat resupply, surveillance/reconnasaince, and sensor network delivery. The constrained urban terrain requires enough precision to deploy on top of particular buildings, with the possibility of collision avoidance of rooftop obstacles. Thus, target landing areas on the order of feet are required.

Previous research has investigated two delivery strategies: ballistic and parafoil. The ballistic type was a free-fall, fin stabilized projectile, with a round parachute deployed just above ground level to arrest descent speed [1]. A problem with this strategy was that the limited glide slope of the projectile required an overly precise deployment from the parent vehicle. Additionally, the large shock of the parachute opening at the end of the mission was potentially harmful to payloads. The second parafoil type immediately deployed its parafoil upon release from the parent vehicle [18]. This strategy was deterred by limited controllability and a strong susceptibility to wind disturbances.

A third strategy was proposed and is the subject of this thesis. The drop vehicle in this case is a gliding autogyro. An autogyro is very similar to an airplane with a tractor type propeller, but replaces the wing with an unpowered rotor. The aircraft used as an example is the GyroBee remote controlled model aircraft produced by the Autogyro Company of Arizona (Figure 1-1).

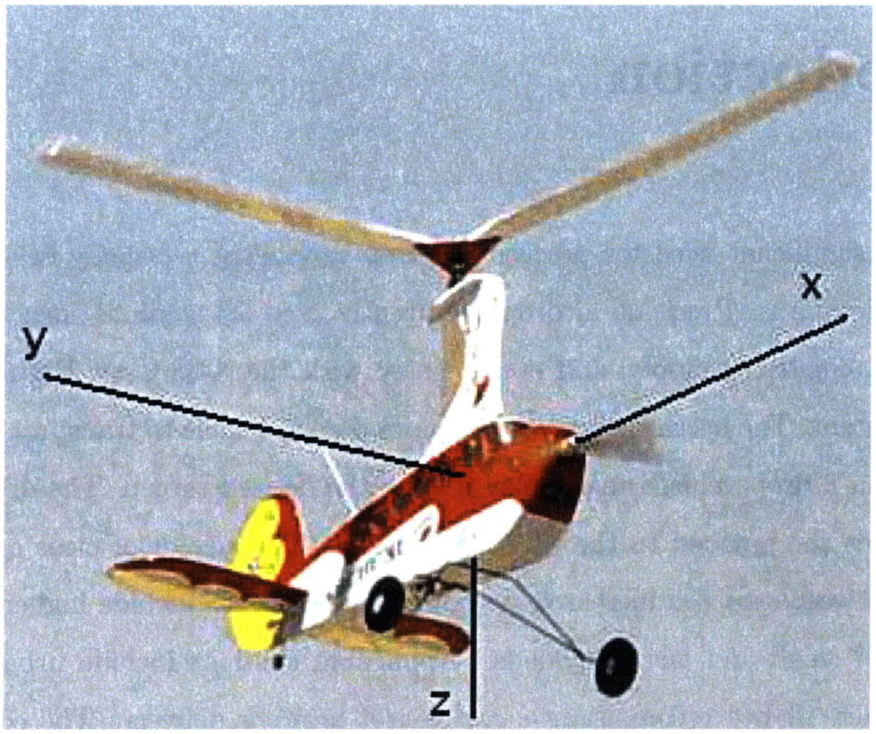

Figure 1-1: GyroBee model aircraft and coordinate system (reprinted with permission from the Autogyro Company of Arizona)

In the precision airdrop scenario, the propeller would be removed, and the vehicle would glide in a controlled descent. The mission scenario for the gliding autogyro resembles that of helicopters performing an unpowered emergency autorotation landing. The air flowing up through the rotor causes it to rotate, and in turn the rotating blades push air down to create lift. The process is similar to a maple seed spiraling its way to the ground. The descent speed performance of a rotor in autorotation has been shown to be similar to that of a parachute of the same diameter.

The advantage of the autogyro strategy is increased glide slope over the ballistic type, and improved control authority and wind resistance compared to the parafoil type. These factors combine to create a drop vehicle with the precision required for modern missions. Although possibly more bulky than a projectile vehicle, folding rotors could reduce packing size and enable deployment from standard airdrop platforms.

This thesis will consider the gliding autogyro vehicle following release from the parent vehicle and in full autorotative forward glide. The task is for the autogyro to autonomously reach a target on the ground. Available control inputs include forward and side tilt of the rotor hub. Some consideration must be made when deciding if the additional complexity and weight justifies an additional third input of rotor collective. For the purposes of the task outlined above, collective will be assumed fixed to reduce controller complexity. The final landing phase, however, may require some combination of backward rotor tilt and collective in order to induce a flare to reduce descent speeds sufficiently for a safe landing. Additionally, collective control may be required to start the rotor in autorotation upon initial deployment from the parent vehicle.

In Chapter 2, a simulation for a gliding autogyro will be developed. Houston [7] has developed a high fidelity simulation for helicopters in autorotation based on dynamic inflow and a blade element method. It was decided, however, that a simplified simulation based upon classical helicopter theory would allow rapid development for proof of concept. Chapter 3 includes a control algorithm and path planning strategy to guide the vehicle to the target area. Chapter 4 develops an estimator to derive state information from sensor measurements, including inertial, magnetic, range finder, and a unique visual measurement that relates the pixel offset of the target from the center of an onboard camera to attitude and position. Finally, Chapter 5 includes the results of running a variety of simulations and shows the accuracy achieved.

[This page intentionally left blank.]

Chapter 2

Simulation Development

In this chapter, the dynamics of the autogyro vehicle are defined, and the resulting simulation is compared with the theoretical results from the literature. The simulation is based upon the Draper Small Autonomous Aerial Vehicle (DSAAV) Dynamics Model [8]. The rigid body equations of motion are presented, followed by force and moment equations created from the aerodynamics of both the rotor and the vehicle body. The software simulation was developed in MATLAB's Simulink environment. This allowed great flexibility and ease of use along with control analysis tools.

2.1 Equations of Motion

First, the six degree of freedom rigid body dynamics equations must be developed. The fundamental relations are between forces and linear momentum rates and torques and angular momentum rates

$$\mathbf{F} = m\dot{\mathbf{v}}^L \qquad\qquad \mathbf{T} = \dot{\mathbf{H}}^L \tag{2.1}$$

with the derivatives taken with respect to the local (inertial) frame, denoted by the superscript L. In these equations the force and torque vectors are composed as $\mathbf{F} = [X\ Y\ Z]^T$, and $\mathbf{T} = [L\ M\ N]^T$. To take the derivatives with respect to the body (vehicle) frame, the transport theorem must be used

$$\mathbf{F} = m\left(\dot{\mathbf{v}} + \boldsymbol{\omega} \times \mathbf{v}\right) \qquad\qquad \mathbf{T} = \dot{\mathbf{H}} + \boldsymbol{\omega} \times \mathbf{H} \tag{2.2}$$

where the body relative velocities are $\mathbf{v} = [U\ V\ W]^T$, and the body relative rotation rates are $\boldsymbol{\omega} = [P\ Q\ R]^T$.

The rigid body angular momentum can be found by the product of inertia and rotation rate. It is assumed that the body coordinate frame is aligned with the principle axes. Therefore the cross coupled inertia terms are set to zero. Then, using the matrix product definition of the cross product,

$$\boldsymbol{\omega} \times \mathbf{v} = \boldsymbol{\Omega}_3 \mathbf{v} = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2.3}$$

the force and moment equations can be rewritten as

$$\begin{aligned} \frac{1}{m}\mathbf{F} &= \dot{\mathbf{v}} + \boldsymbol{\Omega}_3 \mathbf{v} \\ \mathbf{T} &= \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\Omega}_3 \mathbf{I} \boldsymbol{\omega} \end{aligned}$$

to yield the individual state derivatives

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \frac{1}{m}\left(\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} + \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} + \mathbf{C}^T \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \right) + \begin{bmatrix} RV - QW \\ PW - RU \\ QU - PV \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} \left(\begin{bmatrix} L_A \\ M_A \\ N_A \end{bmatrix} + \begin{bmatrix} L_r \\ M_r \\ N_r \end{bmatrix} + \begin{bmatrix} QR\left(I_{yy} - I_{zz}\right) \\ RP\left(I_{zz} - I_{xx}\right) \\ PQ\left(I_{xx} - I_{yy}\right) \end{bmatrix} \right). \tag{2.5}$$

In these equations the forces and moments have been decomposed into the components contributed by aerodynamic effects, the rotor, and the weight. Note the weight vector $\mathbf{g} = [0\ 0\ mg]^T$ must be included with the direction cosine matrix $\mathbf{C}$, which rotates vectors from the body to the inertial frame, and will be defined next. These are the standard equations of motion, with simplified inertia terms, as can be seen in, for example, Stevens and Lewis [17].

Euler angles describe the rotation of the aircraft with respect to the local frame. Refer to Figure 1-1 for a description of the vehicle coordinate system. First a rotation ψ about the local z axis describes the heading angle of the vehicle. Second, a rotation θ about this new frame's y axis describes the pitch angle. Last, a rotation ϕ about the new frame's x axis produces the roll angle.

The matrix that transforms vectors from the body frame to the local frame, the direction cosine matrix, can be found by multiplying the transformation matrices of each successive rotation. The result is defined as $\mathbf{C}$ and is found by

$$\begin{aligned}
\mathbf{C} &= T_3(\psi)T_2(\theta)T_1(\phi) \\
&= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
&= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\pi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \\
&= \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{zz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix}.
\end{aligned} \tag{2.6}$$

The Euler angle state derivatives can be found by relating them to body angular velocities through rotation matrices.

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = T_1^T(\phi)T_2^T(\theta)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + T_1^T(\phi)\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ -\dot{\theta}\sin\phi + \dot{\psi}\cos\theta\cos\phi \end{bmatrix}$$

Solving these equations for the Euler angle derivatives yields the state derivative equations

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} P + (Q\sin\phi + R\cos\phi)\tan\theta \\ Q\cos\phi - R\sin\phi \\ (Q\sin\phi + R\cos\phi)\sec\theta \end{bmatrix}. \tag{2.7}$$

The final state equations are those of position. These are defined in the local coordinate frame for ease of interpretation.

$$\begin{bmatrix} {}_p\dot{N} \\ {}_p\dot{E} \\ {}_p\dot{D} \end{bmatrix} = \mathbf{C}\begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2.8}$$

2.1.1 Quaternions

While the Euler angles are useful for visualizing the rotations, there is a possible singularity in $\dot{\psi}$ at $\theta = 90°$, and they are computationally expensive to work with when compared to quaternions. Quaternions are sets of four parameters that define a magnitude of rotation about a single axis not aligned with the coordinate frame. In this research, Euler angles were used for linearization and modal analysis, where visualization is important, and quaternions were used for simulations, where computational integrity and complexity is important. The quaternion relations are shown here, and the reader is referred to [17] for a more detailed treatment.

The relation between quaternions and Euler angles is

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{bmatrix}. \tag{2.9}$$

The direction cosine matrix with respect to quaternions is

$$\mathbf{C} = \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix} = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_2q_3\right) \\ 2\left(q_1q_2+q_0q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_1q_3-q_0q_2\right) & 2\left(q_2q_3+q_0q_1\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix}. \tag{2.10}$$

Finally, the quaternion state equations are

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{2.11}$$

Note that this can also be written as

$$\dot{\mathbf{q}} = -\frac{1}{2}\mathbf{\Omega}_4\mathbf{q} = -\frac{1}{2} \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.12}$$

2.1.2 Wind

The final considerations are effects of wind. Wind is introduced in the local frame, so the body velocities $[U \; V \; W]^T$ must be converted to local velocities through the direction cosine matrix $\mathbf{C}$. Then the wind velocities can be subtracted and the result converted back to the body frame to achieve the air-relative velocity

$$\begin{bmatrix} U_a \\ V_a \\ W_a \end{bmatrix} = \mathbf{C}^T \left(\mathbf{C} \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \begin{bmatrix} U_w \\ V_w \\ W_w \end{bmatrix} \right). \tag{2.13}$$

During full simulation tests in Chapter 5, the wind is generated by the equation

$$\mathbf{v}_w = (1 + \boldsymbol{\nu}_{wind}) \mathbf{v}_{w_i} G_w. \tag{2.14}$$

The goal was to create a noisy disturbance about the constant wind intensity $\mathbf{v}_{w_i}$. A zero mean Gaussian noise "turbulence" term ν_{wind} with standard deviation σ_{wind} is added to 1 to perturb $\mathbf{v}_{w_i}$. This signal is then shaped by G_w to smooth the turbulence, where G_w is a first order low pass filter with a time constant of two seconds.

2.2 Forces and Moments

Rotor and aerodynamic forces and moments are calculated using a standard treatment for rotorcraft [10]. The primary source of external force is the rotor thrust.

2.2.1 Rotor Forces and Moments

The forces produced by the rotor are governed principally by the rotor rotation rate and the local flow velocity. The primary equations will be presented followed by a brief discussion of their origin.

The main relations driving the rotor forces are for thrust and induced velocity [10]

$$T = \frac{1}{4} (W_b - V_i) \Omega r^2 \rho abc. \tag{2.15}$$

$$V_i = \frac{\eta_{GE} T}{2\rho\pi r^2 \sqrt{U_a^2 + V_a^2 + (W_r - V_i)^2}} \tag{2.16}$$

where Ω is the rotor rotation rate, ρ is the air density, r is the rotor blade radius, a is the blade lift curve slope, b is the number of blades, and c is the blade chord. Other parameters will defined next. Note that in the above equations, T and V_i are functions of each other, and therefore special consideration is required for a solution. Iterative and direct methods have been considered, and are discussed in Appendix A.2.

Local blade velocities must be computed for use in the above equations. W_r is the flow velocity normal to the rotor plane, and W_b is the velocity normal to the blades.

$$W_r = W_a + a_1 U_a - b_1 V_a \tag{2.17}$$

$$W_b = W_r + \frac{2}{3}\Omega r \delta_{col} \tag{2.18}$$

Here a_1 and b_1 are the forward and side flapping angles of the rotor disk, δ_{col} is the (constant) collective angle of the blades, and $\frac{2}{3}$ is used to find the average velocity along the span of the blade.

The thrust equation is based on blade element momentum theory, where the coefficient of thrust is related to the blade tip angle of attack

$$\begin{aligned} C_T &= \frac{T}{\rho \pi r^2 V_{tip}^2} = \frac{\sigma_r a}{4}\alpha_{tip} \\ \alpha_{tip} &= \frac{W_b - V_i}{V_{tip}} = \frac{W_b - V_i}{\Omega r} \end{aligned}$$

where the blade tip speed is $V_{tip} = \Omega r$, and the rotor solidity is $\sigma_r = \frac{bc}{\pi r}$. Solving for T gives Equation (2.15).

The equation for induced velocity is based upon momentum theory, which states that thrust is twice the induced velocity times the mass flow rate through the rotor

$$T = 2V_i \rho \pi r^2 \sqrt{U_a^2 + V_a^2 + (W_r - V_i)^2}.$$

The square root term is the resultant velocity at the rotor assuming a small angle of attack of the rotor disk. Solving for V_i gives Equation (2.16).

Note the factor η_{GE}, which has been added to take into account ground effects when the vehicle is flying close to the ground. The height of the rotor above the ground is the vehicle altitude plus the height of the rotor above the center of gravity rotated into the inertial frame.

$$h_r = -_pD - C_{zz}z_r \tag{2.19}$$

The ground effect factor, negligible for $h_r > r$, is dependent upon an empirical term $K_{GE} \approx 0.03794$ [9].

$$\eta_{GE} = \frac{1}{1 + K_{GE}\left(2\frac{r}{h_r}\right)} \tag{2.20}$$

In addition to the thrust and induced velocity equations, the equations governing the rotor dynamics must be developed. First, induced and profile power will be calculated. The induced power drives the rotor, and is found from the product of thrust and velocity flow through the rotor.

$$P_i = T\left(V_i - W_r\right) \tag{2.21}$$

Profile power is the loss associated with the rotor traveling through the air, and is found from integrating the drag at each element along the blades times the element location and multiplying by the rotation rate.

$$P_p = \Omega b \int_0^r \frac{1}{2}\rho c\left(\text{Velocity}_y^2\right) C_{D_0} y dy \tag{2.22}$$

$$P_p = \frac{\rho F_r \Omega r}{8}\left[\Omega^2 r^2 + 4.6\left(U_a^2 + V_a^2\right)\right] \tag{2.23}$$

The equivalent flat plate area of the blades is defined as $F_r = C_{D_0}rbc$, where C_{D_0} is the zero lift drag coefficient of the blades. The velocity used in the equation above takes into account rotor rotation rate and body velocities [5].

The power can be used to calculate the torque on the rotor, which then propagates the rotor rotation rate state Ω.

$$\tau = \frac{P_i + P_p}{\Omega} \tag{2.24}$$

$$\dot{\Omega} = \frac{-\tau}{I_b b} \tag{2.25}$$

Some additional parameters must be calculated to find the flapping dynamics of the rotor. The Lock number, coning angle [16], and advance ratio are.

$$\gamma_L = \frac{\rho a c r^4}{I_b} \tag{2.26}$$

$$a_0 = \frac{2\gamma C_T}{3a\sigma_r} \tag{2.27}$$

$$\mu = \frac{U_a}{V_{tip}}. \tag{2.28}$$

Using these parameters, the first order flapping sensitivities can be found from the following equations [8].

$$\frac{\partial b_1}{\partial V} = -\frac{2}{V_{tip}}\left(\frac{8|C_T|}{a\sigma_r} + \frac{2V_i}{V_{tip}}\right) \tag{2.29}$$

$$\frac{\partial b_1}{\partial U} = -\frac{4a_0}{3\Omega r} \tag{2.30}$$

$$\frac{\partial a_1}{\partial U} = -\frac{\partial b_1}{\partial U} \tag{2.31}$$

$$\frac{\partial a_1}{\partial V} = \frac{\partial b_1}{\partial U} \tag{2.32}$$

which define the steady-state flapping angles

$$a_{1_{ss}} = \frac{\partial a_1}{\partial U}U_a + \frac{\partial a_1}{\partial V}V_a - \frac{16}{\gamma_L \Omega}Q - \delta_f \tag{2.33}$$

$$b_{1_{ss}} = \frac{\partial b_1}{\partial U}U_a + \frac{\partial b_1}{\partial V}V_a - \frac{16}{\gamma_L \Omega}P - \delta_s \tag{2.34}$$

where δ_f and δ_s are the rotor tilt inputs, forward and side respectively. Actuator dynamics are also included here. A simple lag with input of desired rotor tilt and output of actual tilt

$$H_\delta(s) = \frac{1}{0.04s + 1} \tag{2.35}$$

was used to model the servos driving the rotor tilt inputs, where the break frequency was selected to represent actual radio controlled servos used for hobby aircraft. Additionally, each input was limited to $\pm 15^\circ$ to account for servo and physical limitations.

The flapping state derivatives are

$$\dot{a}_1 = -\frac{\gamma_L \Omega}{16}\left(a_1 - a_{1_{ss}}\right) \tag{2.36}$$

$$\dot{b}_1 = -\frac{\gamma_L \Omega}{16}\left(b_1 - b_{1_{ss}}\right). \tag{2.37}$$

Finally the forces and moments produced by the rotor are collected. The moments are generated by considering the distance of the rotor hub from the vehicle center of gravity. If the flapping angles are assumed small, the equations can be simplified with sine and cosine approximations.

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} -Ta_1 \\ Tb_1 \\ -T \end{bmatrix} \tag{2.38}$$

$$\begin{bmatrix} L_r \\ M_r \\ N_r \end{bmatrix} = \begin{bmatrix} -Tb_1 z_r \\ Tx_r - Ta_1 z_r \\ 0 \end{bmatrix} \tag{2.39}$$

2.2.2 Induced Velocity Near Vertical Descent

A helicopter in autorotation is typically in the vortex ring state, which means the descent rate is close to the downwash velocity. Momentum theory breaks down in the vortex ring state for small advance ratios (below about 0.1). A different process must be used for conditions near vertical descent. Due to the lack of theory, a best fit approximation for induced velocity in vertical descent based on experimental results was used [10]

$$\frac{V_{i_{\text{vertical}}}}{V_h} = 1.15 + 1.125\frac{W_r}{V_h} - 1.372\left(\frac{W_r}{V_h}\right)^2 + 1.718\left(\frac{W_r}{V_h}\right)^3 - 0.655\left(\frac{W_r}{V_h}\right)^4 \tag{2.40}$$

where the induced velocity in hover is $V_h = \frac{mg}{2\rho\pi r^2}$.

Bridging theory states there should be a combination of vertical descent induced velocity and momentum theory induced velocity for low values of μ. A linear combination was chosen, based upon where the vehicle's advance ratio was in the interval between the critical advance ratios, 0.1 and 0

$$\Delta = \frac{0.1-\mu}{0.1}\frac{1}{4} \tag{2.41}$$

$$V_i = (1.0-\Delta)\, V_{i_{\text{momentum}}} + \Delta V_{i_{\text{vertical}}} \tag{2.42}$$

where the weighting term $\frac{1}{4}$ was chosen to match simulation trim outputs with experimental results.

2.2.3 Aerodynamic Forces and Moments

Airflow over the fuselage and tail section of the vehicle produce forces and moments. Drag on the fuselage is calculated from the fuselage drag coefficients, used here in a dimensional form.

$$X_{fus} = \frac{1}{2}\rho X_{uu,fus}\left|U_a\right| U_a \tag{2.43}$$

$$Y_{fus} = \frac{1}{2}\rho Y_{vv,fus}\left|V_a\right| V_a \tag{2.44}$$

$$Z_{fus} = \frac{1}{2}\rho Z_{ww,fus}\left|W_a\right| W_a \tag{2.45}$$

The forces from the vertical tail or fin are calculated by using a lift coefficient. First, the local velocity at the fin (from the spinning vehicle) is found by adding the spin rate times the displacement of the tail from the center of gravity.

$$V_{a.vt} = V_a + x_{vt}R \tag{2.46}$$

Assuming a constant lift coefficient from the fin, the side force is found from

$$Y_{vt} = \frac{1}{2}\rho \left(Y_{uu,vt} \left| U_a \right| U_a + Y_{uv,vt} \left| U_a \right| V_{a,vt} + Y_{vv,vt} \left| V_{a,vt} \right| V_{a,vt} \right). \tag{2.47}$$

This equation does not consider stall effects. To account for this, the maximal side force is calculated using the maximum lift coefficient

$$Y_{vt,max} = \frac{1}{2}\rho Y_{vv,vt,max} \left(U_a^2 + V_{a,vt}^2 \right). \tag{2.48}$$

The fin force is limited in magnitude to this maximum value.

The forces from the horizontal tail are calculated in much the same manner as the fin, but must take into account the downwash from the rotor. The approach is to assume a uniform downwash velocity profile, and to check whether or not the horizontal tail is in the downwash stream. Note that downwash effects only need to be considered if the rotor is pushing air downwards ($V_i > W_a$), which is almost never the case for a gliding autogyro. However, the extra downwash term will be included for completeness and to allow for the possibility of an aggressive flare maneuver during the landing phase of the mission. The term d_{dw} depicts the relative location of the horizontal tail to the downwash.

$$d_{dw} = \frac{U_a \left(z_{ht} - z_{mr} \right)}{V_i - W_a} + x_{ht} - x_r + r \tag{2.49}$$

The first term describes how far the downwash stream "travels" from the rotor to the tail. The last three terms account for the relative location of the tail with respect to the edge of the rotor disk. The local velocity at the tail is then

$$W_{a,ht} = W_a - V_i - x_{ht} Q \tag{2.50}$$

where the term V_i is included only if $0 < d_{dw} < r$ (the horizontal tail is in the downwash stream).

The lift force from the horizontal tail is then calculated from

$$Z_{ht} = \frac{1}{2}\rho \left(Z_{uu,ht} \left| U_a \right| U_a + Z_{uw,ht} \left| U_a \right| W_{a,ht} + Z_{ww,ht} \left| W_{a,ht} \right| W_{a,ht} \right). \tag{2.51}$$

Once again, the maximum lift generated by the tail is limited to

$$Z_{ht,max} = \frac{1}{2}\rho Z_{VV,ht,max} \left(U_a^2 + V_a^2 \right). \tag{2.52}$$

The total aerodynamic forces and moments are collected for use in the equations of motion

$$\begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = \begin{bmatrix} X_{fus} \\ Y_{fus} + Y_{vt} \\ Z_{fus} + Z_{ht} \end{bmatrix} \tag{2.53}$$

$$\begin{bmatrix} L_A \\ M_A \\ N_A \end{bmatrix} = \begin{bmatrix} -Y_{fus} z_{fus} - Y_{vt} z_{vt} \\ X_{fus} z_{fus} - Z_{ht} x_{ht} \\ Y_{vt} x_{vt} \end{bmatrix}. \tag{2.54}$$

2.3 Implementation

The equations of motion and the force and moment equations were implemented in Matlab's Simulink environment in an s-function. This approach allows for easy implementation and modification of both simulation and controller. The specific vehicle parameters used in the simulation are given in Appendix B.1. The values were selected to model a small remote-controlled type helicopter similar to the GyroBee aircraft shown in Figure 1-1. This type of vehicle is consistent with the mission of precise delivery of small payloads.

During the simulation, the state derivatives for body velocities (2.4), body angular rates (2.5), Euler angles (2.7) or quaternions (2.11), local position (2.8), rotor rotation rate (2.25), and rotor flapping angles (2.36), (2.37) are integrated using a fourth order Runge-Kutta numerical method. These derivatives are calculated from the body force and moment equations for the rotor (2.38), (2.39) and vehicle (2.53), (2.54). Given an initial state condition, the integration continues until the autogyro reaches the ground.

2.4 Validation

In order to have some type of validation of the simulation described above, the trim conditions of the simulation were compared with other studies from the literature. The paper by Bailey [2] describes a method for calculating the performance of helicopters. It uses a different set of assumptions and keeps higher order terms than the simulation defined in this paper. A parameterized code was developed by Oliver [13], based upon the work of Bailey, and was found to be experimentally accurate. The parameters of this vehicle model were modified to correspond with Appendix B.1.

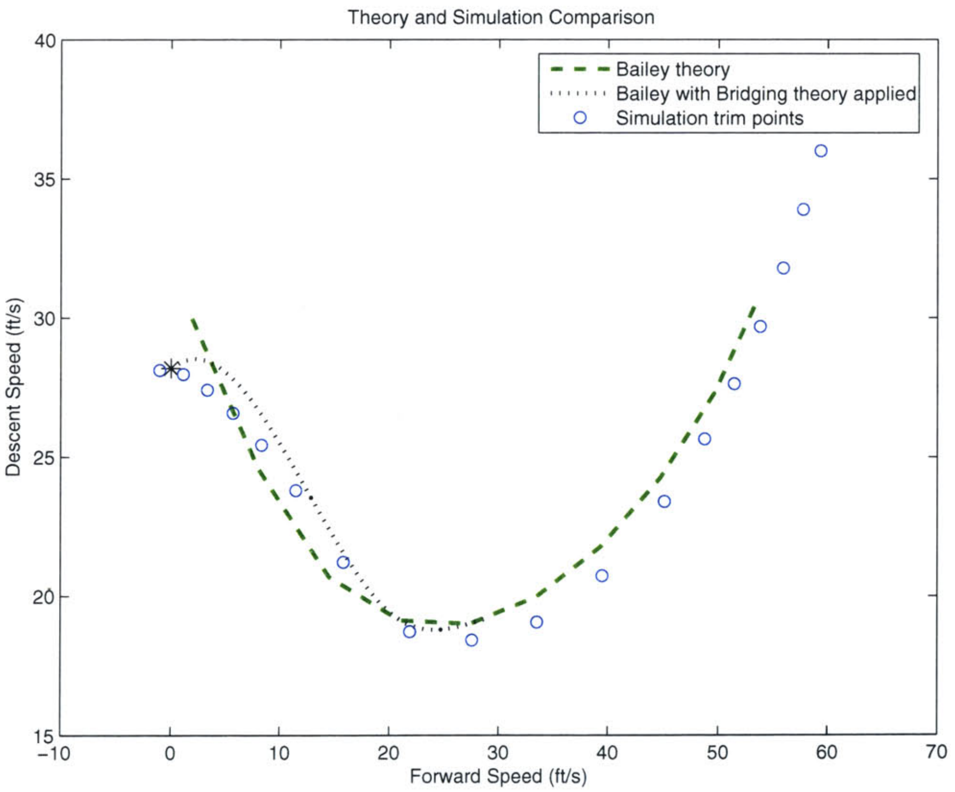

Figure 2-1: Comparison of theory and simulation trim response

A comparison of the output of the code based on the Bailey theory and the simulation is shown in Figure 2-1. The dashed line shows the Bailey theory, and the star shows the vertical descent speed, found from equilibrium arguments. The Bailey theory overestimates the vertical descent speed, highlighting the need for bridging theory, which is represented by the dotted line. Bridging theory dictates a smooth transition between vertical descent and and the Bailey theory for values of advance ratio, μ, larger than 0.1. The simulation trim points, shown as circles, match up well with the theory, promoting validity.

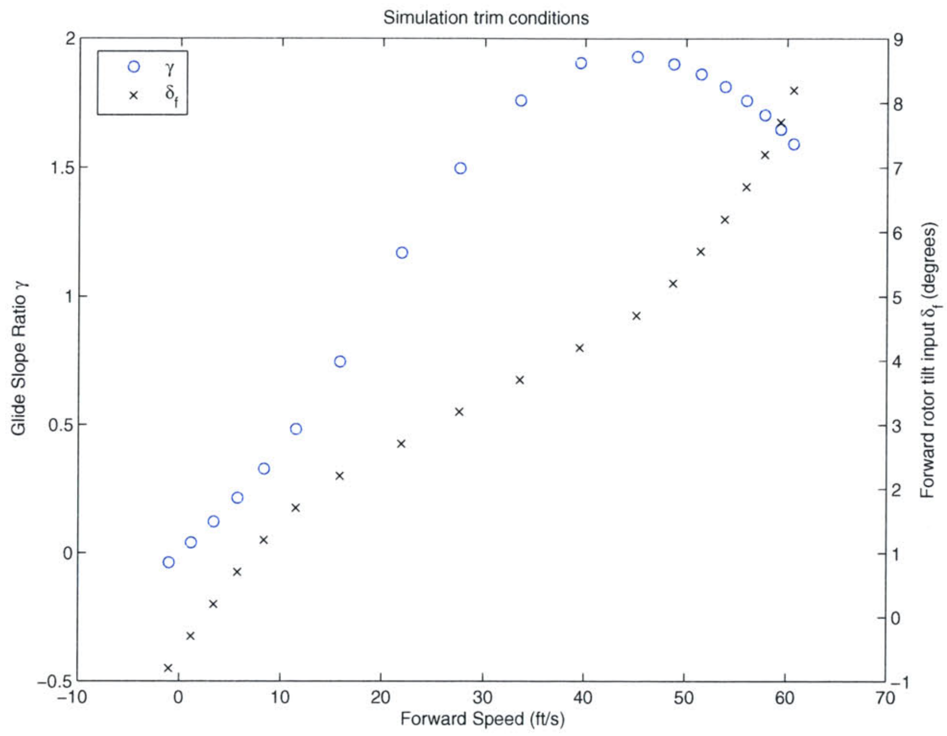

Figure 2-2: Trim forward rotor tilt input and glide slope

Figure 2-2 shows the glide slope ratio, γ, and forward rotor tilt input, δ_f, for each of the trim points in Figure 2-1. The glide slope ratio is defined as the forward speed divided by the descent speed, and is a measure of gliding performance, denoting the achievable range for a given altitude. Note the wide range of achievable glide slopes, providing increased longitudinal control authority compared to parafoil systems which have nearly static glide slopes. Most of the autogyro flight will take place close to the maximum γ of about 2, to yield the maximum range. Although this glide slope ratio is relatively small when compared to gliders or even airplanes, it is good enough, considering the parent vehicle will drop the autogyro either close to the target, or at high enough altitudes to reach the target.

Chapter 3

Guidance Control Design

With the simulation realized, analysis can be performed, and a controller can be designed. In the following section, the nonlinear model is linearized and analyzed. Then the control structure is presented, and the specific controller is designed for the gliding autogyro vehicle. A guidance path planning method is also developed.

3.1 Linearization

In order to use classical control design techniques, a linear model of the autogyro must be obtained. A trim point was selected from Figures 2-1 and 2-2 such that good performance was maintained, and also to be on the rising side of the glide slope curve. This allows the controller to operate more logically so that moving faster yields a larger glide slope. The trim point selected was at a forward rotor tilt of 3.5°, corresponding to a glide slope of $\gamma_t = 1.65$ and a forward speed of 34ft/s. The entire trim state is given in Section B.2, and is denoted by subscript t.

To perform the linearization, the MATLAB command `linmod` was used. This command uses a Simulink diagram and a trim point as an input and calculates a linear system from small perturbation analysis. The result was a state space model of order 15 for the state vector $\tilde{\mathbf{x}}$ and with inputs $\mathbf{u}$.

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{u} \tag{3.1}$$

$$\tilde{\mathbf{x}} = [\phi\ \theta\ \psi\ U\ V\ W\ P\ Q\ R\ {}_pN\ {}_pE\ {}_pD\ \Omega\ a_1\ b_1]^T$$
$$\mathbf{u} = [\delta_f\ \delta_s]^T$$

The specific **A** and **B** matrices are given in Section B.3.

This linear modal can be compared with the nonlinear by considering the outputs of each to an impulse in the separate inputs. Figure 3.1 shows both longitudinal and lateral representative responses. The plots show that the comparison is good and thus the linear model is a good approximation, suitable for linear control design for flight conditions close to a steady glide.

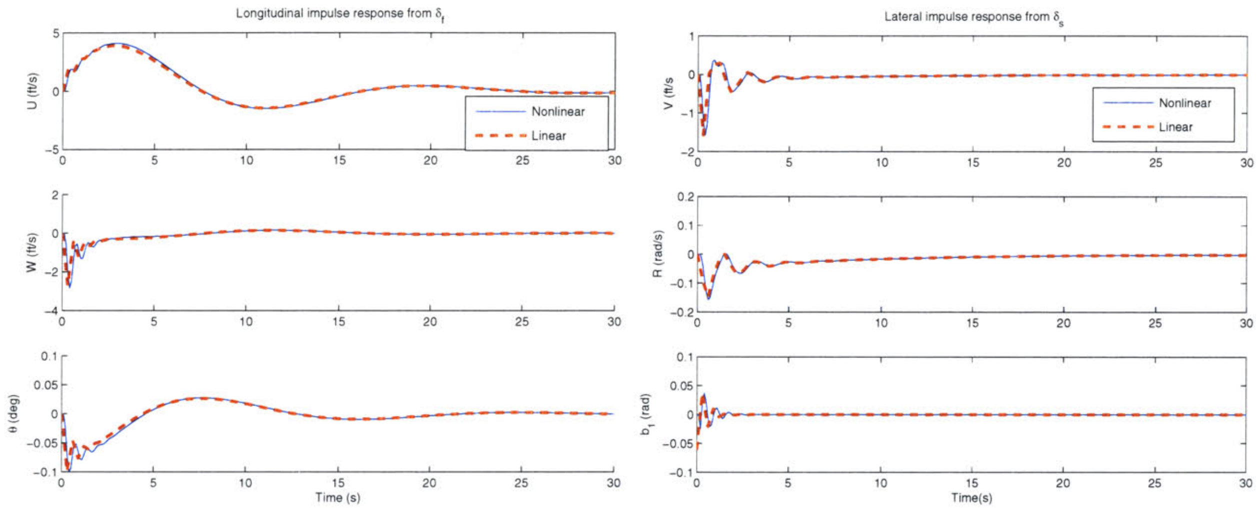

Figure 3-1: Comparison of linear and nonlinear impulse responses

3.2 Modal Analysis

Before proceeding, it is useful to look at the modal decomposition of the linear system to gain insight into the behavior of the vehicle. Figure 3.2 shows the eigenvalues of **A** on the imaginary plane. In this plot, the oblong oval grid lines are points equidistant from the origin (constant natural frequency), and are labeled on the right. The ray grid lines (constant damping) are labeled on the the other sides of the plot.

The first observation to make is that the system is completely separable into lateral and longitudinal modes. This will be important later when setting up the structure

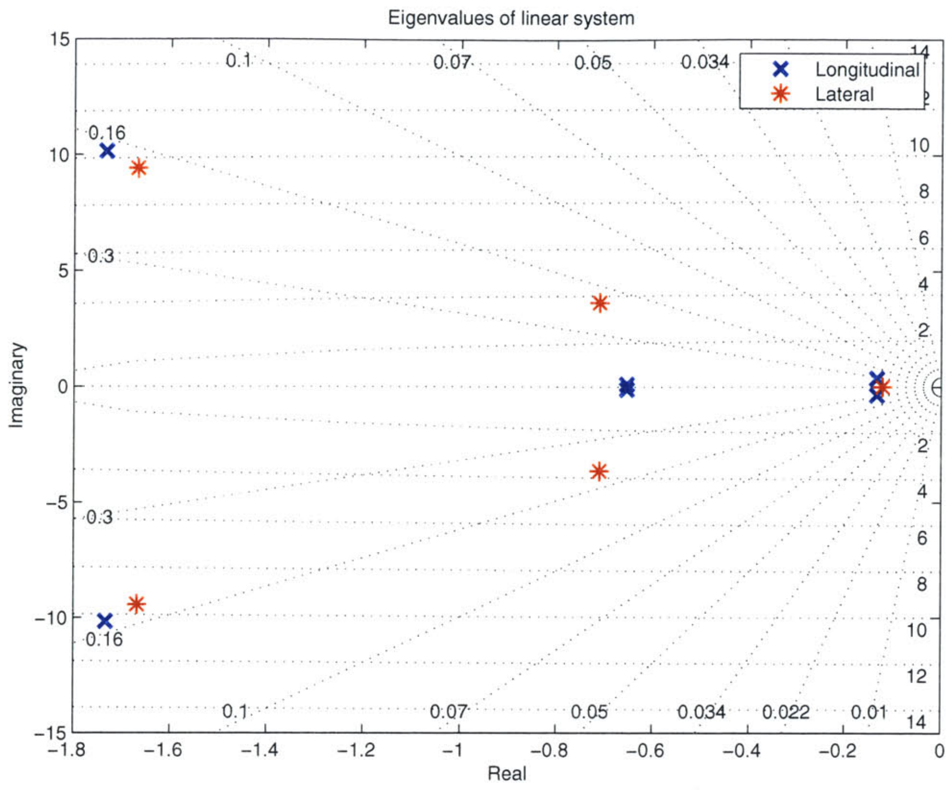

Figure 3-2: Eigenvalues of the linear system on the imaginary plane

of the controller.

By using the real part of the eigenvalues as the initial condition for the linear model, the modes of the system can be visualized. The responses are shown in Figure 3-3.

From the responses, it is clear that the autogyro vehicle has modes similar to those common to normal airplanes. This provides additional confirmation of the system model dynamics, as autogyros have been shown to exhibit modes similar to aircraft [6], [11]. The short period, phugoid, and rotor speed modes are longitudinal, and the flapping, dutch roll, and spiral modes are lateral. Note that there are two additional modes not associated with aircraft (rotor speed and flapping) due to the unique addition of Ω, a_1, and b_1 as states of the system. The eigenvalues and their interpretations are summarized in Table 3.1.

All of the modes have fairly light damping. Additionally, the spiral and phugoid

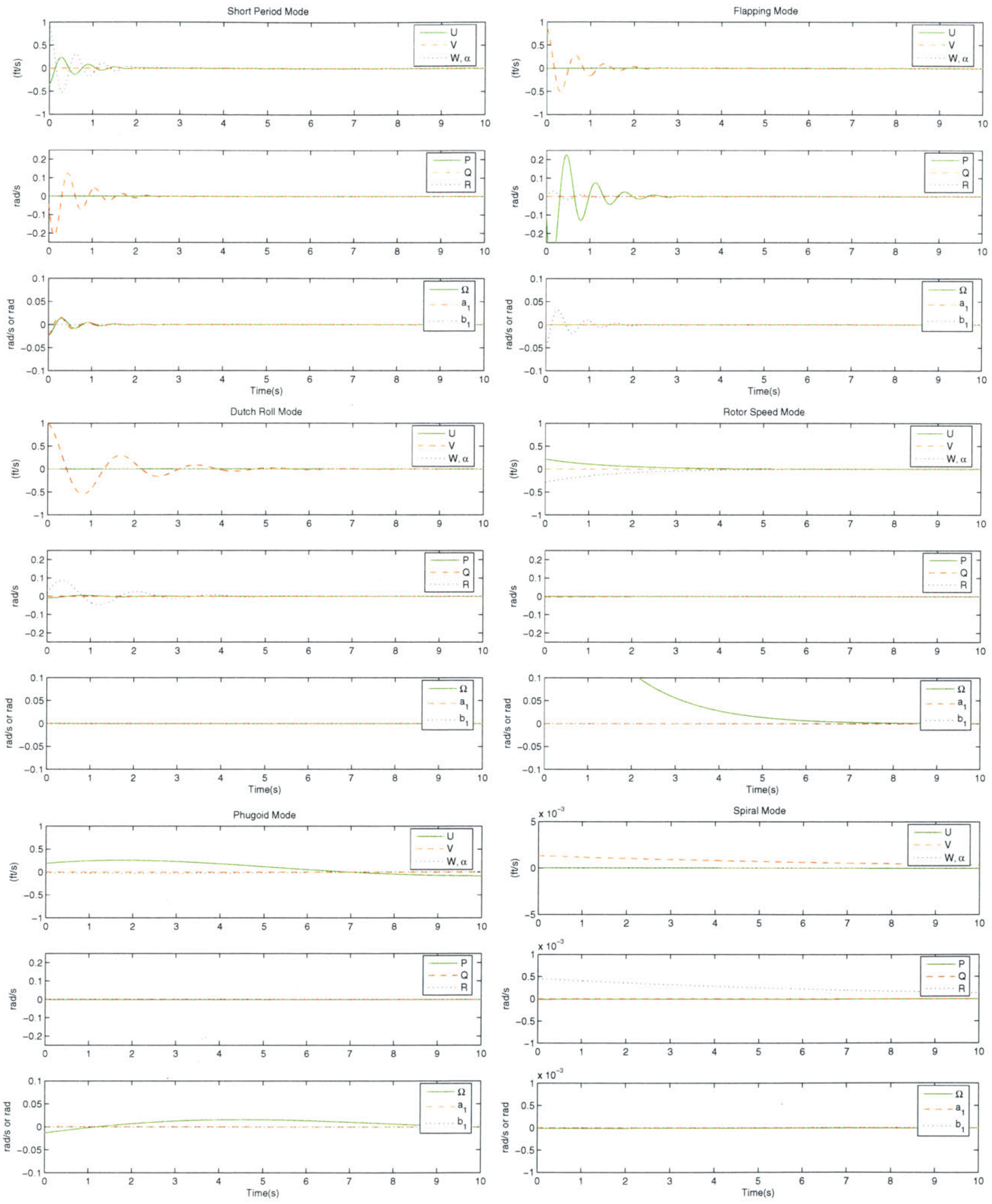

Figure 3-3: Modal response of linear system

Eigenvalue	Interpretation
$-1.73 \pm 10.15j$	Short Period
$-1.67 \pm 8.42j$	Flapping
$-0.71 \pm 3.64j$	Dutch Roll
$-0.65 \pm 0.11j$	Rotor Speed
$-0.14 \pm 0.37j$	Phugoid
-0.12	Spiral

Table 3.1: Eigenvalues of the linear system with their interpretations

modes are very slow in comparison to the short period and dutch roll modes. However, because the spiral mode has little excitation, it is of less importance. Thus the aim of the controller should be to add damping to the modes and decrease response time. Note that it will be difficult to speed up the longitudinal responses because of the slow dynamics.

3.3 Control Structure

Separate lateral and longitudinal controllers can be constructed because of the decoupled modes, and the two can be integrated with minimal risk of interference. Both controllers were designed using classical, multiple loop closure techniques.

The longitudinal controller was based upon getting a good response to a commanded glide slope ratio, γ_c, and then modifying γ_c based upon altitude error. The outer-loop altitude controller was chosen to be proportional integral to eliminate steady-state errors. This is the procedure outlined by Etkin and Reid [3].

The outer-loop lateral controller is based upon the algorithm developed by Park [14]. This nonlinear control law guides a vehicle along a path, and has shown particularly good performance with aircraft flying along both straight and circular paths. First, a reference aim point is selected based on the commanded path. This point is selected to be a distance L_1 away from the vehicle. The angle between the vehicle's horizontal velocity vector and the vector joining the vehicle and the reference point is defined as η. The geometry of this setup is shown in Figure 3-4. The output of the algorithm, which is the lateral acceleration needed to reach the aim point in a circle, is the acceleration command

$$a_{s_{cmd}} = 2\frac{{}_p\dot{N}^2 +_p \dot{E}^2}{L_1} \sin\eta. \tag{3.2}$$

If a coordinated turn is assumed, this acceleration command can be related to a bank angle command. Thus $L\cos\phi = mg$ and $L\sin\phi = ma_s$. Dividing these formulas yields

$$\phi_c = \tan^{-1}\left(\frac{a_{s_{cmd}}}{g}\right) \tag{3.3}$$

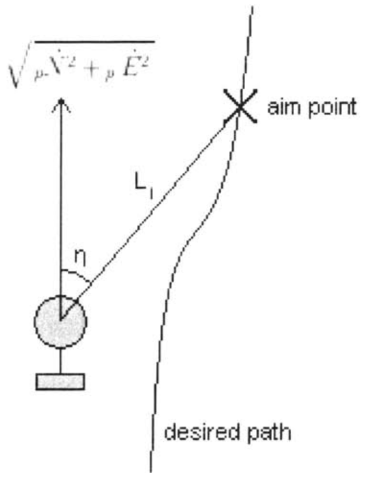

Figure 3-4: Nonlinear guidance diagram

The overall control strategy is shown in the block diagram of Figure 3-5. The control design remaining is to make good roll and glide slope controllers.

3.4 Longitudinal Control Design

The longitudinal controller is based upon a glide slope controller as an inner loop with an outer altitude control loop. The inner loop is shown in Figure 3-6. The controller for γ is designed with a multiple loop closure strategy. In each successive loop, the outer controller must be slower than the inner. This allows outer loops to assume stability of inner loops. Therefore care must be taken to ensure that inner loops perform very well.

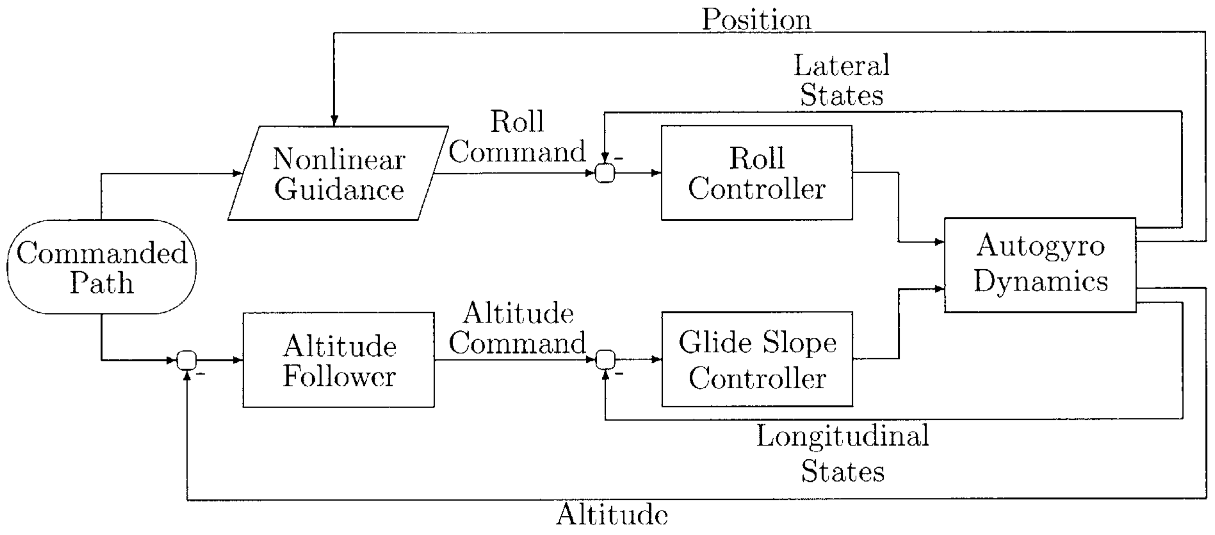

Figure 3-5: Control structure

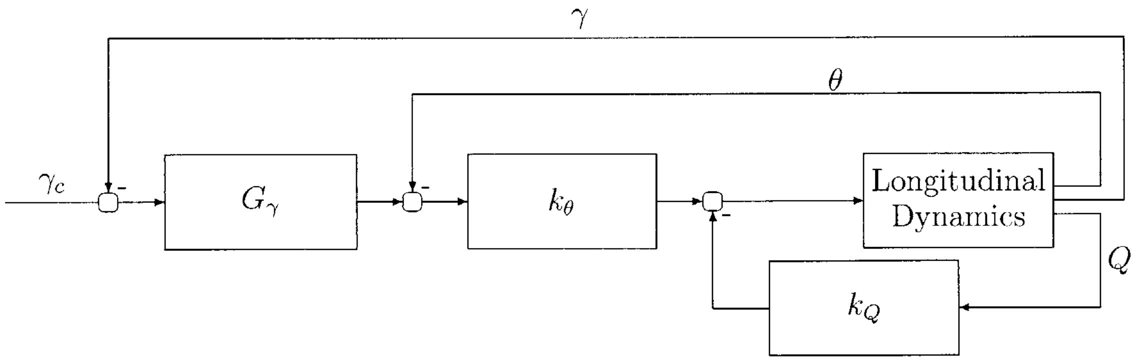

Figure 3-6: Longitudinal glide slope controller

First, pitch rate, Q, and pitch attitude, θ, are fed back in successive loops through simple gains. More aggressive controllers were considered, but the required control authority was unreasonable. The corresponding bode plots of each successive system are shown in Figure 3-7. The figure shows the systems both compensated and uncompensated. The dotted line represents just the plant, and the solid line represents the forward transfer function with the controller in place. In both cases the gain selected increased the time response of the system while maintaining large phase margins.

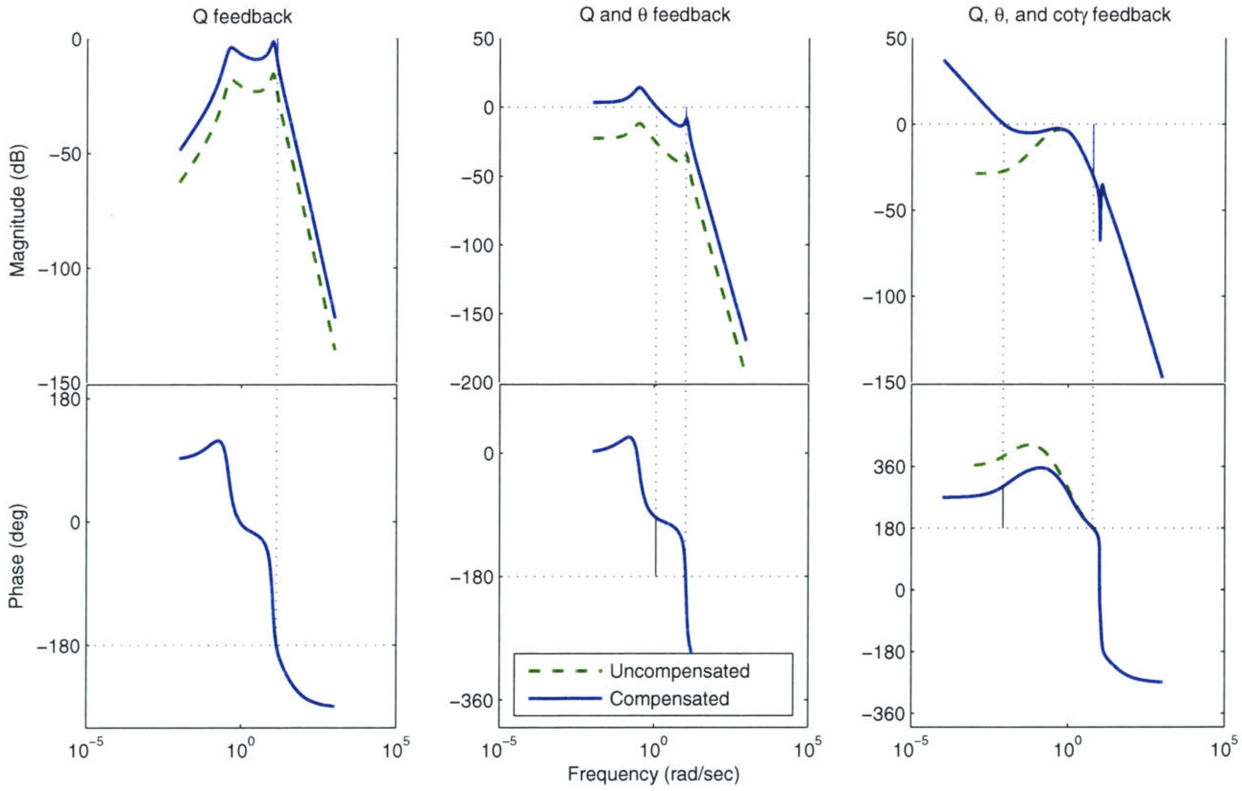

Figure 3-7: Bode plots of successive longitudinal inner loop transfer functions, from δ_f to the controlled variable

In order to gain access to the glide slope as an output from the linear model, it must be created as a state in the system. This can be done by observing that the glide angle is $\theta - \alpha$, where α is the angle of attack of the autogyro. Note that the glide angle is the angle between the horizontal and the velocity vector, which is equivalent to cotangent of the glide slope. For relatively constant forward speeds with small angles of attack, α can be approximated as $\tan^{-1}\left(\frac{W}{U}\right) \approx \frac{W}{U_{trim}}$. Thus the new state is added $\cot\gamma = \frac{W}{U_{trim}} - \theta$.

To eliminate steady state errors in the glide angle, a PI controller was used,

$$G_\gamma = k_\gamma \frac{s+0.2}{s}. \tag{3.4}$$

The last bode plot of Figure 3-7 is a comparison of the forward transfer function from δ_f to $\cot\gamma$, with and without compensation. The integral feedback trades some phase margin and speed for eliminating steady state error. A plot of the poles of the compensated system is shown in Figure 3-8. The original, open-loop poles are also shown for comparison. Note that our objective of increasing speed and damping, especially of the dominant phugoid mode, has been achieved.

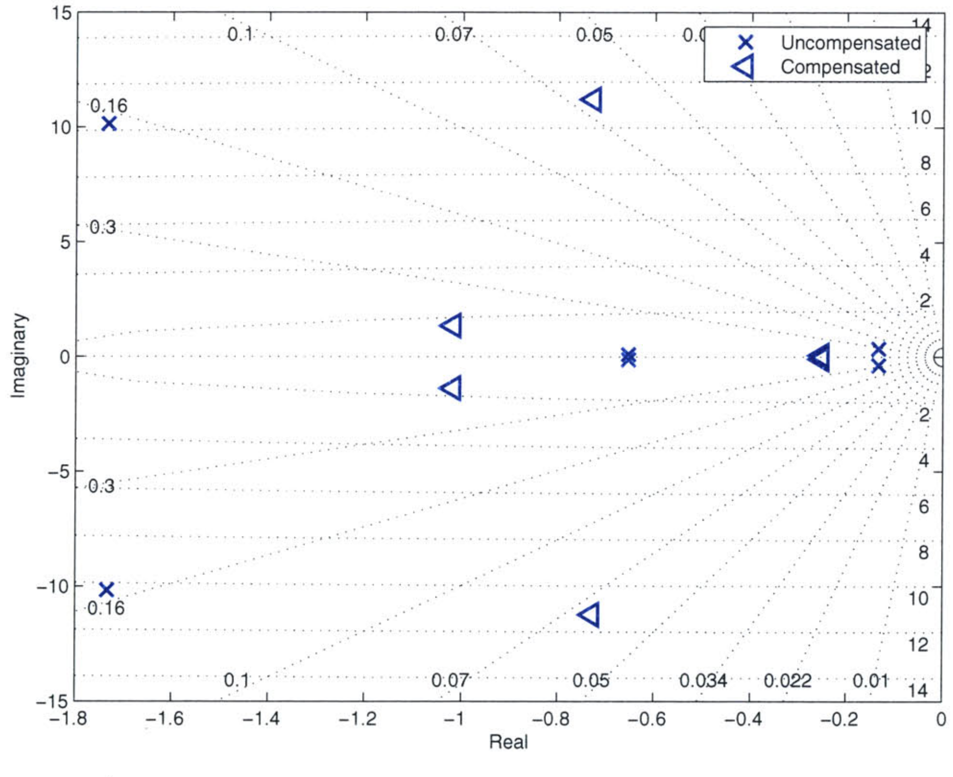

Figure 3-8: Eigenvalues of inner-loop controlled longitudinal system

The entire inner-loop glide angle controller was implemented with the linear system as in Figure 3-6, except the more accurate definition of the glide slope being the forward speed divided by the descent speed was used. The response of this augmented system to a step in $\cot \gamma_c$ is shown in Figure 3-9. Although the response is not particularly fast (with a rise time of about five seconds), it will be adequate for the needs of the autogyro following a benign ramp altitude profile. Also note the undershoot caused by the initial increase in descent speed from the loss of lift associated with tilting the rotor forward.

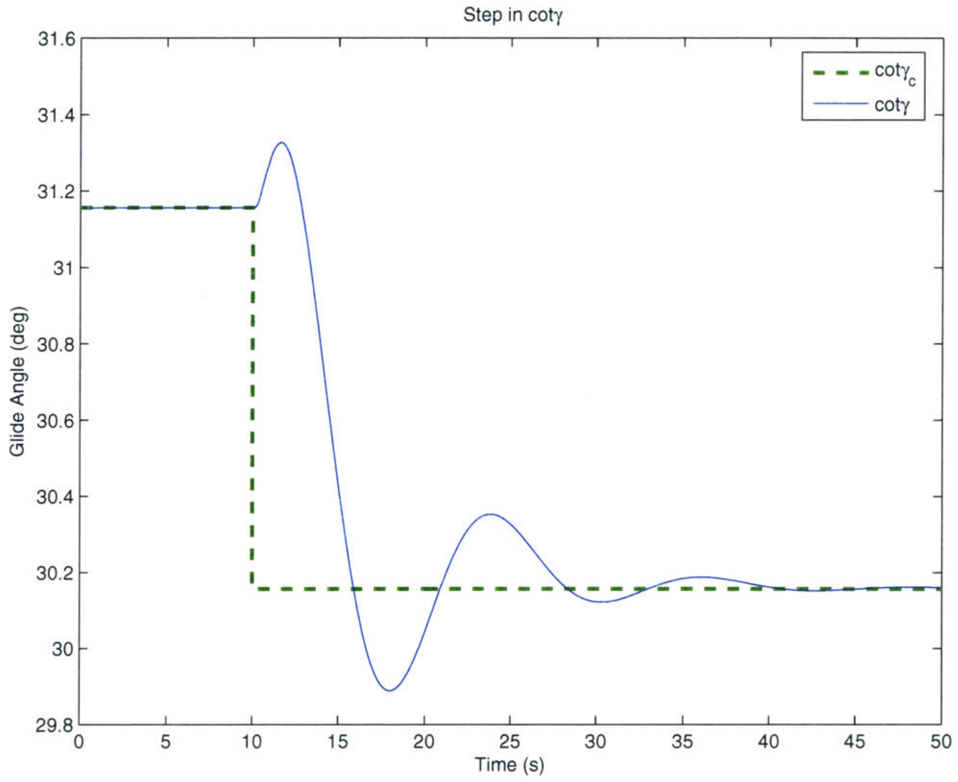

Figure 3-9: Response of linear model to step in γ_c

The final step of the longitudinal controller design is the altitude controller. The altitude of the autogyro is compared to the commanded altitude to form an error. This error is used to adjust the commanded glide angle to achieve the desired altitude. A proportional-integral controller was used

$$G_h = k_h \frac{s + 1/30}{s}. \tag{3.5}$$

The output of the controller is limited to the maximum trim glide slope achievable by the vehicle so that unreasonable commands are not issued. Since this is the second integrator in the loop, there will be zero steady-state error to a ramp command (the nominal altitude profile).

The altitude controller was implemented on the linear model. Saturations of inputs and commanded glide slopes were also included. A response was generated for a commanded altitude profile along the nominal glide slope. To illustrate the response dynamics, the initial altitude of the vehicle was offset by 200 feet from the commanded profile. The response is shown in Figure 3-10. Initially, the vehicle is below the commanded path, so the glide slope is commanded at maximum until the vehicle approaches the commanded altitude.

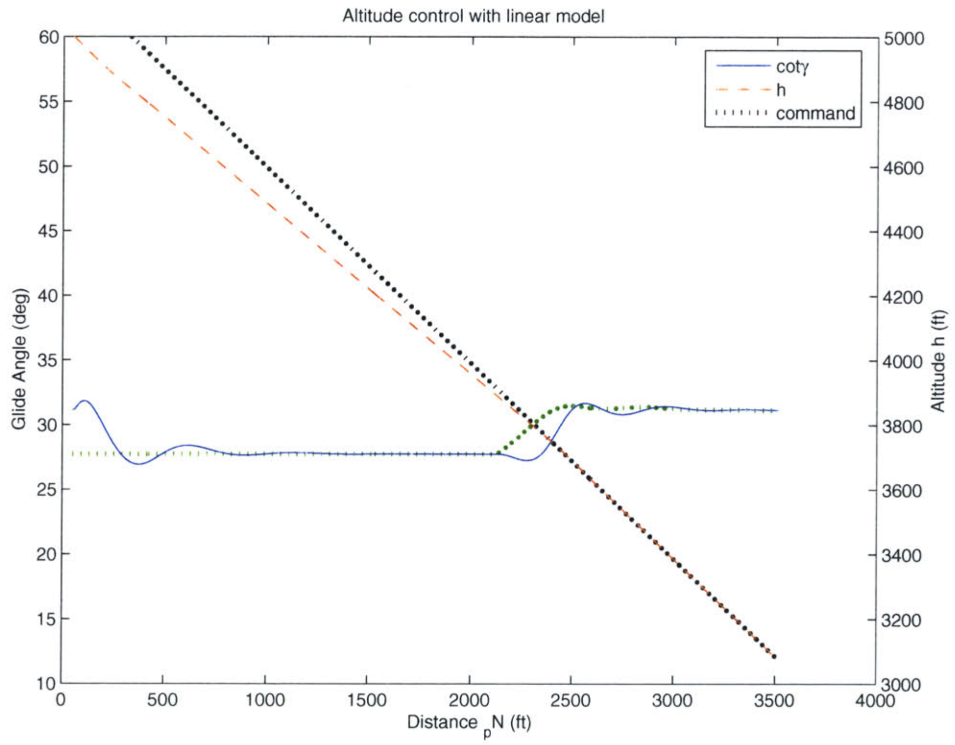

Figure 3-10: Response of linear model to initial altitude offset

3.5 Lateral Control Design

The lateral controller is similar to the longitudinal, except in this case the inner loop is a bank angle controller, and the outer loop is the nonlinear guidance law. An overview of the inner ϕ controller is in Figure 3-11.

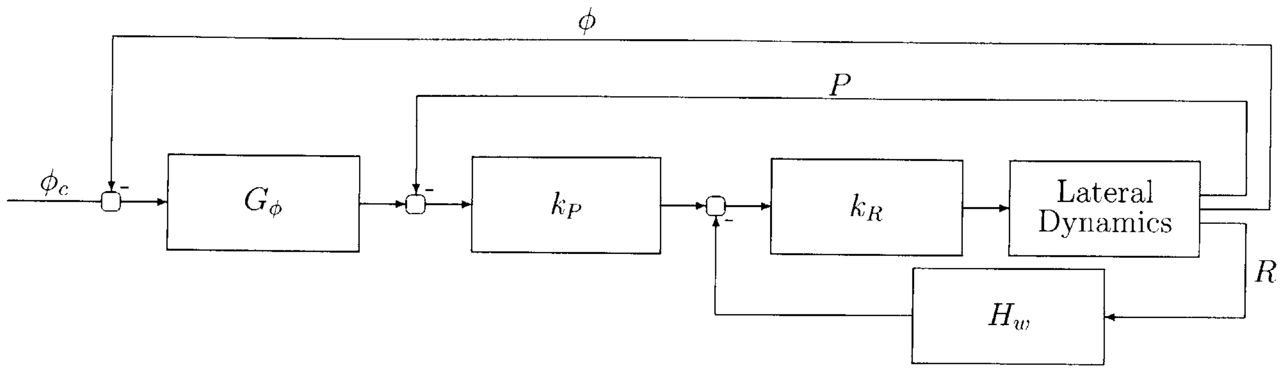

Figure 3-11: Lateral bank controller

A similar approach to that from above is used to get a good ϕ response. First, R is fed back using proportional control. Observe, however, that sustained turns are required for path following, implying that these steady turn rates should not be driven to zero. Therefore a washout high-pass filter is used in the R feedback loop, so that the controller does not act upon the low frequency steady turn rates.

$$H_w(s) = \frac{4s}{4s+1} \tag{3.6}$$

The bode plot of the forward transfer function from δ_s to R, with and without the controller with washout, is shown in the first plot of Figure 3-12.

Second, P is fed back through a pure gain as the next loop. The corresponding bode plots are shown in the second graph of Figure 3-12. Here the gain was selected to increase system response speed.

Finally ϕ was fed back through a PI controller

$$G_\phi(s) = k_\phi \frac{s+1}{s} \tag{3.7}$$

so that a bank angle can be tracked without error. The gain k_ϕ was selected to maintain ample gain and phase margins on the third plot of Figure 3-12. The eigenvalues of the completed inner-loop control can be seen in Figure 3-13 (note the extra state due to the washout filter). The spiral mode has been sped up, along with adding damping to the dutch roll mode.

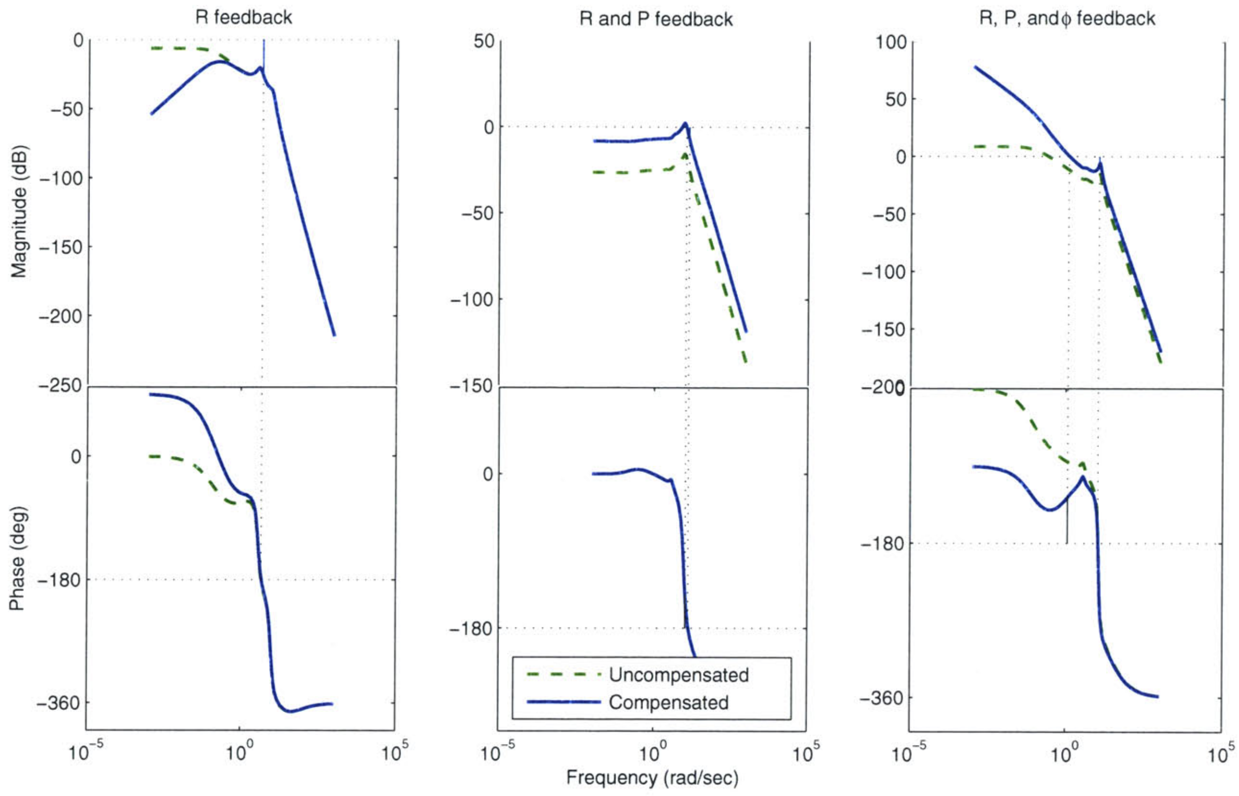

Figure 3-12: Bode plots of successively closed lateral inner loop transfer functions, from δ_s to the controlled variable

The response of the linear system to a step input in ϕ_c is shown in Figure 3-14. The relatively fast response of this controlled system will help the outer loop path tracking algorithm.

The commanded bank angles are generated from the nonlinear guidance logic described by Equations (3.2) and (3.3) in Section 3.3. In this algorithm, the reference point distance L_1 acts like an inverse gain: a low value represents an aggressive controller that could go unstable, and a high value represents a sluggish controller that will cut corners of the path. The value of L_1 for the lateral controller was selected to provide good tracking of circular paths with radii larger than about 200 feet. Additionally, the output of the guidance algorithm was limited to bank angle commands of ± 15 degrees, both to eliminate infeasible commands for large heading

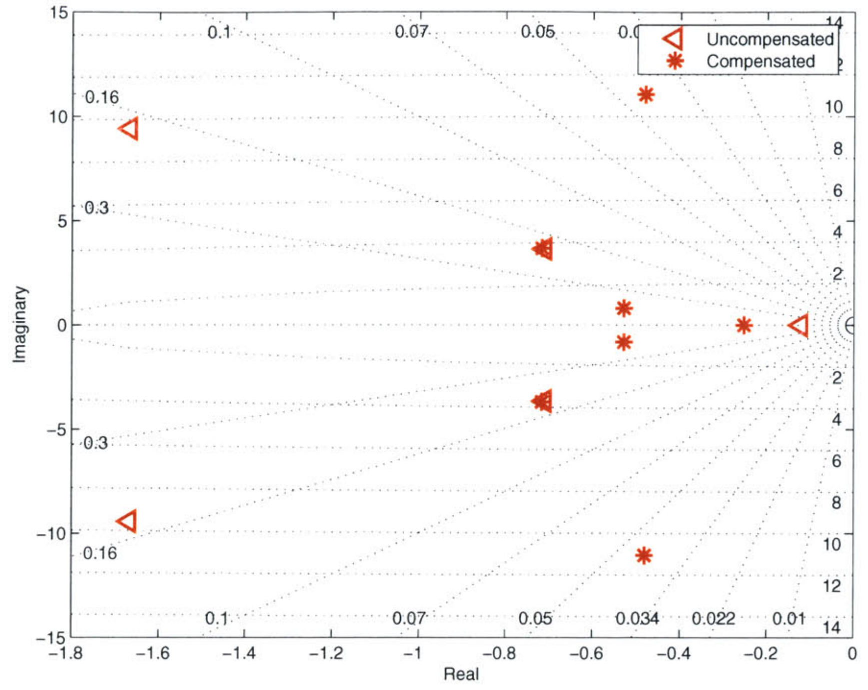

Figure 3-13: Eigenvalues of inner-loop controlled lateral system

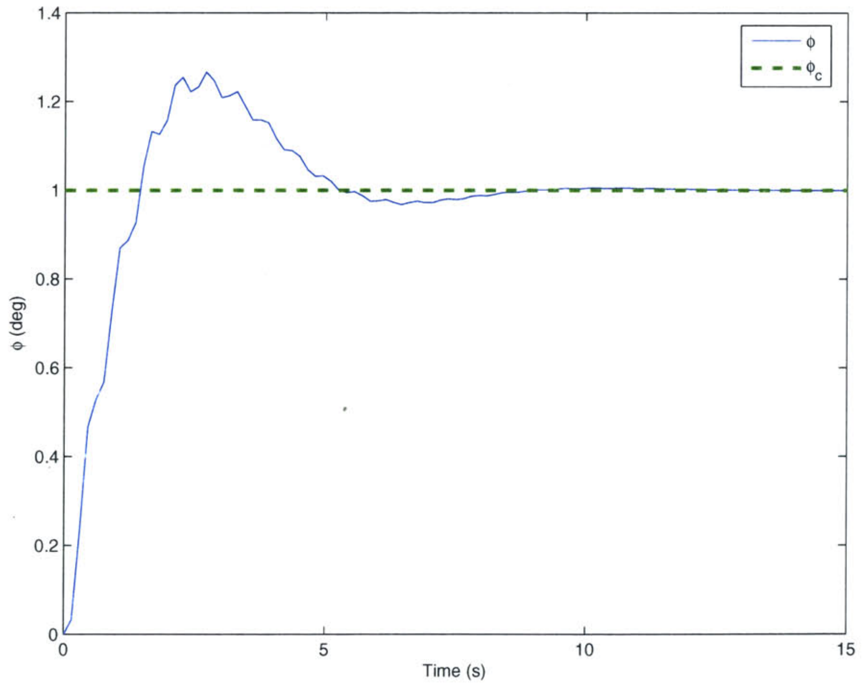

Figure 3-14: Response of linear model to step in ϕ_c

deviations and to maintain close proximity to the trim point.

The full lateral controller was implemented on the linear model, and responses were found to various commanded paths, which can be seen in Figure 3-15. The more interesting case is the first, with a circle following task, here with radius 200ft. This is important because the path planning algorithm developed in the next section will use a combination of straight lines and circles to guide the autogyro to the target. The second path is a sine wave steadily increasing in amplitude. The tracking performance degrades when the radius of the turn becomes too small (below about 200 feet).

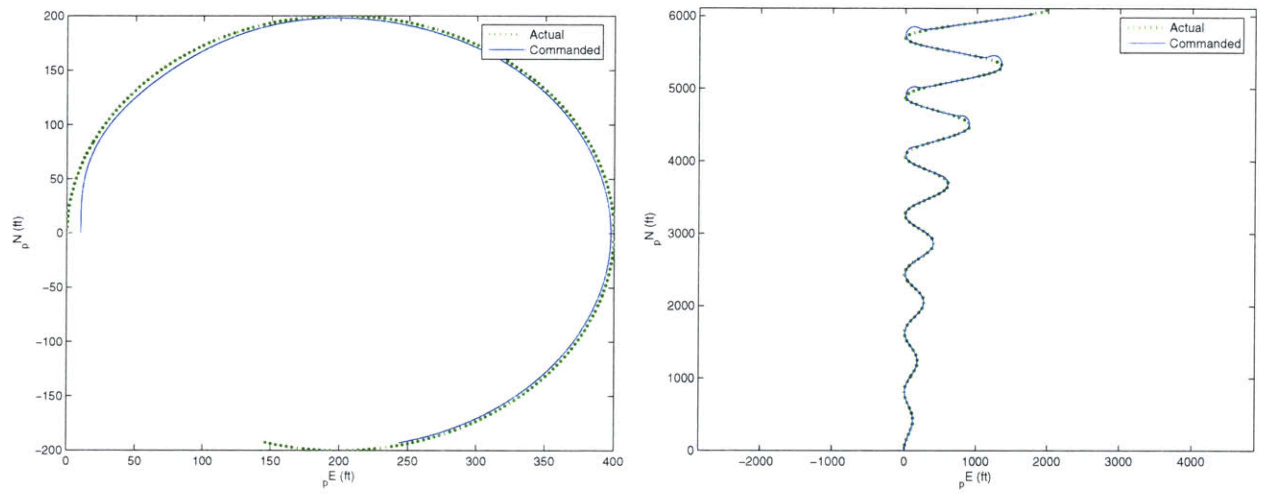

Figure 3-15: Response of linear model to various commanded paths

The parameters of both the longitudinal and lateral controllers are summarized in Table 3.2.

	k_Q	k_θ	k_γ	k_h
Longitudinal	-5	-20	-1	0.0026
	k_R	k_P	k_ϕ	L_1
Lateral	-1	8	3	100

Table 3.2: Controller parameter summary

3.6 Path Planning

An algorithm must be developed to create a path from a somewhat arbitrary initial starting location to the desired target location. The approach is geometrical assuming a constant glide slope. The path will be comprised of straight segments and circular turns, thereby taking advantage of the lateral control structure. The first segment is straight, permitting control and estimation initialization and allowing the vehicle to settle into its trim condition. Next comes a circular turn, followed by a second straight gliding segment. Then there is a final circular turn, and the final approach leads the vehicle to the target.

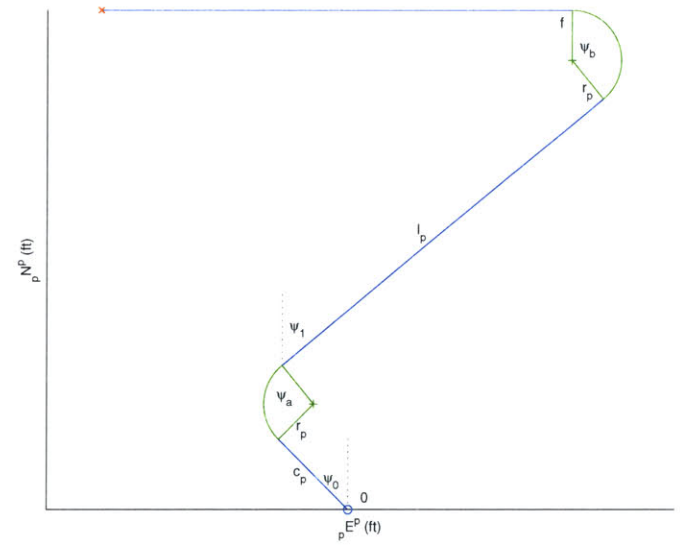

Figure 3-16: Trajectory Setup

Figure 3-16 shows the details of the flight path. Without loss of generality, a new axis can be defined (with superscript p) such that the final approach leg is along the x axis and the target is at the origin, requiring only a translation and rotation of coordinate frames.

The figure shows the key variables used in the algorithm. Each transition point is depicted by the subscript corresponding to the appropriate label on the figure. The known parameters are first defined. The starting point is the 0 location: $_pN_0^p$, $_pE_0^p$, $_pD_0^p$. The initial heading is an angle ψ_0. The (assumed constant) glide slope is taken to be the trim glide slope γ_t, defined as the trim forward velocity divided by the trim descent velocity. Finally, the initial settling leg length c_p is also known.

Because of the cumbersome notation, assume that the path and local frames are aligned so the superscript p can be dropped. It is easy to generalize to the path frame case by transformation of the transition points.

Using these known parameters, a series of equations will be defined to solve for a series of unknowns. The first turning angle, ψ_a turns the vehicle to a new heading ψ_1. The vehicle will stay on this heading throughout the gliding segment's length l_p. The vehicle makes the final turn through an angle ψ_b to reach the final approach leg, which begins at location ${}_pN_f$ and altitude $-{}_pD_f$, and ends at the target location (the origin).

So the objective of the algorithm is given parameters $\{c_p,\ \psi_0,\ {}_pN_0,\ {}_pE_0,\ {}_pD_0,\ \gamma_t\}$, find the quantities $\{\psi_a,\ \psi_b,\ l_p,\ {}_pN_f,\ {}_pD_f,\ \psi_1\}$. Thus six equations are needed for the six unknowns.

First, consider the altitude at each of the transition points. The altitude after completing the second turn must be constrained to be the final approach transition point altitude, ${}_pD_f$. The altitude lost in a given distance can be found by realizing that the glide slope ratio, γ_t, is the distance traveled divided by altitude lost when in a trim condition.

$$-{}_p D_0 - \frac{c_p}{\gamma_t} - \frac{r_p\psi_a}{\gamma_t} - \frac{l_p}{\gamma_t} - \frac{r_p\psi_b}{\gamma_t} = -{}_pD_f \tag{3.8}$$

A simple angle observation produces another two equations and simplifies the following analysis.

$$\psi_b = \psi_1 + \frac{\pi}{2} \tag{3.9}$$

$$\psi_a = \psi_1 + \psi_0 \tag{3.10}$$

Lateral constraints similar to the altitude equation yield two additional equations (for ${}_pN$ and ${}_pE$). First, however, the turn geometry must be investigated in more detail (see Figure 3-17). Considering the first turn, a rotation of axes by ψ_0, such that the initial heading of the turn in the new frame is vertical, is shown in the figure. The change in ${}_pN$ along the turn in this frame is $r_p - r_p \cos\psi_a$, and the change in ${}_pE$ is $r_p \sin\psi_a$. After rotating back into the path frame, we have the change in position ${}_pN$ after the turn is $(r_p - r_p\cos\psi_a)\cos\psi_0 - r_p \sin\psi_0$, and the change in ${}_pE$ is $r_p \sin\psi_a \cos\psi_0 + (r_p - r_p \cos\psi_a)\sin\psi_0$. Thus the equations defining the constraints between the initial and final North/East positions are

$$\text{North} :_p N_0 - c_p \sin\psi_0 + (r_p - r_p\cos\psi_a)\cos\psi_0 - r_p \sin psi_a + l_p \sin\psi_1 - r_p \cos\psi =_p N_f \tag{3.11}$$

$$\text{East} :_p E_0 + c_p \cos\psi_0 + (r_p - r_p\cos\psi_a)\sin\psi_0 + r_p \sin\psi_a \cos\psi_0 + l_p \cos\psi_1 + r_p \sin\psi_1 + r_p = 0. \tag{3.12}$$

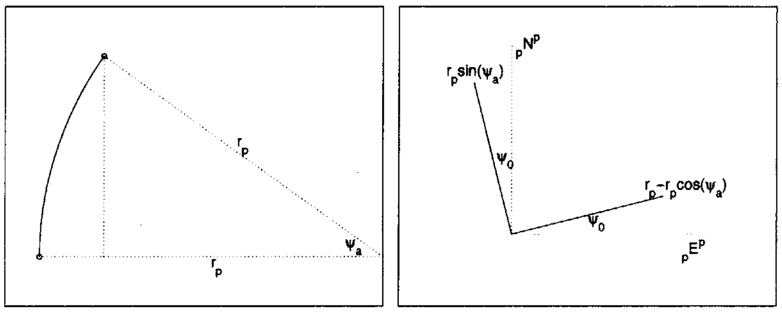

Figure 3-17: Detailed geometry of the first turn in the commanded path

The last equation comes from the constraint on the glide slope during the final approach leg.

$$\frac{{}_pN_f}{{}_pD_f} = \gamma_t \tag{3.13}$$

In general, the approach glide slope could be set to a different value than the glide slope of the rest of the path, but here it is kept constant for simplicity.

These equations can be solved iteratively, or a numerical solver can be used. In the latter case, equations (3.8) through (3.13) can be combined into one equation of ψ_1 only. This can be solved, and the other parameters easily follow.

3.6.1 Steady Wind

An additional challenge arises when wind is introduced into the simulation. When flying into the wind, the ground relative glide slope is decreased, causing a drop in the range of the vehicle. It can be be assumed that the wind speed is known by the autogyro either by uploaded information from the parent vehicle or an onboard estimation scheme. This motivates creating an altered path planner based upon known wind speeds: using this extra piece of information to create a path that is followable by the autogyro.

The strategy is to create a wind-frame coordinate system. These axes are translated from the path axes according to the wind speed and the trim rate of descent of the vehicle. Each point in the wind frame takes into account the extra wind relative distance that must be traveled to reach the corresponding point in the path frame. The transformation to the wind frame is given by

$$ {}_pN^w =_p N - \frac{{}_pD}{{}_p\dot{D}_t} U_w \tag{3.14} $$

$$ {}_pE^w =_p E - \frac{{}_pD}{{}_p\dot{D}_t} W_w \tag{3.15} $$

where ${}_p\dot{D}_t$ is the trim rate of descent. Thus the wind frame is shifted in the direction of the wind with increasing magnitude as altitude increases.

This transformation implies that a path created in the wind frame will have a constant glide slope throughout the flight. Thus the path planning algorithm is altered by using inputs of the initial vehicle position in the wind frame to create a wind relative path. This path is then followed by the controller using wind relative states.

An example result of the path generation algorithm is shown in Figure 3-18. The solid and dashed lines are the results when a steady wind of $U_w = -2$ft/s (in the ${}_pN$ direction), $W_w = 10$ft/s (in the ${}_pE$ direction) is used. Notice how the wind relative path is shifted in the direction of the wind according to altitude. The dotted line shows the result with no wind. The outcome is exactly what is expected and required. With wind, the final approach point is shifted closer to the target, so less ground distance must be traveled, and a constant wind relative glide slope can be used. Additionally, the point is shifted up, so the final approach leg can be flown, and the wind will push the vehicle towards the target.

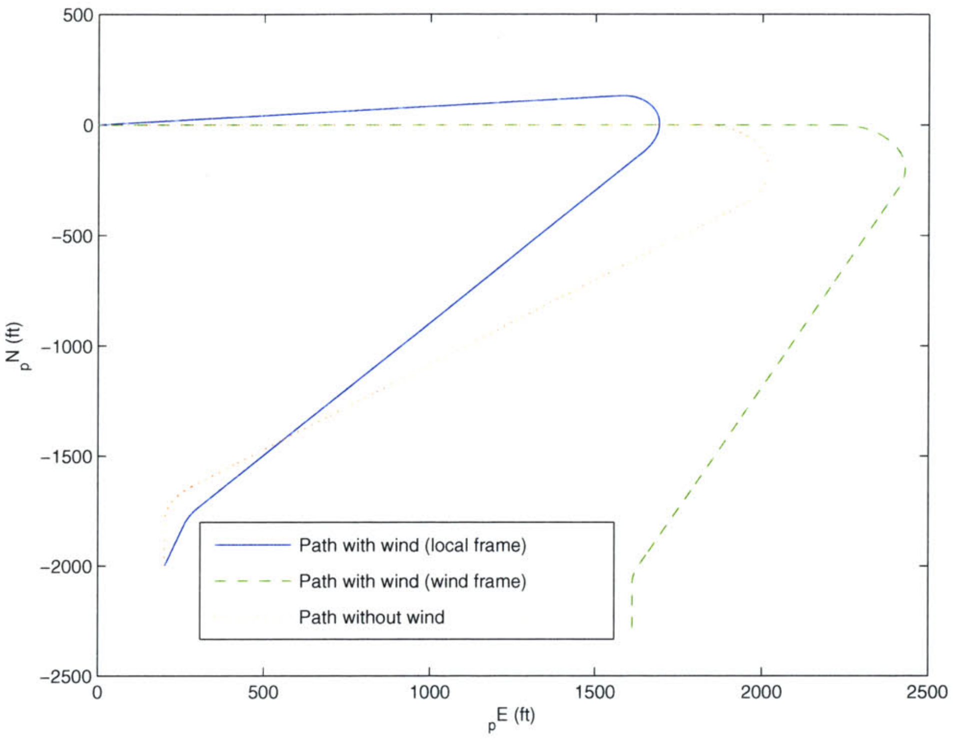

Figure 3-18: Path generation with and without wind $U_w = -2\text{ft/s}$, $W_w = 10\text{ft/s}$

3.7 Simulation Results

The lateral and longitudinal controllers were implemented on the full nonlinear simulation from Chapter 2 using the steady wind condition $U_w = -2\text{ft/s}$, $W_w = 10\text{ft/s}$. The output of the autogyro following the commanded path toward the target at the origin is shown in Figure 3-19.

The first graph of the figure shows good tracking of the commanded path, even in the face of a wind disturbance which is a significant portion of the autogyro's trim forward speed (34ft/s). The next plot shows the tracking of glide slope and

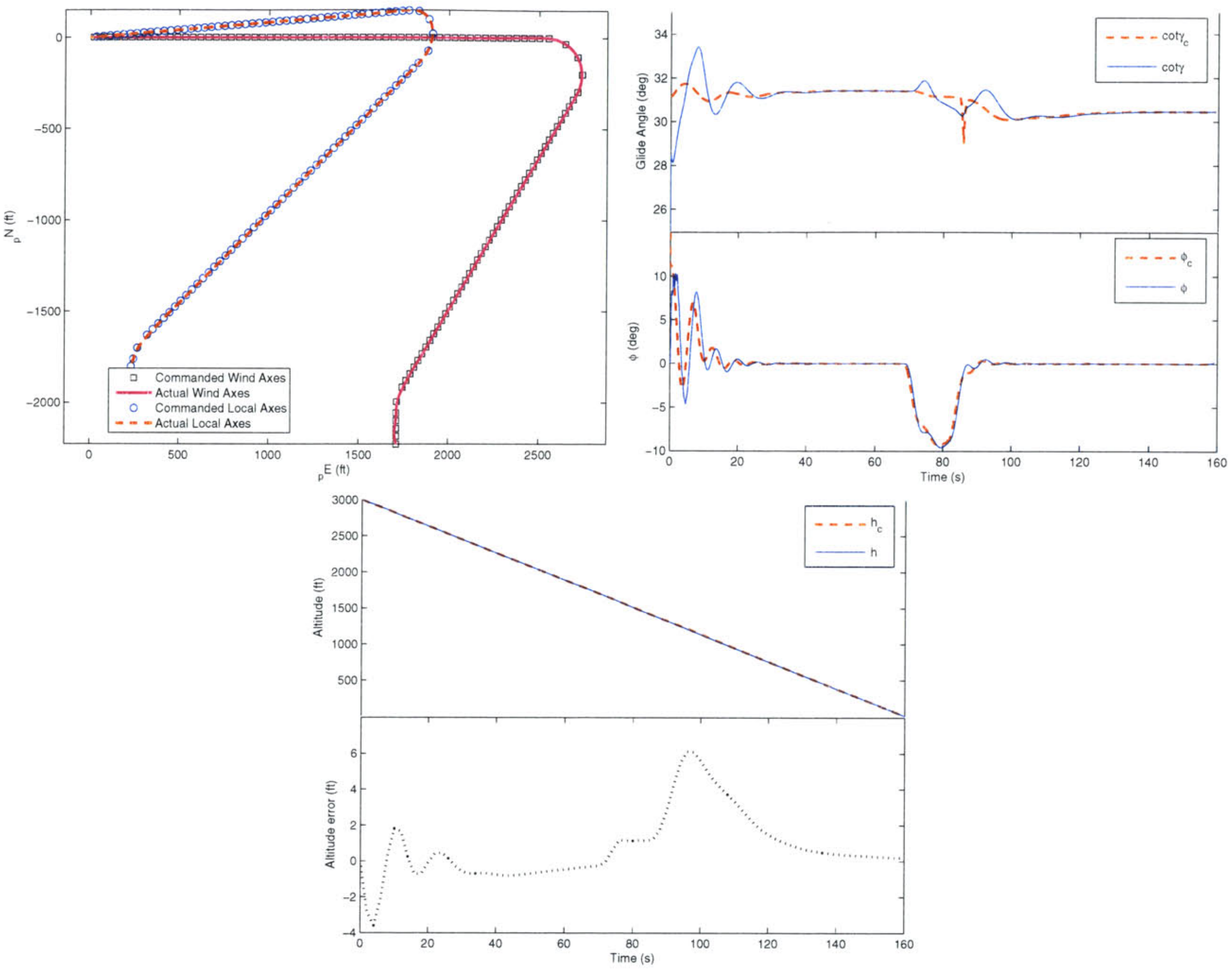

Figure 3-19: Controlled simulation result

bank angle. As the vehicle makes the long second turn (negative bank corresponding to left side down) to reach the final approach leg, the maximum altitude error is encountered. This is because during the turn there is a loss of lift associated with the rotation of the lift vector, and it takes some time for the glide slope controller to catch up with the altitude error. The last plot shows good tracking of altitude, with never more than 10ft of error. The autogyro touches down on the target point with a final position error of about a half a foot.

[This page intentionally left blank.]

Chapter 4

Estimation

The controller designed in Chapter 3 will not have direct access to the states it uses, so an estimator will be developed which will be used to gather state information from sensors. The statistics of the estimator are propagated with inertial sensors. Measurement updates are made by various sensors including GPS, magnetic, range finder, and target pixel position. Images from an onboard camera are sent to the ground station and the target is tracked by an operator. The target pixel offset from the image center is sent back to the vehicle as an additional measurement.

4.1 Extended Kalman Filter

The extended Kalman filter is used for state estimation of nonlinear plants [4]. A continuous-discrete filter was chosen such that there is a continuous time propagation of statistics with discrete measurement updates. The nonlinear dynamics of the system are used for state estimate propagation, and linearized dynamics are used for error covariance propagation and measurement updates.

4.2 Filter Dynamics

First the dynamics that the filter is modeling must be defined. The dynamics will take advantage of the inertial sensor's measurements of acceleration and angular rates.

These measurements are inherently noisy and have an unknown bias. The measurements (denoted by subscript m) are related to the actual values by

$$\begin{aligned}
\mathbf{a} &= \mathbf{a}_m + \boldsymbol{\lambda}_{\mathbf{a}} + \boldsymbol{\nu}_{\mathbf{a}} \\
\boldsymbol{\Omega}_3 &= \boldsymbol{\Omega}_{3_m} + \boldsymbol{\lambda}_{\boldsymbol{\Omega}_3} + \boldsymbol{\nu}_{\boldsymbol{\Omega}_3} \qquad (4.1) \\
\boldsymbol{\Omega}_4 &= \boldsymbol{\Omega}_{4_m} + \boldsymbol{\lambda}_{\boldsymbol{\Omega}_4} + \boldsymbol{\nu}_{\boldsymbol{\Omega}_4}
\end{aligned}$$

where ν is a zero mean Gaussian sequence with standard deviation σ, λ is the constant bias of the sensor, and $\boldsymbol{\Omega}_3$ and $\boldsymbol{\Omega}_4$ were defined in Equations (2.3) and (2.12) as matrices of angular rates. Also, $\mathbf{a}$, the accelerometer measurement, is the vector of rotor and aerodynamic forces divided by mass. During simulations, the standard deviation of the noises above were taken to be $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 0.3$ ft/s^2 and $\sigma_P = \sigma_Q = \sigma_R = 0.01$ rad/s. The biases were modeled as constants due to the short mission time. The values used were $\lambda_{a_x} = \lambda_{a_y} = \lambda_{a_z} = 0.3$ ft/s^2 and $\lambda_P = \lambda_Q = \lambda_R = 0.02$ rad/s.

Using these measurement equations (4.1) in the state equations (2.8), (2.4), and (2.11), the filter dynamics are obtained.

$$\begin{aligned}
{}_p\dot{\mathbf{x}} &= \mathbf{C}\mathbf{v} \\
\dot{\mathbf{v}} &= (\mathbf{a}_m + \boldsymbol{\lambda}_{\mathbf{a}}) + \mathbf{C}^T\mathbf{g} - (\boldsymbol{\Omega}_{3_m} + \boldsymbol{\lambda}_{\boldsymbol{\Omega}_3})\,\mathbf{v} + \mathbf{w}_1 \qquad (4.2) \\
\dot{\mathbf{q}} &= -\frac{1}{2}\,(\boldsymbol{\Omega}_{4_m} + \boldsymbol{\lambda}_{\boldsymbol{\Omega}_4})\,\mathbf{q} + \mathbf{w}_2
\end{aligned}$$

In the above equations the noises have been collected as

$$\mathbf{w}_1 = \boldsymbol{\nu}_{\mathbf{a}} - \boldsymbol{\nu}_{\boldsymbol{\Omega}_3}\mathbf{v}, \qquad \mathbf{w}_2 = -\frac{1}{2}\boldsymbol{\nu}_{\boldsymbol{\Omega}_4}. \qquad (4.3)$$

An additional step is to model the measurement biases. To make a robust estimate, a small amount of process noise must be injected into the dynamics of the bias. This ensures that the estimator gain does not converge to zero as time increases.

$$\begin{aligned}\dot{\boldsymbol{\lambda}}_{P,Q,R} &= \mathbf{w}_3 \\ \dot{\boldsymbol{\lambda}}_{a_x,a_y,a_z} &= \mathbf{w}_4\end{aligned} \tag{4.4}$$

Thus the state of the estimator is

$$\mathbf{x} = \left[{}_p\mathbf{x}^T \ \mathbf{v}^T \ \mathbf{q}^T \ \boldsymbol{\lambda}_{P,Q,R}^T \ \boldsymbol{\lambda}_{a_x,a_y,a_z}^T\right]^T \tag{4.5}$$

and the filter dynamics (4.2), (4.4) can be written in the standard nonlinear filter dynamics model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{w} \tag{4.6}$$

where $\mathbf{w} = \left[0_{1\times 3} \ \mathbf{w}_1^T \ \mathbf{w}_2^T \ \mathbf{w}_3^T \ \mathbf{w}_4^T\right]^T$ is the vector of noise terms with spectral density matrix

$$\mathbf{Q} = E(\mathbf{w}\mathbf{w}^T) \tag{4.7}$$

which has been defined in the appendix in Equation (C.4).

4.3 Time Propagation

To propagate the statistics of the state estimate through time, the filter dynamics are used. For the state estimate, the full nonlinear dynamics are integrated

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}\left(\hat{\mathbf{x}}\right). \tag{4.8}$$

The error covariance, $\mathbf{P}$, is propagated by integration of

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{Q} \tag{4.9}$$

where $\mathbf{F}$ represents linearized filter dynamics about the current state estimate

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}. \tag{4.10}$$

Matrices $\mathbf{F}$ and $\mathbf{Q}$ are calculated in Appendix C.1. A fourth order Runge-Kutta integration scheme was used for propagation of both the state estimate and error covariance during simulations.

4.4 Measurement Updates

Measurement updates in the Extended Kalman Filter are handled the same way as in the linear case, except that the updates use linearized measurement equations. Each discrete measurement is modeled by

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\nu} \tag{4.11}$$

where $\mathbf{h}$ is the nonlinear measurement equation and the uncertainties in the measurement instrument are represented by zero mean normally distributed sequences, ν, with covariance matrix

$$\mathbf{R} = E\left(\boldsymbol{\nu}\boldsymbol{\nu}^T\right). \tag{4.12}$$

It is assumed that the measurement are independent and thus $\mathbf{R}$ is a diagonal matrix of the sensor noise variances.

The Kalman gain is found from

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{4.13}$$

where $\mathbf{H}$ is the measurement equation linearized about the current state estimate

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}. \tag{4.14}$$

This gain is used to update the state estimate by weighting the innovation, which is the difference between the actual measurement and the expected measurement based upon the filter model.

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathbf{K}\left(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})\right) \tag{4.15}$$

Finally, the error covariance matrix is updated by

$$\mathbf{P}^{+} = (\mathbf{I} - \mathbf{KH})\,\mathbf{P}. \tag{4.16}$$

The following are descriptions of the particular discrete sensors modeled, and how their measurement equations are found. The linearized measurement equations for each sensor are derived in Appendix C.2.

4.4.1 GPS

The GPS sensor gives independent measurements of both inertial position and inertial velocity.

$$\mathbf{z}_{GPS} = \mathbf{h}_{GPS}(\mathbf{x}) + \boldsymbol{\nu}_{GPS} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{C} \end{bmatrix} \begin{bmatrix} {}_{p}\mathbf{x} \\ \mathbf{v} \end{bmatrix} + \boldsymbol{\nu}_{GPS} \tag{4.17}$$

The measurement covariance matrix is

$$\mathbf{R}_{GPS} = \operatorname{diag}\left[\begin{array}{cccccc} \sigma^2_{GPS_{pN}} & \sigma^2_{GPS_{pE}} & \sigma^2_{GPS_{pD}} & \sigma^2_{GPS_{\dot{p}N}} & \sigma^2_{GPS_{\dot{p}E}} & \sigma^2_{GPS_{\dot{p}N}} \end{array}\right]. \tag{4.18}$$

For the purposes of simulation, the GPS was sampled every 0.2 s and the standard deviations were assumed to be $\sigma_{GPS_{pN}} = \sigma_{GPS_{pE}} = 50$ ft, $\sigma_{GPS_{pD}} = 75$ ft, and $\sigma_{GPS_{\dot{p}N}} = \sigma_{GPS_{\dot{p}E}} = \sigma_{GPS_{\dot{p}D}} = 10$ ft/s.

4.4.2 Magnetometer

The magnetometer takes measurements of Earth's magnetic field in the vehicle frame. These readings can be compared to the known local magnetic field to find the vehicle attitude by the measurement equation

$$\mathbf{z}_{mag} = \mathbf{h}_{mag}(\mathbf{x}) + \boldsymbol{\nu}_{mag} = \mathbf{C} \begin{bmatrix} b_N \\ b_E \\ b_D \end{bmatrix} + \boldsymbol{\nu}_{mag} \tag{4.19}$$

where $\mathbf{b} = [b_N \; b_E \; b_D]^T$ are the known local magnetic field intensities. The measurement covariance matrix is

$$\mathbf{R}_{mag} = \text{diag}\left[\begin{array}{ccc} \sigma_{b_x}^2 & \sigma_{b_y}^2 & \sigma_{b_z}^2 \end{array}\right] \tag{4.20}$$

During simulations, the magnetometer was sampled every 0.1 s, and the standard deviations were assumed to be $\sigma_{b_x} = \sigma_{b_y} = \sigma_{b_z} = 0.005$ gauss.

4.4.3 Range Finder

The range finder is a sensor that returns the range from the vehicle to the first object its beam intersects with. Sonar or laser types could both be used. Due to its limited range, it will only be activated when the vehicle is close to the ground (for the purposes of simulation, the sensor was activated once the vehicle was under 150ft of altitude). It is assumed that the sensor is attached rigidly to the vehicle pointing down (along the body axis z-direction). Thus the measurement is of altitude rotated into the body frame

$$z_{rf} = h_{rf}(\mathbf{x}) + \boldsymbol{\nu}_{rf} = \frac{{}_pD}{\cos\theta\cos\phi} + \boldsymbol{\nu}_{rf}, \tag{4.21}$$

and the measurement covariance is just a single variance for this one-dimensional input

$$\mathbf{R}_{rf} = \left[\begin{array}{c} \sigma_{rf}^2 \end{array}\right] \tag{4.22}$$

where $\sigma_{rf} = 0.1$ ft, and the sampling rate was 0.05 s.

4.4.4 Pixel Position

The final measurement is the target pixel position. The images captured by the onboard camera are sent to the ground station. The measurement is the target offset from the center of the viewing screen, in pixels, as the target is tracked by the ground station personnel. Note that this measurement will only be available during the final approach phase of the flight, after the last turn has been completed and the target is viewable. This pixel position is an indication of both vehicle position and attitude. The technique of integrating this sensor into the estimator is similar to [1], except that in this case the vehicle's camera is mounted at location $\mathbf{r}^B_{cam}$ with respect to the center of gravity of the vehicle in the body frame, and oriented such that it is aligned with the trim glide angle, $\cot \gamma_t$. Thus when the vehicle is on the proper trajectory the target will be aligned with the center of the camera's view.

First, a vector must be constructed that defines the position of the target with respect to the camera. Refer to Figure 4-1 for a description of the geometry. The vector starts at the camera position, travels the opposite direction of $\mathbf{r}^B_{cam}$ to reach the center of gravity, and then travels the opposite direction of ${}_p\mathbf{x}$ rotated to the body frame to reach the target

$$\mathbf{x}^C_{tar} = \mathbf{C}_{C/B}\left(\mathbf{C}^T\left(-{}_p\mathbf{x}\right) - \mathbf{r}^B_{cam}\right) \tag{4.23}$$

where the superscript C denotes the camera frame and $\mathbf{C}_{C/B}$ is the transformation matrix from the body to camera frame

$$\mathbf{C}_{C/B} = \begin{bmatrix} \cos\gamma_t & 0 & -\sin\gamma_t \\ 0 & 1 & 0 \\ \sin\gamma_t & 0 & \cos\gamma_t \end{bmatrix}. \tag{4.24}$$

The camera has horizontal and vertical viewing angles of β_y and β_z. The image in this field of view is broken up into a grid of pixels: N_y horizontally and N_z vertically (N_y and N_z define the camera resolution). The tangent of the viewing half angle is the ratio of half the horizontal image size, $\frac{N_y}{2}$, to the perpendicular component of the position vector. Then, considering Figure 4-2, the position of the pixel, as a fraction of the total image size, is

$$\tau_y = \frac{1}{\tan\frac{\beta_y}{2}} \times \frac{(\mathbf{x}^C_{tar})_y}{(\mathbf{x}^C_{tar})_x}. \tag{4.25}$$

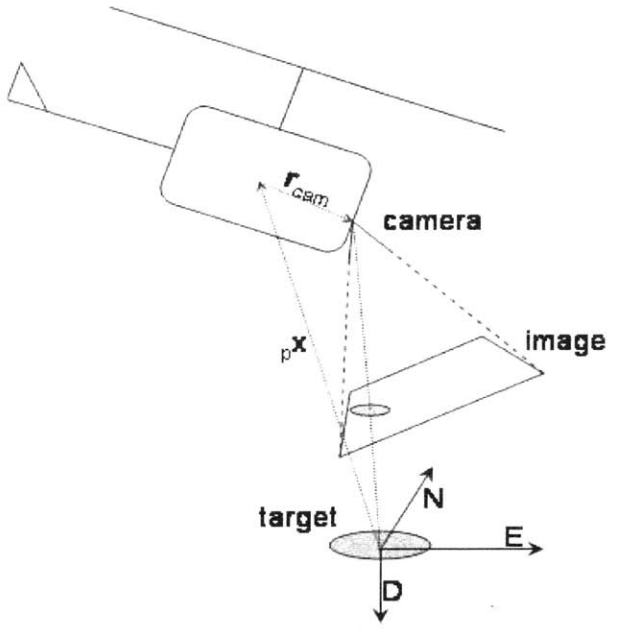

Figure 4-1: Geometry of camera showing vector relations

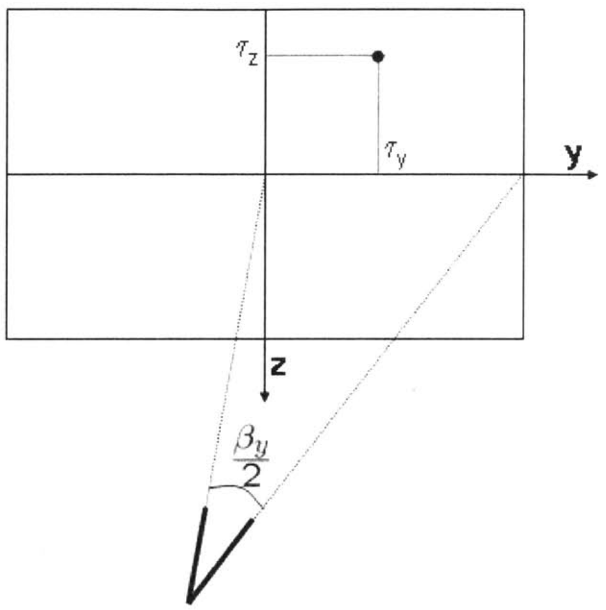

Figure 4-2: Camera image details

The actual pixel location is the product of this ratio and half the horizontal resolution

$$\text{pixel}_y = \tau_y \times \frac{N_y}{2}. \tag{4.26}$$

A similar relation is found for the pixel location in the z direction. Thus the final measurement equations are

$$\begin{aligned}
\mathbf{z}_{img} &= \begin{bmatrix} \text{pixel}_y \\ \text{pixel}_z \end{bmatrix} = \mathbf{h}_{img}(\mathbf{x}) + \boldsymbol{\nu}_{img} \\
\text{pixel}_y &= \frac{(\mathbf{x}^C_{tar})_y}{(\mathbf{x}^C_{tar})_x} \frac{1}{\tan\frac{\beta_y}{2}} \frac{N_y}{2} + \nu_{\text{pixel}_y} \\
\text{pixel}_z &= -\frac{(\mathbf{x}^C_{tar})_z}{(\mathbf{x}^C_{tar})_x} \frac{1}{\tan\frac{\beta_z}{2}} \frac{N_z}{2} + \nu_{\text{pixel}_z}.
\end{aligned} \tag{4.27}$$

with

$$\mathbf{R}_{img} = \text{diag}\left[\begin{array}{cc} \sigma^2_{pixel_x} & \sigma^2_{pixel_y} \end{array}\right] \tag{4.28}$$

where $\sigma_{pixel_x} = \sigma_{pixel_y} = 3$ pixels, and the target pixel position was measured every 0.1 s.

4.5 Estimator Performance

In order to find some indication of performance of the estimator, a baseline simulation was run. The initial condition of the autogyro was 3000 ft away from the target and pointed straight at it with 1875 ft of altitude, corresponding to the trim glide slope. The controller developed in Chapter 3 was used with full state knowledge to maintain the proper trajectory to hit the target, however essentially no inputs were required due to the initial condition. The estimator was run during the simulation with an initial condition offset of the true state by 16.5 feet for positions, 1 ft/s for velocities, and 0.01 for quaternions. The estimator output is shown in Figure 4-3. In these plots, the lighter lines are the errors: the difference between the true and estimated states. The darker lines are an indication of the standard deviation of the error, and are plus and minus the square root of the diagonal elements of the error covariance matrix P.

Consider the first two plots showing position and velocity estimation error. In the position plot notice how the standard deviation lines for altitude and lateral deviation decrease steadily. This is because as the vehicle approaches the target, the camera information becomes more reliable. Furthermore, note the sharp decrease in the standard deviation of altitude and vertical velocity error near the very end of the simulation. This is due to the activation of the range finder sensor. From the last two plots of the bias error, observe that injecting the process noise into the filter dynamics ensures steady convergence to zero error, but keeps the standard deviations relatively high.

	$_pN$	$_pE$	$_pD$	U	V	W
Full	4.1ft	4.1m	4.8ft	0.6ft/s	1.0ft/s	1.0ft/s
Last 10s	2.5ftm	1.0ft	2.5ft	0.7ft/s	0.7ft/s	0.6ft/s

Table 4.1: Root mean square values of estimation error during the entire and last ten seconds of the basaline estimator simulation

The root mean square values of the estimation error during this simulation are shown in Table 4.1. Although the errors over the course of the entire simulation are relatively high, the moments of the simulation closest to the landing phase are the

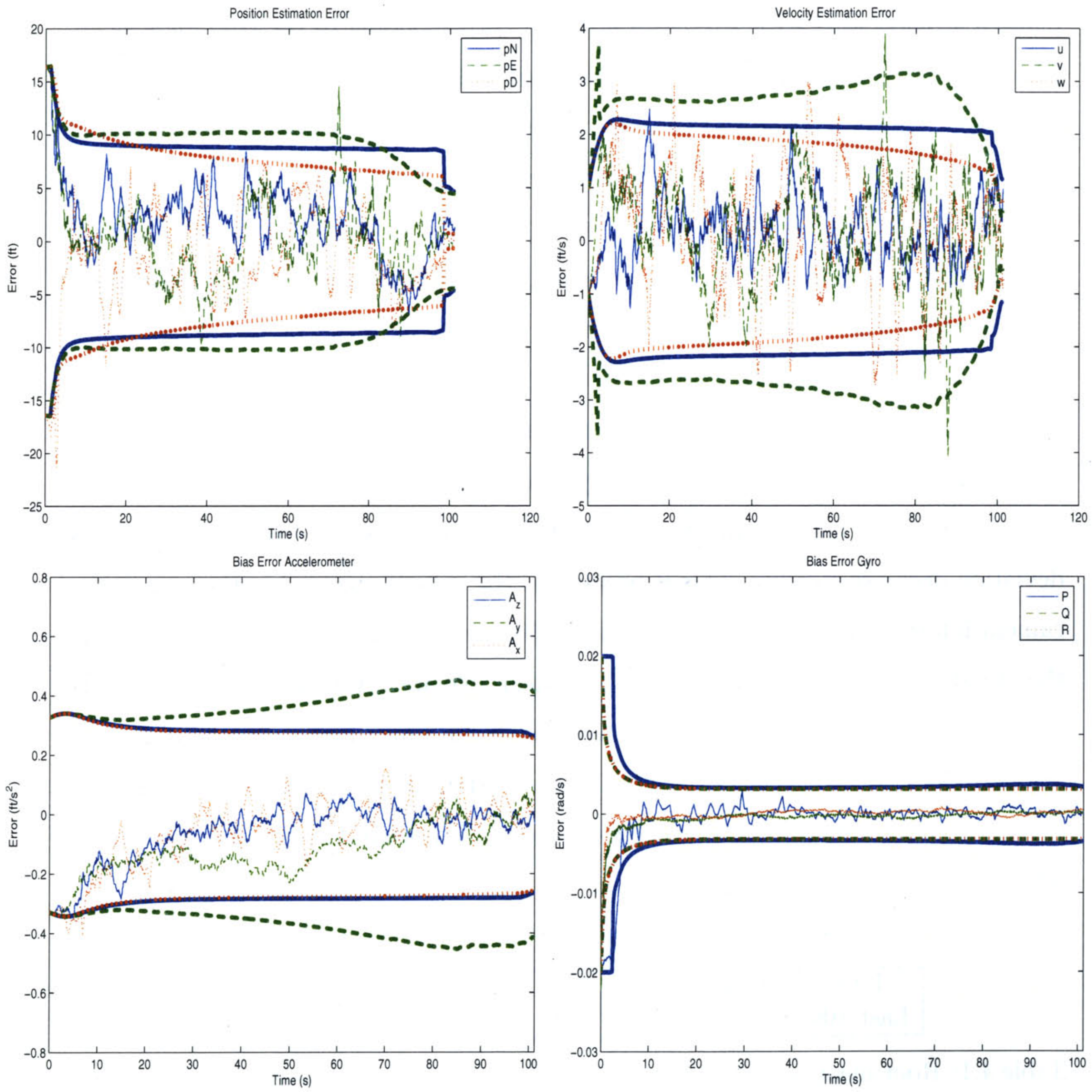

Figure 4-3: Results of baseline estimator simulation

most important. The errors during the last ten seconds are small enough to reach the required accuracy.

[This page intentionally left blank.]

Chapter 5

Results

The complete model including controller and estimator was tested using random parameters in a Monte-Carlo simulation. The achieved accuracy statistics are shown here.

5.1 Parameter Setup

In order to test the simulation under a variety of conditions, different parameters were given normal distributions, and then re-initialized with a different random variable for each simulation run. The parameters considered included initial position and heading, steady wind speed, and turbulence level. Additionally, the wind disturbance was independently sampled from its distribution for different simulations. Table 5.1 shows the variances and means of each normally distributed parameter. The location of the initial position means that the vehicle will generally be traveling from east to west during its final approach phase, with a flight path similar to that of Figure 3-19.

	U_{w_i}	V_{w_i}	ν_{wind}	$_pN$	$_pE$	$_pD$	ψ
mean	-0.5 ft/s	5 ft/s	0	-2000 ft	200 ft	-3000 ft	0
variance	0.1 ft/s	0.1 ft/s	1%	100 ft	100 ft	100 ft	10°

Table 5.1: Variances of the normally distributed parameters for simulation trials

5.2 Accuracy

The resulting final position errors of several simulation runs are plotted in Figure 5-1. The mean position error magnitude in the ${}_pN$ direction, roughly corresponding to cross-track error, was 2.1 ft. The mean in the ${}_pE$ direction, or flight-track error, was 3.6 ft.

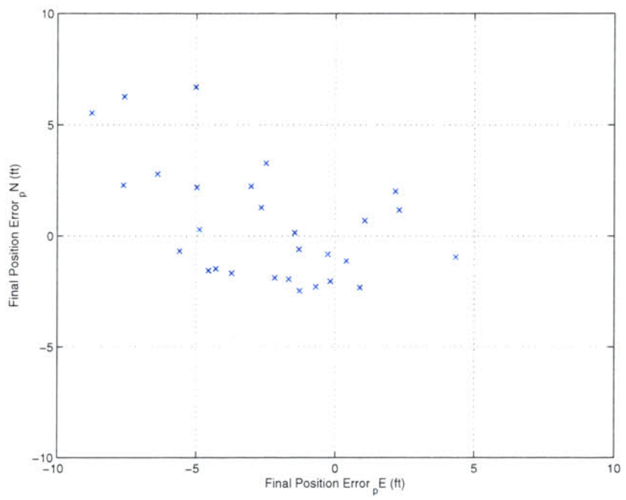

Figure 5-1: Final absolute position error of several simulation runs

Notice that the distribution of landing errors has a slight skewing in the ${}_pN$ and $-{}_pE$ directions. The path planner and controller are slightly overcompensating for the wind disturbance, leading to overshooting the target.

Chapter 6

Conclusion

The use of a gliding autogyro as a delivery vehicle in precision airdrop missions has been shown to be a good option in fulfilling modern accuracy requirements. Several developments in this work have shown this feasibility.

- Simulation: A six degree of freedom simulation was derived based on rigid body dynamics and rotor aerodynamics including blade element and momentum theory. Although simulations with higher fidelity are possible, Sections 2.4 and 3.2 showed the validity of the simulation in both a trim and dynamic sense by comparison with other work.

- Controller: A controller was designed to guide the autogyro to the target location. This controller utilized a classical, multiple loop closure technique, with a new nonlinear guidance law for path following. A path generating algorithm was also created that utilizes known constant wind speed to maintain a constant nominal glide slope throughout the mission.

- Estimation: The extended Kalman filter with continuous propagation and discrete measurement updates was used to estimate the state information needed by the controller.

Simulation results show an impressive accuracy of about 5 ft, even in the face of large wind disturbances.

This project is still in development, and several steps must be taken to ensure success of the final design of the autogyro system.

- Path Generation: The assumption of a constant wind profile with altitude is not very realistic. Assuming a constant rate of descent, the path planning algorithm can be modified to use the average wind speed as an input. However, a wind estimation and path regeneration scheme must be created to include the effects of time-varying wind.

- Sensor Suite Selection: The particular sensors to be used must be selected to tune the parameters of the estimator. Additionally, the relative importance of each sensor must be determined.

- Landing: Even though final position accuracy is good, the rate of descent may be too high for different payloads and applications. This could be achieved by an appropriate flaring strategy, either by backwards tilting of the rotor and trading forward energy for vertical energy, or by adding a variable collective control that would allow trading rotor rotation rate for thrust increase. In each case, an energy necessary for flight is used to provide a momentary decrease in descent speed. A challenge would be to time the landing for a minimum descent speed while maintaining position accuracy.

- Startup and Transition: An autorotating rotor in equilibrium will maintain its rotation rate, but starting the rotor can be an issue. The problem is essentially breaking a stall barrier on the rotor blades. Also, the transition from vertical descent to forward glide will pose additional challenges due to the vehicle nonlinearities between these two trim conditions.

Appendix A

Solutions for Thrust and Induced Velocity

The equations for the thrust and induced velocity produced by the rotor are

$$T = \frac{1}{4}\left(W_b - V_i\right)\Omega r^2 \rho abc \tag{A.1}$$

$$V_i = \frac{\eta_{GE} T}{2\rho\pi r^2 \sqrt{U_a^2 + V_a^2 + \left(W_r - V_i\right)^2}} \tag{A.2}$$

Note that the above quantities are defined in terms of each other and require special attention. An iterative scheme was suggested in [8], but it was found that for autorotation in close to vertical descent that the scheme was not appropriate.

A.1 Iterative Solution

In order to investigate the fallacies of the iterative scheme, the above equations can be rewritten by substituting T from equation (A.1) into V_i from equation (A.2). By defining $A_i = \frac{8\pi}{\eta_{GE}\Omega abc}$, the resulting equation reduces to

$$V_i^4 - 2W_r V_i^3 + \left(U_a^2 + V_a^2 + W_r^2 - A_i^{-2}\right) V_i^2 + 2\frac{W_b}{A_i^2} V_i - V_b^2 = 0 \tag{A.3}$$

which is a quartic in V_i.

In forward flight, the four roots of the equation are arranged such that one is negative, two are imaginary, and the fourth is the only positive answer. In this case an iterative scheme converges quickly and there are no problems. When approaching vertical descent, however, the two imaginary roots become real and positive, so that the iterative scheme has multiple possible convergence points. This ambiguous solution results in large jumps in induced velocity during a simulation as the scheme switches between the solutions it is tracking. This behavior can be seen in figure A-1, where the roots of equation (A.3) are plotted for varying trim conditions with increasing forward speed. The roots start at the x locations, which correspond to a high forward speed. The imaginary roots converge onto the real line and split at sufficiently low forward speeds (close to vertical descent).

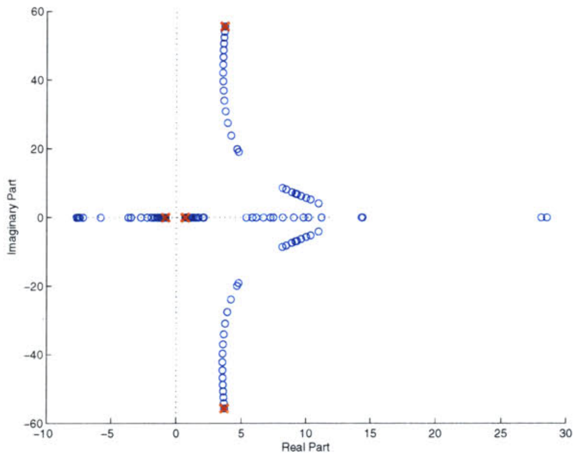

Figure A-1: Locus of roots of the induced velocity equation ((A.3)) with varying forward flight speeds

Another method of solution for the induced velocity and thrust is to solve equation (A.3) directly. An analytical solution of the quartic polynomial problem was utilized to solve this problem, which unlike the iterative scheme, finds all roots of the polynomial. Since we know that the vehicle's performance in vertical descent is being underestimated (namely the descent speed is too great), the smallest value of induced velocity can be chosen so that thrust is the greatest, which most closely matches experimental data (see Figure 2-1). A description of this method follows.

A.2 Quartic Solution

In order to solve equation (A.3) for induced velocity, the quartic equation will be solved analytically.

The general equation

$$z_i^4 + a_3 z_i^3 + a_2 z_i^2 + a_1 z_i + a_0 = 0 \tag{A.4}$$

has a resolvent cubic

$$x_i^3 - a_2 x_i^2 + (a_1 a_3 - 4a_0)\, y_i + \left(4a_2 a_0 - a_1^2 - a_3^2 a_0\right) = 0 \tag{A.5}$$

which must first be solved using the cubic formula as follows.

The general form of the cubic equation,

$$x_i^3 + b_2 x_i^2 + b_1 x_i + b_0 = 0, \tag{A.6}$$

can be written as

$$x_i^3 + P_i x_i = Q_i$$

with

$$Q_i = \frac{b_2^2 - 3b_1}{9}$$

$$R_i = \frac{2b_2^3 - 9b_1 b_2 + 27b_0}{54}$$

With these definitions the roots of the equations based upon the relative sizes of R_i and Q_i can be found [15]. A more thorough derivation can be found in [12].

If $R_i^2 < Q_i^3$, then define

$$\theta_i = arccos\left(\frac{R_i}{\sqrt{Q_i^3}}\right)$$

$$y_1 = -\sqrt{Q_i} cos\left(\frac{\theta_i}{3}\right) - \frac{b_2}{3}$$

If $R_i^2 > Q_i^3$, then define

$$A_i = -sign(R_i)\sqrt[3]{|R_i| + \sqrt{R_i^2 - Q_i^3}}$$

$$B_i = Q_i/A_i$$

$$y_1 = (A_i + B_i) - \frac{b_2}{3}$$

This root can be used to find the solution of the original quartic equation (A.4) [19]. First define

$$S_i = \sqrt{\frac{1}{4}a_3^2 - a_3 + y_1}$$

Then define two other parameters based upon the value of S.

If $S_i = 0$,

$$D_i = \sqrt{\frac{3}{4}a_3^2 - 2a_2 + 2\sqrt{y_1^2 - 4a_0}}$$

$$E_i = \sqrt{\frac{3}{4}a_3^2 - 2a_2 - 2\sqrt{y_1^2 - 4a_0}}$$

Otherwise,

$$D_i = \sqrt{\frac{\frac{3}{4}a_3^2 - S^2 - 2a_2 + \frac{1}{4}\left(4a_3a_2 - 8a_1 - a_3^3\right)}{S_i}}$$

$$E_i = \sqrt{\frac{\frac{3}{4}a_3^2 - S^2 - 2a_2 - \frac{1}{4}\left(4a_3a_2 - 8a_1 - a_3^3\right)}{S_i}}$$

Finally, the roots of the original quartic polynomial, (A.4), are

$$z_1 = -\frac{1}{4}a_3 + \frac{1}{2}S_i + \frac{1}{2}D_i \qquad z_2 = -\frac{1}{4}a_3 + \frac{1}{2}S_i - \frac{1}{2}D_i$$

$$z_3 = -\frac{1}{4}a_3 - \frac{1}{2}S_i + \frac{1}{2}E_i \qquad z_4 = -\frac{1}{4}a_3 - \frac{1}{2}S_i - \frac{1}{2}E_i$$

Appendix B

Simulation Details

This appendix shows the details of the simulation not covered in the main text. Included are simulation parameters, trim point, and the linear system.

B.1 Parameters

Table B.1 details the specific parameters used when constructing the simulation developed in Chapter 2 in Matlab's Simulink environment.

Parameter	Symbol	Value	Unit	Parameter	Symbol	Value	Unit
mass	m	10	lb	blade chord	c	1.73	in
roll inertia	I_{xx}	0.1	slug-ft^2	rotor radius	r	21.25	in
pitch inertia	I_{yy}	0.2	slug-ft^2	C_L slope	a	5.79	-
yaw inertia	I_{zz}	0.2	slug-ft^2	zero-lift C_D	C_{D_0}	0.01	-
flapping inertia	I_b	0.04	slug-ft^2	hub location	x_r	0	ft
# of blades	b	3	-	hub location	z_r	-1	ft
axial fus C_D	$X_{uu,fus}$	-0.5	ft^2	vt location	x_{vt}	-3.23	ft
lateral fus C_D	$Y_{vv,fus}$	-1.3	ft^2	vt location	z_{vt}	0.04	ft
vertical fus C_D	$Z_{vv,fus}$	-1.9	ft^2	ht location	x_{ht}	-2	ft
fus cp dist	z_{fus}	0	ft	ht location	z_{ht}	-0.12	ft
vt C_L from sideslip	$Y_{uv,vt}$	-0.66	ft^2	vt trim C_L	$Y_{uu,vt}$	0	ft^2
vt C_L from sidewash	$Y_{vv,vt}$	-0.17	ft^2	ht trim C_L	$Z_{uu,ht}$	0	ft^2
ht C_L due to α	$Z_{uw,ht}$	-0.48	ft^2	vt $C_{L,max}$	$Y_{VV,vt,max}$	0.17	ft^2
ht C_L from downwash	$Z_{ww,ht}$	-0.12	ft^2	ht $C_{L,max}$	$Z_{VV,ht,max}$	0.12	ft^2

Table B.1: Simulation parameters

B.2 Trim Point

Table B.2 shows the trim point used to linearize the simulation and create a linear model.

δ_{f_t}	δ_{s_t}	ϕ_t	θ_t	ψ_t	U_t	V_t	W_t	Ω_t	a_{1_t}	b_{1_t}
3.5	-0.09	0	-5.75	0	32.5	0	15.5	1080	0.77	0
deg	deg	deg	deg	deg	ft/s	ft/s	ft/s	rpm	deg	deg

Table B.2: Simulation trim point

B.3 Linear Model

The nonlinear model defined in Chapter 2 was linearized by the Matlab tool `linmod`. The tool uses small perturbation analysis to generate a linear model of the system, based upon a trim point, of the form

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{u} \tag{B.1}$$

where the state $\mathbf{x}$ and input $\mathbf{u}$ are defined as

$$\tilde{\mathbf{x}} = [\phi \; \theta \; \psi \; U \; V \; W \; P \; Q \; R \; {}_pN \; {}_pE \; {}_pD \; \Omega \; a_1 \; b_1]^T \tag{B.2}$$

$$\mathbf{u} = [\delta_f \; \delta_s]^T . \tag{B.3}$$

The matrices that define the system using the trim point shown in Table B.2 were found to be

$$\mathbf{B} = \left[\begin{array}{c|ccccccccccccccc} & \phi & \theta & \psi & U & V & W & P & Q & R & {}_pN & {}_pE & {}_pD & \Omega & a_1 & b_1 \\ \hline \delta_f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.06 & 0 \\ \delta_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.06 \end{array}\right]^T \tag{B.4}$$

$$\mathbf{A} = \left. \begin{array}{c|ccccccccccccccc}
 & \phi & \theta & \psi & U & V & W & P & Q & R & {}_pN & {}_pE & {}_pD & \Omega & a_1 & b_1 \\
\hline
\phi & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\psi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.01 & 0 & 0 & 0 & 0 & 0 & 0 \\
U & 0 & -32.01 & 0 & -0.14 & 0 & -0.03 & 0 & -15.47 & 0 & 0 & 0 & 0 & -0.01 & -68.06 & 4.82 \\
V & 32.01 & 0 & 0 & 0 & -0.08 & 0 & 15.47 & 0 & -32.30 & 0 & 0 & 0 & 0 & 0 & 29.67 \\
W & 0 & 3.22 & 0 & -0.26 & 0 & -1.57 & 0 & 32.53 & 0 & 0 & 0 & 0 & -0.27 & -44.65 & 0.07 \\
P & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & -0.03 & 0 & 0 & 0 & 0 & 0 & 92.21 \\
Q & 0 & 0 & 0 & -0.07 & 0 & 0.01 & 0 & -0.09 & 0 & 0 & 0 & 0 & 0.02 & 105.76 & -7.50 \\
R & 0 & 0 & 0 & 0 & 0.41 & 0 & 0 & 0 & -1.33 & 0 & 0 & 0 & 0 & 0 & 0 \\
{}_pN & 0 & 18.65 & 0 & 1 & 0 & -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
{}_pE & -15.47 & 0 & 30.85 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
{}_pD & 0 & -30.85 & 0 & 0.1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\Omega & 0 & 0 & 0 & 0.1 & 0 & 0.84 & 0 & 0 & 0 & 0 & 0 & 0 & -0.14 & 27.20 & 0 \\
a_1 & 0 & 0 & 0 & 0.01 & 0 & 0.01 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -3.1 & 0 \\
b_1 & 0 & 0 & 0 & 0 & -0.01 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -0.01 & -3.46
\end{array} \right] \tag{B.5}$$

[This page intentionally left blank.]

Appendix C

Estimator Definition

The specifics of the extended Kalman filter are derived here, including the linearized dynamics and measurement equations.

C.1 Error Covariance Propagation

First, recall the dynamics equations with state $\mathbf{x}^T = \left[{}_p\mathbf{x}^T\ \mathbf{v}^T\ \mathbf{q}^T\ \lambda_{P,Q,R}^T\ \lambda_{a_x,a_y,a_z}^T\right]$ from Equations (4.2) and (4.4) and repeated here.

$$\begin{aligned}
{}_p\dot{\mathbf{x}} &= \mathbf{C}\mathbf{v} = \mathbf{f}_1(\mathbf{x}) \\
\dot{\mathbf{v}} &= (\mathbf{a}_m + \lambda_{\mathbf{a}}) + \mathbf{C}^T\mathbf{g} - (\mathbf{\Omega}_{3_m} + \lambda_{\mathbf{\Omega}_3})\,\mathbf{v} + \mathbf{w}_1 = \mathbf{f}_2(\mathbf{x}) + \mathbf{w}_1 \\
\dot{\mathbf{q}} &= -\frac{1}{2}(\mathbf{\Omega}_{4_m} + \lambda_{\mathbf{\Omega}_4})\,\mathbf{q} + \mathbf{w}_2 = \mathbf{f}_3(\mathbf{x}) + \mathbf{w}_3 \\
\dot{\lambda}_{P,Q,R} &= \mathbf{w}_3 \\
\dot{\lambda}_{a_x,a_y,a_z} &= \mathbf{w}_4
\end{aligned} \tag{C.1}$$

Partial derivatives must be taken to construct the linearized dynamics equations, $\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big|_{\mathbf{x}=\hat{\mathbf{x}}}$. This dynamics matrix is used to propagate the error covariance matrix through time. The derivatives are taken such that each state occupies a row, and the partial derivative with respect to a state occupies a column.

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{C} & \frac{\partial \mathbf{Cv}}{\partial \mathbf{q}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{\Omega}_{3_m} - \boldsymbol{\lambda}_{\Omega_3} & \frac{\partial \mathbf{C}^T\mathbf{g}}{\partial \mathbf{q}} & -\frac{\partial \boldsymbol{\lambda}_{\Omega_3}\mathbf{v}}{\partial \lambda_{P,Q,R}} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & -\frac{1}{2}\left(\mathbf{\Omega}_{4_m} + \boldsymbol{\lambda}_{\Omega_4}\right) & -\frac{1}{2}\frac{\partial \boldsymbol{\lambda}_{\Omega_4}}{\partial \boldsymbol{\lambda}_{P,Q,R}} & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{C.2}$$

The transformation matrix partial derivatives are

$$\frac{\partial \mathbf{Cv}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_0U - q_3V + q_2W & q_1U + q_2V + q_3W & -q_2U + q_1V + q_0W & -q_3U - Q_0V + q_1W \\ q_3U + q_0V - q_1W & q_2U - q_1V - q_0W & q_1U + q_2V + q_3W & q_0U - q_3V + q_2W \\ -q_2U + q_1V + q_0W & q_3U + q_0V - q_1W & -q_0U + q_3V - q_2W & q_1U + q_2V + q_3W \end{bmatrix} \tag{C.3}$$

$$\frac{\partial \mathbf{C}^T\mathbf{g}}{\partial \mathbf{q}} = 2\mathrm{g}\begin{bmatrix} -q_2 & q_3 & -q_0 & q_1 \\ q_1 & q_0 & q_3 & q_2 \\ q_0 & -q_1 & -q_2 & q_3 \end{bmatrix}$$

The bias partial derivatives are

$$\frac{\partial \boldsymbol{\lambda}_{\Omega_3}\mathbf{v}}{\partial \boldsymbol{\lambda}_{P,Q,R}} = \begin{bmatrix} 0 & W & -V \\ -W & 0 & U \\ V & -U & 0 \end{bmatrix} \qquad \frac{\partial \boldsymbol{\lambda}_{\Omega_4}}{\partial \boldsymbol{\lambda}_{P,Q,R}} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{\lambda}_{\Omega_4}}{\partial \boldsymbol{\lambda}_{P,Q,R}} = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

The second matrix needed for error covariance propagation is the process noise matrix $\mathbf{Q} = E(\mathbf{ww}^T)$ where $\mathbf{w}$ is the concatenation of the process noises from the dynamics equations (C.1).

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & E[\mathbf{w}_1\mathbf{w}_1^T] & E[\mathbf{w}_1\mathbf{w}_2^T] & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} & E[\mathbf{w}_2\mathbf{w}_1^T] & E[\mathbf{w}_2\mathbf{w}_2^T] & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & E[\mathbf{w}_3\mathbf{w}_3^T] & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & E[\mathbf{w}_4\mathbf{w}_4^T] \end{bmatrix} \tag{C.4}$$

The internal expectations can be calculated assuming no correlation between ν_{a_x}, ν_{a_y}, ν_{a_z}, ν_P, ν_Q, and ν_R.

$$\mathbf{w}_1 = \boldsymbol{\nu}_\mathbf{a} - \boldsymbol{\nu}_{\boldsymbol{\Omega}_3}\mathbf{v} = \begin{bmatrix} \nu_{a_x} + \nu_R V - \nu_Q W \\ \nu_{a_y} - \nu_R U + \nu_P W \\ \nu_{a_z} + \nu_Q U + \nu_P V \end{bmatrix}$$

$$E[\mathbf{w}_1\mathbf{w}_1^T] = \begin{bmatrix} \sigma_{a_x}^2 + \sigma_R^2 V^2 + \sigma_Q^2 W^2 & -\sigma_R^2 VU & -\sigma_Q^2 WU \\ -\sigma_R^2 VU & \sigma_{a_y}^2 + \sigma_R^2 U^2 + \sigma_P^2 W^2 & -\sigma_P^2 WV \\ -\sigma_Q^2 WU & -\sigma_P^2 WV & \sigma_{a_z}^2 + \sigma_Q^2 U^2 + \sigma_P^2 V^2 \end{bmatrix}$$

$$\mathbf{w}_2 = -\frac{1}{2}\boldsymbol{\nu}_{\boldsymbol{\Omega}_4}\mathbf{q} = -\frac{1}{2}\begin{bmatrix} \nu_P q_1 + \nu_Q q_2 + \nu_R q_3 \\ -\nu_P q_0 - \nu_R q_2 + \nu_Q q_3 \\ -\nu_Q q_0 + \nu_R q_1 - \nu_P q_3 \\ -\nu_P q_0 - \nu_Q q_1 + \nu_P q_2 \end{bmatrix}$$

$$E[\mathbf{w}_2\mathbf{w}_2] = \frac{1}{4}\begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{12} & e_{22} & e_{23} & e_{24} \\ e_{13} & e_{23} & e_{33} & e_{34} \\ e_{14} & e_{24} & e_{34} & e_{44} \end{bmatrix}$$

$$e_{11} = \sigma_P^2 q_1^2 + \sigma_Q^2 q_2^2 + \sigma_R^2 q_3^2 \qquad e_{22} = \sigma_P^2 q_0^2 + \sigma_R^2 q_2^2 + \sigma_Q^2 q_3^2$$

$$e_{33} = \sigma_Q^2 q_0^2 + \sigma_R^2 q_1^2 + \sigma_P^2 q_3^2 \qquad e_{44} = \sigma_R^2 q_0^2 + \sigma_Q^2 q_1^2 + \sigma_P^2 q_2^2$$

$$e_{12} = -\sigma_P^2 q_1 q_0 + \sigma_Q^2 q_2 q_3 - \sigma_R^2 q_3 q_2 \qquad e_{13} = -\sigma_P^2 q_1 q_3 - \sigma_Q^2 q_0 q_2 + \sigma_R^2 q_1 q_3$$

$$e_{14} = \sigma_P^2 q_1 q_2 - \sigma_Q^2 q_2 q_1 - \sigma_R^2 q_3 q_0 \qquad e_{23} = \sigma_P^2 q_0 q_3 - \sigma_Q^2 q_3 q_0 - \sigma_R^2 q_2 q_1$$

$$e_{24} = -\sigma_P^2 q_0 q_2 - \sigma_Q q_3 q_1 + \sigma_R^2 q_2 q_0 \qquad e_{34} = -\sigma_P^2 q_3 q_2 + \sigma_Q^2 q_0 q_1 - \sigma_R^2 q_1 q_0$$

$$E[\mathbf{w}_2\mathbf{w}_1^T] = -\frac{1}{2}\begin{bmatrix} -\sigma_Q^2 q_2 W + \sigma_R^2 q_3 V & -\sigma_R^2 q_3 U + \sigma_P^2 q_1 W & -\sigma_P^2 q_1 V + \sigma_Q^2 q_2 U \\ -\sigma_Q^2 q_3 W - \sigma_R^2 q_2 V & \sigma_R^2 q_2 U - \sigma_P^2 q_0 W & \sigma_P^2 q_0 V + \sigma_Q^2 q_3 U \\ \sigma_Q^2 q_0 W + \sigma_R^2 q_1 V & -\sigma_R^2 q_1 U - \sigma_P^2 q_3 W & \sigma_P^2 q_3 V - \sigma_Q^2 q_0 U \\ \sigma_Q^2 q_1 W - \sigma_R^2 q_0 V & \sigma_R^2 q_0 U + \sigma_P^2 q_2 W & -\sigma_P^2 q_2 V - \sigma_Q^2 q_1 U \end{bmatrix}$$

$$E[\mathbf{w}_3\mathbf{w}_3^T] = \begin{bmatrix} \sigma_{\lambda_P}^2 & 0 & 0 \\ 0 & \sigma_{\lambda_Q}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_R}^2 \end{bmatrix} \qquad E[\mathbf{w}_4\mathbf{w}_4^T] = \begin{bmatrix} \sigma_{\lambda_{a_x}}^2 & 0 & 0 \\ 0 & \sigma_{\lambda_{a_y}}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_{a_z}}^2 \end{bmatrix}$$

In the above equations, $\sigma_{\lambda_P} = \sigma_{\lambda_Q} = \sigma_{\lambda_R} = 0.001$ rad/s, and $\sigma_{\lambda_{a_x}} = \sigma_{\lambda_{a_y}} = \sigma_{\lambda_{a_z}} = 0.02$ m/s^2.

C.2 Linearized Measurement Equations

The measurement equations must be linearized to use the discrete measurement updates in Equations (4.13) and (4.16).

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \qquad \text{(C.5)}$$

C.2.1 GPS

Recall the GPS measurement equation

$$\mathbf{h}_{GPS}(\mathbf{x}) = \left[{}_p\mathbf{x}^T \; \mathbf{C}\mathbf{V}^T\right]^T.$$

The linearized measurement equation is

$$\mathbf{H}_{GPS} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{C} & \frac{\partial \mathbf{Cv}}{\partial \mathbf{q}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{C.6}$$

where $\frac{\partial \mathbf{Cv}}{\partial \mathbf{q}}$ was defined in (C.3).

C.2.2 Magnetometer

Recall the magnetometer measurement equation

$$\mathbf{h}_{mag} = \mathbf{C}\left[b_N \; b_E \; b_D\right]^T = \mathbf{Cb}.$$

Solving for the linearized equation yields

$$\mathbf{H}_{mag} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{Cb}}{\partial \mathbf{q}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{C.7}$$

where $\frac{\partial \mathbf{Cb}}{\partial \mathbf{q}}$ is found from (C.3), replacing $\mathbf{v}$ with $\mathbf{b}$.

C.2.3 Range Finder

The range finder measurement equation is

$$h_{rf} = \frac{{}_pD}{\cos\theta\cos\phi}.$$

The linearized form is

$$\mathbf{H}_{rf} = \begin{bmatrix} \frac{\partial h_{rf}}{\partial_p \mathbf{x}} & \mathbf{0}_{1\times3} & \frac{\partial h_{rf}}{\mathbf{q}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{bmatrix} \tag{C.8}$$

where the internal partial derivatives follow. The partial with respect to position is simply

$$\frac{\partial h_{rf}}{\partial_p \mathbf{x}} = \begin{bmatrix} 0 & 0 & {}_pD \end{bmatrix}.$$

Using the chain rule for the partial derivative with respect to the quaternions gives four other partial derivatives which are easier to solve.

$$\frac{\partial h_{rf}}{\partial \mathbf{q}} = \frac{\partial h_{rf}}{\partial \theta}\frac{\partial \theta}{\partial \mathbf{q}} + \frac{\partial h_{rf}}{\partial \phi}\frac{\partial \phi}{\partial \mathbf{q}}$$

$$\frac{\partial h_{rf}}{\partial \theta} = \frac{{}_pD \sin\theta}{\cos\phi \cos^2\theta} \qquad \frac{\partial h_{rf}}{\partial \phi} = \frac{{}_pD \sin\phi}{\cos\theta \cos^2\phi}$$

$$\frac{\partial \theta}{\partial \mathbf{q}} = -\frac{\frac{\partial C_{31}}{\partial \mathbf{q}}}{\sqrt{1 - C_{31}^2}} \qquad \frac{\partial \phi}{\partial \mathbf{q}} = \frac{\frac{\partial C_{32}}{\partial \mathbf{q}} C_{33} - C_{32}\frac{\partial C_{33}}{\partial \mathbf{q}}}{C_{33}^2 + C_{32}^2}$$

$$\frac{\partial C_{31}}{\partial \mathbf{q}} = 2\begin{bmatrix} -q_2 & q_3 & -q_0 & q_1 \end{bmatrix} \quad \frac{\partial C_{32}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_1 & q_0 & q_3 & q_2 \end{bmatrix} \quad \frac{\partial C_{33}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_0 & -q_1 & -q_2 & q_3 \end{bmatrix}$$

C.2.4 Camera Image

The camera pixel location measurement equation is

$$\mathbf{h}_{img} = \begin{bmatrix} \frac{(\mathbf{x}_{tar}^C)_y}{(\mathbf{x}_{tar}^C)_x} \frac{1}{\tan\frac{\beta_y}{2}} \frac{N_y}{2} \\ -\frac{(\mathbf{x}_{tar}^C)_z}{(\mathbf{x}_{tar}^C)_x} \frac{1}{\tan\frac{\beta_z}{2}} \frac{N_z}{2} \end{bmatrix} = \begin{bmatrix} \frac{(\mathbf{x}_{tar}^C)_y}{(\mathbf{x}_{tar}^C)_x} l_y \\ -\frac{(\mathbf{x}_{tar}^C)_z}{(\mathbf{x}_{tar}^C)_x} l_z \end{bmatrix} = \begin{bmatrix} h_{img1} \\ h_{img2} \end{bmatrix}$$

which has the linear form

$$\mathbf{H}_{img} = \begin{bmatrix} \frac{\partial \mathbf{h}_{img}}{\partial_p \mathbf{x}} & \mathbf{0}_{6\times3} & \frac{\partial \mathbf{h}_{img}}{\partial \mathbf{q}} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} \end{bmatrix}.$$

The partial derivatives with respect to position are

$$\frac{\partial h_{img_1}}{\partial_p \mathbf{x}} = \frac{\frac{\partial ({}_p\mathbf{x}^C_{tar})_y}{\partial_p \mathbf{x}} ({}_p\mathbf{x}^C_{tar})_x - ({}_p\mathbf{x}^C_{tar})_y \frac{\partial ({}_p\mathbf{x}^C_{tar})_x}{\partial_p \mathbf{x}}}{({}_p\mathbf{x}^C_{tar})^2_x} l_y$$

$$\frac{\partial h_{img_2}}{\partial_p \mathbf{x}} = -\frac{\frac{\partial ({}_p\mathbf{x}^C_{tar})_z}{\partial_p \mathbf{x}} ({}_p\mathbf{x}^C_{tar})_x - ({}_p\mathbf{x}^C_{tar})_z \frac{\partial ({}_p\mathbf{x}^C_{tar})_x}{\partial_p \mathbf{x}}}{({}_p\mathbf{x}^C_{tar})^2_x} l_z$$

where

$$\frac{\partial_p \mathbf{x}^C_{tar}}{\partial_p \mathbf{x}} = \frac{\partial}{\partial_p \mathbf{x}} \begin{bmatrix} ({}_p\mathbf{x}^C_{tar})_x & ({}_p\mathbf{x}^C_{tar})_y & ({}_p\mathbf{x}^C_{tar})_z \end{bmatrix}^T = \frac{\partial}{\partial_p \mathbf{x}} \left(-\mathbf{C}_{C/B}\mathbf{C}_p^T \mathbf{x}\right) = -\left(\mathbf{C}_{C/B}\mathbf{C}^T\right)$$

$$= \begin{bmatrix} C_{11}\cos\gamma_0 - C_{31}\sin\gamma_0 & C_{12}\cos\gamma_0 - C_{32}\sin\gamma_0 & C_{13}\cos\gamma_0 - C_{33}\sin\gamma_0 \\ C_{21} & C_{22} & C_{23} \\ C_{11}\sin\gamma_0 + C_{31}\cos\gamma_0 & C_{12}\sin\gamma_0 + C_{32}\cos\gamma_0 & C_{13}\sin\gamma_0 + C_{33}\cos\gamma_0 \end{bmatrix}.$$

The partial derivatives with respect to attitude are

$$\frac{\partial h_{img_1}}{\partial \mathbf{q}} = \frac{\frac{\partial ({}_p\mathbf{x}^C_{tar})_y}{\partial \mathbf{q}} ({}_p\mathbf{x}^C_{tar})_x - ({}_p\mathbf{x}^C_{tar})_y \frac{\partial ({}_p\mathbf{x}^C_{tar})_x}{\partial \mathbf{q}}}{({}_p\mathbf{x}^C_{tar})^2_x} l_y$$

$$\frac{\partial h_{img_2}}{\partial \mathbf{q}} = -\frac{\frac{\partial ({}_p\mathbf{x}^C_{tar})_z}{\partial \mathbf{q}} ({}_p\mathbf{x}^C_{tar})_x - ({}_p\mathbf{x}^C_{tar})_z \frac{\partial ({}_p\mathbf{x}^C_{tar})_x}{\partial \mathbf{q}}}{({}_p\mathbf{x}^C_{tar})^2_x} l_z$$

where

$$\frac{\partial_p \mathbf{x}^C_{tar}}{\partial_\mathbf{q}} = -\frac{\partial}{\partial \mathbf{q}} \left(\mathbf{C}_{C/B}\mathbf{C}_p^T \mathbf{x}\right) = -\mathbf{C}_{C/B}\frac{\partial}{\partial \mathbf{q}}\mathbf{C}_p^T \mathbf{x}$$

$$= -\frac{\partial}{\partial \mathbf{q}} \begin{bmatrix} {}_p x(C_{11}\cos\gamma_0 - C_{13}\sin\gamma_0) +_p y(C_{21}\cos\gamma_0 - C_{23}\sin\gamma_0) +_p z(C_{31}\cos\gamma_0 - C_{33}\sin\gamma_0) \\ {}_p x C_{12} +_p y C_{22} +_p z C_{32} \\ {}_p x(C_{11}\sin\gamma_0 + C_{13}\cos\gamma_0) +_p y(C_{21}\sin\gamma_0 + C_{23}\cos\gamma_0) +_p z(C_{31}\sin\gamma_0 + C_{33}\cos\gamma_0) \end{bmatrix}$$

and

$$\frac{\partial C_{11}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_0 & q_1 & -q_2 & -q_3 \end{bmatrix} \quad \frac{\partial C_{12}}{\partial \mathbf{q}} = 2\begin{bmatrix} -q_3 & q_2 & q_1 & -q_0 \end{bmatrix} \quad \frac{\partial C_{13}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_2 & q_3 & q_0 & q_1 \end{bmatrix}$$

$$\frac{\partial C_{21}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_3 & q_2 & q_1 & q_0 \end{bmatrix} \quad \frac{\partial C_{22}}{\partial \mathbf{q}} = 2\begin{bmatrix} q_0 & -q_1 & q_2 & -q_3 \end{bmatrix} \quad \frac{\partial C_{23}}{\partial \mathbf{q}} = 2\begin{bmatrix} -q_1 & -q_0 & q_3 & q_2 \end{bmatrix}.$$

Bibliography

[1] Hyungil Ahn. Vision-based estimation and control of airdrop vehicles for aerial deployment of sensor networks. Master's thesis, Massachusetts Institute of Technology, June 2004.

[2] F.J. Bailey. A simplified theoretical method of determining the characteristics of a lifting rotor in forward flight. *National Advisory Committe for Aeronautics Report*, (716), 1941.

[3] Bernard Etkin and Lloyd Duff Reid. *Dynamics of Flight, Stability and Control.* John Wiley and Sons, Inc., third edition, 1996.

[4] Arthur Gelb. *Applied optimal estimation.* MIT Press, 1974.

[5] Robert K. Heffley and Marc A. Mnich. Minimum-complexity helicopter simulation math model. NASA Contractor Report 177476, NASA, April 1988.

[6] S. S. Houston. Identification of autogyro longitudinal stability and control characteristics. *Journal of Guidance, Control, and Dynamics*, 21(3):391–399, 1998.

[7] S. S. Houston. Rotor-wake modeling for simulation of helicopter flight mechanics in autorotation. *Journal of Aircraft*, 40(5):938–945, 2003.

[8] Eric N. Johnson and Paul A. DeBitetto. Modeling and simulation for small autonomous helicopter development. *AIAA Modeling and Simulation Technologies Conference*, New Orleans, LA, Aug. 11-13, 1997, Collection of Technical Papers (A97-37186 10-01).

[9] Wayne Johnson. *Helicopter Theory.* Princeton University Press, 1980.

[10] J. Gordon Leishman. *Principles of Helicopter Aerodynamics.* Cambridge University Press, 2000.

[11] C. A. Lopez and V. L. Wells. Dynamics and stability of an autorotating rotor/wing unmanned aircraft. *Journal of Guidance, Control, and Dynamics*, 27(2):258–270, 2004.

[12] S. Neumark. *Solution of Cubic and Quartic Equations.* Pergamon Press, 1965.

[13] Michael Oliver. A parametric analysis of the start-up procedure and flight characteristics of a gliding autogyro. Master's thesis, Massachusetts Institute of Technology, January 2005.

[14] Sanghyuk Park, John Deyst, and Jonathan P. How. A new nonlinear guidance logic for trajectory tracking. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, Aug. 16-19, 2004.

[15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brain P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press, second edition, 2002.

[16] Raymond W. Prouty. *Helicopter Performance, Stability, and Control.* Robert E. Krieger Publishing Company, 1990.

[17] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation.* John Wiley and Sons, Inc., second edition, 2003.

[18] Damian Toohey. Development of a small parafoil vehicle for precision delivery. Master's thesis, Massachusetts Institute of Technology, December 2003.

[19] Eric W. Weisstein. Quartic equation. *Mathworld*, 1999.