Development of a Small Parafoil Vehicle for Precision Delivery

by

Damian Toohey

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 .....
Department of Aeronautics and Astronautics
February 17, 2005

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

Accepted by .....
Jaime Peraire
Professor of Aeronautics and Astronautics
Chairman, Committee of Graduate Studies

Development of a Small Parafoil Vehicle for Precision Delivery

by

Damian Toohey

Submitted to the Department of Aeronautics and Astronautics on February 17, 2005, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

Abstract

The goal of the MIT Draper Partnership Program is to develop a system capable of deploying a cluster of ISR (intelligence, surveillance, reconnaissance) sensors over an area of interest. One of the proposed methods is a guided parafoil vehicle that is deployed from a UAV. After unfurling its canopy, it is envisioned that the vehicle would follow a planned trajectory to the target, and vision-based targeting system would be employed to provide the necessary accuracy.

A 6DOF parafoil simulation was created in the Matlab/Simulink environment to study the parafoil dynamics and assess the feasibility of the delivery method. The parafoil's range and controllability were tested under a variety of simulated wind conditions, and a hypothetical mission was conceived to evaluate the performance of the proposed trajectory designs and guidance laws. Also, an extended Kalman filter was incorporated into the simulation to determine effectiveness of the camera-based targeting system and other navigation sensors.

This paper details the development of the simulation as well as its results.

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

Table of Contents

List of Figures

List of Tables

List of Variables

Variables introduced in Chapter 2

α, $\alpha_{L=0}$: angle of attack, angle of attack at which $C_L = 0$
β = side-slip angle
δ_f, δ_a = symmetric, asymmetric flap deflection
Γ= anhedral angle
γ = glide slope angle
ω = angular velocity vector
ρ = air density
Θ_r = rigging angle
a: lift curve slope
AR: aspect ratio
$A_{0,R}$, $A_{0,A}$ = real and apparent mass contributions to 6x6 inertia matrix
a_{po} = distance from pitch center to origin
$a_{r,o}$ = distance from roll center to origin
a_{pc} = distance from the pitch center to the confluence point
a_{rc} = distance from the roll center to the confluence point
b = parafoil span
c = chord length
C_D= parafoil overall drag coefficient
$C_{D,p}$ $C_{D,\delta f}$ = drag stability derivatives
C_M $C_{M,q}$ $C_{M,C/4}$= pitching moment coefficients
C_L: parafoil lift coefficient
C_Y $C_{Y,\beta}$ $C_{Y,\delta a}$ = side force contribution due to side-slip angle
C_l $C_{l,p}$ $C_{l,\delta a}$ $C_{l,r}$ $C_{l,\beta}$ = roll moment stability derivatives
C_n $C_{N,\delta a}$ $C_{n,\delta a}$ $C_{n,p}$ $C_{n,r}$ $C_{n,\beta}$ = yaw moment stability derivatives
h_0 = angular momentum
I_{11} I_{22} I_{33} = rotational inertia about roll, pitch, yaw centers
I_{f11} = apparent mass constant
$J_{o,R}$ = Real rotational matrix about 0
$J_{o,ii}$ = apparent rotational inertia terms
k: small aspect ratio correction factor
k_1: nonlinear low aspect ratio correction
L = overall lift
M_R = real mass
m_{ii} = apparent mass terms
m_{f22} = apparent mass constant
p, q, r = roll, pitch, yaw rate
P = linear momentum
R_{CG} = distance between the system CG and the canopy CG
R_{SG} = distance between the payload CG and the system CG
S = parafoil canopy surface area
v_0 = velocity vector at point 0
W = parafoil system weight

Variables introduced in Chapter 3

Ψ = vehicle heading
$Drift_N$, $Drift_E$ = difference between predicted and actual position
h_i , h_f = initial vehicle height
p_N, p_E, p_D = vehicle position in NED components
p_{Ntar}, p_{Etar}, p_{Dtar} = target position in NED components
p_{Ntar_w}, p_{Etar_w}, p_{Dtar_w} = wind shifted target position in NED components
$shift_N$, $shift_E$ = integral of wind velocity used to shift the aimpoint from NED reference frame to the wind-relative coordinate system
$Traj_N$, $Traj_E$, $Traj_Z$ = trajectory function in North, East, Down directions
t_{traj} = time before vehicle arrives at approach point
t_{2D} = trajectory time
V_{hor} , V_Z = horizontal, vertical velocity
V_G = ground speed
$wind_N$, $wind_E$ = north and east wind components

Variables introduced in Chapter 4

$\Psi_{Command}$ = heading command
$\Delta\Psi$ = heading error
ε = tracking angle
η = angle between velocity vector and aim point
A_C = centripetal acceleration
K = control gain
L = distance to aim point
R = radius of instantaneous circular segment

Variables introduced in Chapter 5

λ = angle-to-target
$pixel_x$ = pixel offset from target
U,V,W = velocity in body axis
q_0, q_1, q_2, q_3 = quaternions
$bias_p$, $bias_q$, $bias_r$ = rate gyro biases
$bias_{ax}$, $bias_{ay}$, $bias_{az}$ = accelerometer biases
C_{B2E} = direction cosine, body to earth
H(x) = measurement matrix
b_x, b_y, b_z = magnetometer measurements in body axis
b_N, b_E, b_D = magnetometer measurements in NED directions
p_{Ngps}, p_{Egps}, pD_{GPS}, v_{Ngps}, v_{Egps}, v_{Dgps} = GPS position and velocity measurements
x_{camera}, y_{camera}, z_{camera} = camera position relative to IMU

Acknowledgements

I would like to thank Professor Deyst for giving me the opportunity to work on this project. His guidance and support helped make this project an amazing learning experience for me.

I am especially grateful to Draper Laboratory for sponsoring my project. Without its backing, this project would not have been possible. And I very much appreciate the support from our Draper advisors, Sean George and Brent Appleby. Their assistance was invaluable to my work.

My fellow MDPP team members were wonderful. Not only did I learn a lot from each of them, they also made my time at MIT truly enjoyable. I wish them all the best of luck in their future endeavors.

And a special thanks to the Unified Engineering community. Being a TA was perhaps my favorite experience at MIT, and I am so grateful to Professor Harris for giving me the chance to work with so many incredible teachers and students.

1. Introduction

Great emphasis is placed on timely intelligence and information gathering by modern military practices. Technological advances have allowed for the development of increasingly sophisticated observation satellites and unmanned aircraft, however, this renders the military to be increasingly reliant and limited to observation from a distance. There is no doubt that the imagery from the latest satellite or unmanned aircraft platforms is of the highest quality, but no matter the resolution, these ultra hi-tech systems are regularly defeated by low-tech measures, such as trees and caves.

As was made clearly evident in recent conflicts in Bosnia and Afghanistan, there are serious limitations to intelligence gathering from the sky. Satellite orbits are regular and predictable, and even a Predator deployed in a moment's notice has trouble locating objects of interest beneath camouflage. Caves or dense foliage consistently defeat even the best satellite technology, proving that up-close, on-the-ground sensing is still an essential piece of intelligence gathering.

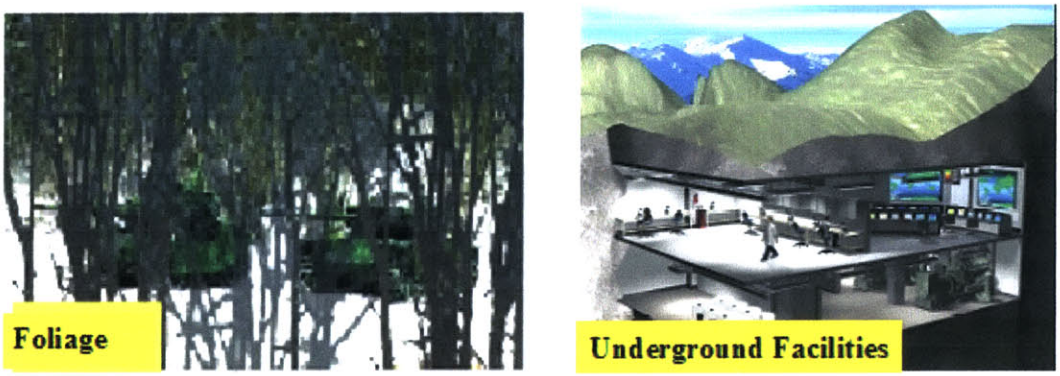


Figure 1-1 – Methods of avoiding detection

Although the advantages of carefully placed ground sensors with means of relaying information are clearly apparent, there are serious barriers to successful implementation. For example, the risks to personnel who place the sensor may be too great to make it practical. Other impediments include the delays in getting personnel in place and the logistical issues of supporting and extracting them. UAV systems have proven capability in delivering a variety of payloads and may provide a solution to this problem as well.

1.2 - MDPP Project Objectives

The goal of the MIT Draper Partnership Program is to utilize existing UAV technologies to develop a system capable of deploying a cluster of ISR (intelligence, surveillance, reconnaissance) sensors over an area of interest. A large, end-to-end demonstration is not the purpose of the project. Rather, the intent is to focus on the various system components with the hope of identifying key concepts and enabling technologies.

1.3 Precision Delivery

To establish successful sensor networks from standoff distances, certain critical sensor nodes may be required to be placed at very specific locations, for example on rooftops or tree canopies. Therefore, the ability to deliver the sensors with high precision and reliability is critical. In addition, a working network may consist of hundreds of sensors, and it is therefore important to minimize cost and complexity, as well as maintain compatibility with today's UAV deployment capabilities.

It is assumed for the purposes of this study that the Predator system will be available as the carrier UAV, so the payload delivery vehicle (PDV) was designed with the Predator's capabilities in mind. In order to take advantage of the Predator's existing flare dispenser-like deployment system, the PDV will be a cylindrical body.

Table 1 – 1 PDV concepts that were considered

	Advantages	**Disadvantages**
Ballistic	Simplicity Packaging	Small glide slope High impact deceleration
Parafoil	Glide Slope Maneuverability Slow descent rate	Long flight duration Susceptibility to wind
Rotorcraft	Slow descent rate	Cost and complexity
Fixed wing	Maneuverability Long range	Cost and complexity

Various delivery vehicle types were initially considered, however the ballistic and parafoil configurations were eventually chosen as the most practical for development. Table 1-1 lists the major advantages and disadvantages of each concept. Mechanically, the ballistic and parafoil configurations are both relatively simple to design while also quite capable of precision delivery. A fixed wing system would have decent maneuverability, but higher speeds would make landing difficult. A rotorcraft may have a slow descent rate, but its complexities would likely drive up the costs and make success less likely. While this paper focuses on the details of the parafoil system, a brief description of the ballistic configuration is given.

The primary advantage of using a guided ballistic drop is its simplicity. With a camera on the descending vehicle to remove target position errors, very accurate placements are

possible without the need for GPS. The 1.5 kg vehicle is cylindrical in shape in order to accommodate deployment from flare dispenser tubes. Similar systems have already been utilized by the Predator UAV. The terminal velocity of the vehicle would be approximately 70 m/s, with control fins providing limited glide slope control of about 20 deg from vertical.

A typical mission may call for the ballistic delivery vehicle to be dropped from a UAV at 10,000 ft. After only a few seconds, the target is acquired by the camera system. Using its movable control fins, small corrections are available to guide the vehicle to its target. A few seconds before impact, an arrestor chute is deployed to soften the landing.

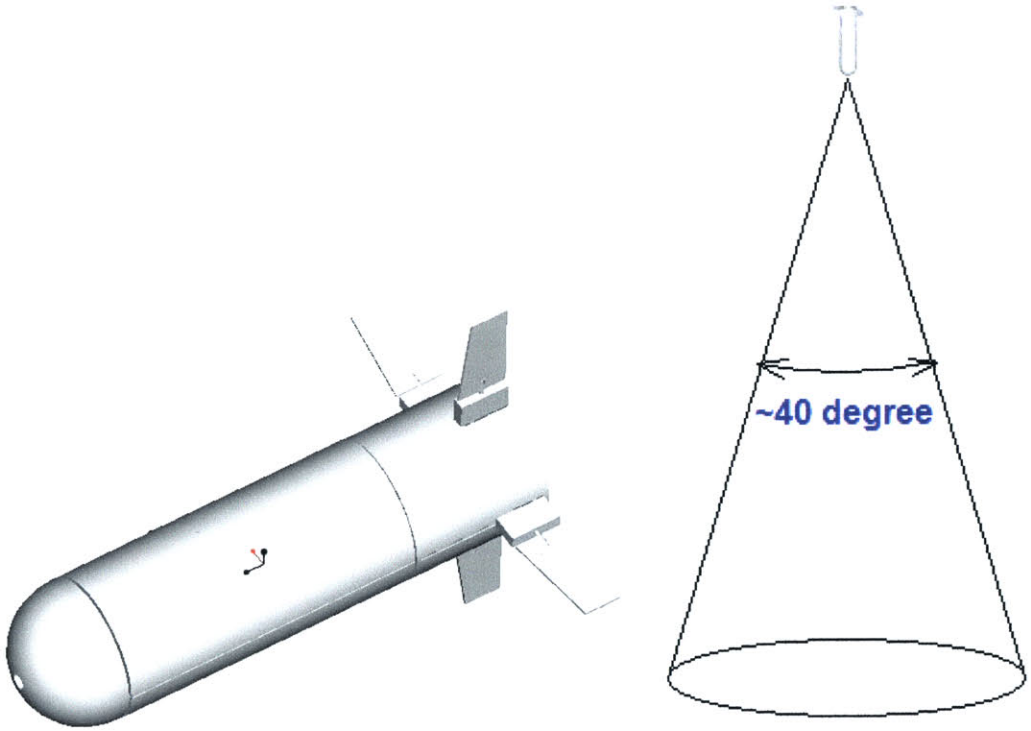


Figure 1-2 – Ballistic configuration

1.4 - Parafoil Method

Parafoil vehicles have already proven to be useful in many situations. Their stable configuration suits autonomous guidance systems, and a low descent rate makes them an attractive choice to safely deliver payload to the ground. Without the need to deploy an arrestor chute, the parafoil can glide all the way to the target without concern about its altitude. Maneuvering is normally achieved by deflecting flaps or trailing edges of the canopy asymmetrically, while some degree of glide slope control is offered with symmetric flap deflections. Dropped from a sufficient altitude, a parafoil-controlled vehicle is capable of reaching a much larger area than the ballistic PDV.

These attributes offer a great deal of flexibility to the precision delivery system. The payload can be dropped further away from the target, which may be desirable for missions over hostile areas. In addition, the longer flight duration would allow for the landing zone to be surveyed during descent. One might imagine a mission where a parafoil PDV is used to examine an area of interest and identify attractive landing zones for future drops.

The relative simplicity of the parafoil system allows for a great deal of compatibility with the ballistic system. It is envisioned that the rear section housing the arrestor chute for the ballistic PDV could be replaced with a module containing the parafoil and the necessary actuators. All other PDV components, such as the on-board computer and avionics, would remain the same.

1.4.1 - Overview of Delivery Method

The parafoil delivery concept is not a new one – there are such systems in use today. However their main function is to deliver cargo to remote areas where an accuracy of about 50 yards is acceptable. Little more than GPS is needed to guide the drop vehicle to the landing zone. For the purposes of our system development, GPS alone cannot offer the required 5m landing accuracy. Under normal circumstances, absolute position can only be estimated to within 10-15m. For a gliding parafoil system, the landing errors are further exacerbated since a 15m error in altitude corresponds to a 45m horizontal error, assuming a glide ratio of 3:1. The proposed solution is to use a vision system to eliminate the targeting errors associated with GPS. The vision system is what sets this project apart from most other guided parafoils.

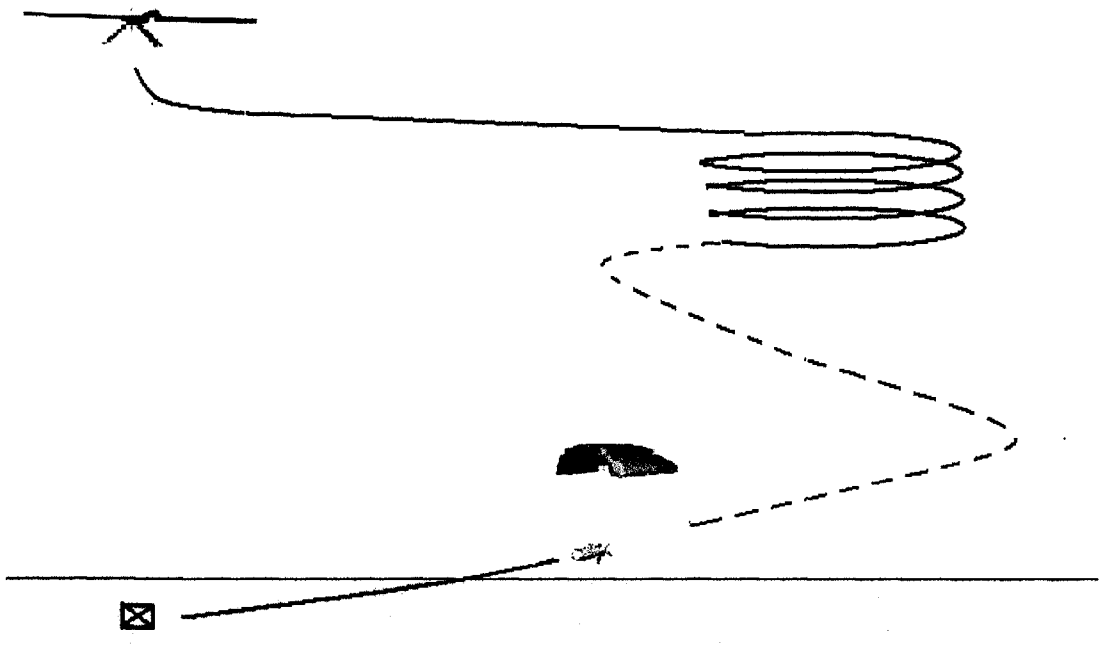


Figure 1-3 – Parafoil delivery concept

A typical mission would have the delivery vehicle deployed from the Predator UAV, falling ballistically until it reaches a lower altitude. At that point, the vehicle unfurls its canopy and begins to follow a planned trajectory that places it in front of the target heading into the wind. A vision system then takes over to provide the necessary accuracy to guide the vehicle to the target.

1.4.2 - Challenges

The benefits of using parafoil delivery do not come without a price. Complications arise due to the slow descent rate and long flight duration. With an airspeed of only 5-7 m/s, wind conditions greatly influence vehicle's groundspeed. Large wind disturbances would play havoc with the light parafoil structure, possibly making the desired accuracy unattainable in gusty conditions.

Whereas the ballistic vehicle is simply dropped above the target, the long flight duration of the parafoil vehicle requires that a more involved trajectory be planned in advance. Wind conditions would have to be measured before the drop and may even need to be monitored during the descent.

The challenges presented by the parafoil delivery system are worth overcoming. The capabilities that the parafoil configuration brings to the precision delivery system nicely compliment the ballistic drop. This project hopes to address the key issues surrounding the parafoil system so that it can become a viable option for the PDV in the future.

2. Parafoil Dynamics

2.1 - Introduction

A 6 DOF dynamical model of the parafoil was created to assist in the development of the precision delivery project. It is a critical component of the parafoil simulation and is essential to the creation of the control and guidance systems. This chapter describes the development of the aerodynamic model, along with its limitations.

Figure 2-1 – Small parafoil from *Airfoil Aviation, Inc.*

Parafoils carry aspects of both airfoils and parachutes. Whereas parachutes only produce drag, a parafoil also generates lift, allowing for useful gliding performance. High-end parafoils can achieve glide ratios of about 3:1.

Most canopies are ram-air chutes – their leading edges are open to allow for inflation as they move forward. The chutes are designed with individually inflatable cells to help maintain structural rigidity, and to add stability during flight. Suspension lines are rigged to give the canopy an anhedral shape. Maneuvering is then achieved by deflecting the

outer trailing edges of the canopy - asymmetric deflections for turns, symmetric deflections for flaring and velocity control.

Rigid body dynamics are applied to this model, therefore the flexible behavior of the parafoil and the relative motion between the canopy and payload are ignored. This limits the fidelity of the simulation, but since the intent is to model behaviors near steady gliding conditions, this approximation of the dynamics should be negligible.

2.2 - Longitudinal Dynamics

We first examine the longitudinal dynamics of the parafoil. The lift curve slope, based on lifting line theory, is given as

$$a = \frac{\pi a_0 AR}{\pi AR + a_0(1+\tau)}$$

This equation is a decent estimation for wings with an aspect ratio above 5, but in our case with the parafoil, aspect ratios are approximately 2.5-3. A correction has been suggested by [10]:

$$a_0' = a_0 k$$

$$\text{where } k = \frac{\pi AR}{a_0} \tanh\left(\frac{a_0}{\pi AR}\right)$$

The lift coefficient in terms of angle of attack can be written as

$$C_L = a(\alpha - \alpha_{L=0}) + k_1 \sin^2(\alpha - \alpha_{L=0})\cos(\alpha - \alpha_{L=0}) \qquad (2\text{-}1)$$

A non-linear component of lift is present in addition to the lift curve slope. This behavior has been observed in previous experiments, and it is believed to be due to drag effects

acting on the normal velocity component of the flow. Based on experimental data for low aspect ratio wings, the constant, k_1, can be approximated from the equation:

$$k_1 = 3.33 - 1.33AR$$

Equation 2.1 has been shown to accurately represent the lift coefficient for low aspect ratio wings, but they do not capture the effects of a parafoil's anhedral shape. A correction can be made using the equation below. Γ is the anhedral angle as described in figure 2-2.

$$C_L = C_{L\Gamma=0} \cos^2 \Gamma$$

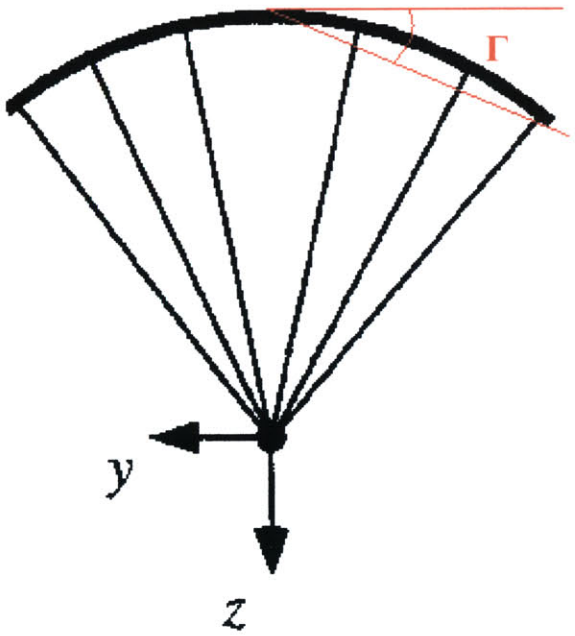


Figure 2-2 – Parafoil anhedral shape

We now examine the drag coefficient for ram air parachutes. Going back to lifting line theory,

$$C_D = C_{D_0} + \frac{C_L^2}{\pi AR}(1+\delta)$$

Similar to the non-linear behavior observed with the lift coefficient, there exists an additional component of drag. Including this component, the drag equation becomes

$$C_D = C_{D_0} + \frac{C_L^2}{\pi AR}(1+\delta) + k_1 \sin^3(\alpha - \alpha_{L=0}) \qquad (2\text{-}2)$$

k_1 is identical to the term used in the equation for the lift coefficient. The parasite drag, C_{D0}, can be estimated by adding up the various contributions of drag in the system.

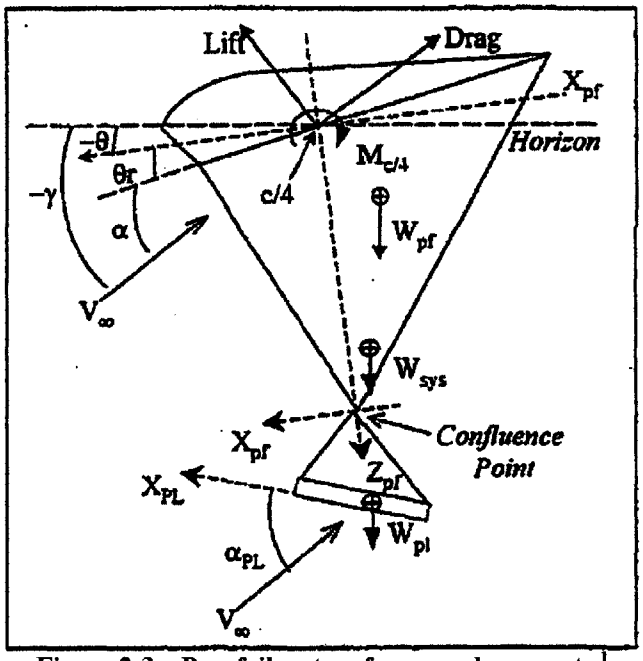


Figure 2-3 – Parafoil system forces and moments [1]

The equation for the pitching moment about the CG of the parafoil-payload system is,

$$C_m = C_{m,c/4} + \frac{R_{cg}}{c}\left[C_D \cos(\alpha+\theta_r) - C_L \sin(\alpha+\theta_r)\right] - \frac{R_{sg}}{c} C_{D,p} \cos(\alpha+\theta_r) \qquad (2\text{-}3)$$

$C_{m,c/4}$ -- pitching moment of the canopy about the ¼ chord point
$C_{D,p}$ – drag caused by the payload
R_{cg} -- distance between the system CG and the canopy CG
R_{sg} -- distance between the payload CG and the system CG
θ_r -- rigging angle

The equilibrium incidence angle (where the pitching moment is zero) is a function of the rigging angle. In terms of gliding performance, there exists an optimal angle of attack at

[1] Diagram from [3]

which to operate. Therefore the rigging angle is usually chosen to put the equilibrium point at that angle.

The static stability relations developed thus far provide an understanding of the parafoil's trim performance. Given that all the forces and moments are balanced during a steady glide, the trim velocity can be obtained from,

$$V = \left(\frac{2}{\rho} \cdot \frac{W}{S} \cdot \frac{\cos\gamma}{C_L} \right)^{1/2} \tag{2-4}$$

where the glide slope angle, γ, is related to the lift to drag ratio by

$$\gamma = \arctan\left(\frac{1}{L/D} \right)$$

2.3 - Trailing Edge Deflection

Control lines run to the outer trailing edges and sides of the parafoil. When pulled, the lift and drag are both increased. The wind tunnel tests conducted at Georgia Tech [6] provide the basis for the estimation of the effect control deflections would have on our vehicle. Flap deflections have less influence on the L/D ratio than might be expected. Any increase in lift due to flaps is accompanied by an increase in drag. Using the wind tunnel data on the small parafoil at Georgia Tech, the C_L values corresponding to the full flap deflection range are between 0.54 and 0.8, and the values for C_D range from 0.22 to 0.38. This corresponds to a change in L/D from zero to full flaps of only 14%.

2.4 - Dynamic Stability

One last component of the longitudinal dynamics needs to be discussed. As the vehicle pitches downward, the local increase of velocity on the parafoil causes an increase in

drag. This produces a negative pitching moment that serves to damp out the pitching rotation.

To estimate the value of the pitch damping term, we approximate the change in drag due to a perturbation in the local velocity:

$$D = (\bar{q}S)C_D = \frac{1}{2}\rho(V_{Trim} + \Delta V)^2 S \cdot C_D$$

Splitting the drag up, we get,

$$D = D_{Trim} + \Delta D = \frac{1}{2}\rho V_{Trim}{}^2 S \cdot C_D + \frac{1}{2}\rho(2V_{Trim}\Delta V + \Delta V^2)S \cdot C_D$$

ΔV is equal to the pitch rate, q, multiplied by the distance from the system CG to the parafoil, R_{CG}. Keeping only the dominant terms, we arrive at

$$\Delta D \cong \frac{1}{2}\rho(2V_{Trim}q \cdot R_{CG})S \cdot C_D$$

It is then useful to write the change in drag in terms of an increase in the drag coefficient. Doing so, then solving for ΔC_D gives us

$$\Delta D = \frac{1}{2}\rho V_{Trim}{}^2 S \cdot \Delta C_D \qquad \Rightarrow \Delta C_D = \frac{2q \cdot R_{CG}}{V_{Trim}} C_{Dtrim}$$

Now we can look at the change in C_M due to the change in C_D:

$$\Delta C_M = -\frac{2R_{CG}{}^2 q}{\bar{c}V_{Trim}}\cos(\alpha + \theta_r)\Delta C_D$$

And finally, we convert to a dimensionless form so we can write

$$\Delta C_M = C_{M,q}\frac{qc}{2V_{Trim}} \quad \text{where } C_{M,q} \cong 4\left(\frac{R_{CG}}{c}\right)^2 \cos(\alpha + \theta_r)\cdot C_{Dtrim} \tag{2-5}$$

Equations 2-6, 2-7, 2-8 are a summary of the parafoil longitudinal dynamics. The last terms of the lift and drag equations represent the flap deflection contributions, both the symmetric, δ_f, and asymmetric, δ_a.

$$C_L = \left\{a(\alpha - \alpha_{L=0}) + k_1 \sin^2(\alpha - \alpha_{L=0})\cos(\alpha - \alpha_{L=0})\right\}\cos^2\Gamma + C_{L,\delta f}(2\delta_f + |\delta a|) \quad (2\text{-}6)$$

$$C_D = C_{D_0} + \frac{C_L^2}{\pi AR}(1+\delta) + k_1 \sin^3(\alpha - \alpha_{L=0}) + C_{D,\delta f}(2\delta_f + |\delta a|) \quad (2\text{-}7)$$

$$C_m = C_{m,c/4} + \frac{R_{cg}}{c}\left[C_D \cos(\alpha + \theta_r) - C_L \sin(\alpha + \theta_r)\right] - \frac{R_{sg}}{c} C_{D_p} \cos(\alpha + \theta_r) - C_{Mq}\frac{q\bar{c}}{2V_T} \quad (2\text{-}8)$$

Table 2-1 Longitudinal Parameters

a	2.556
$\alpha_{L=0}$	-.0782
k_1	0.1984
Γ	10 deg
$C_{L,\delta f}$	0.13
$C_{D,\delta f}$	0.08
C_{D0}	0.15
$C_{m,c/4}$	-0.1
$C_{M,q}$	-6.1
θ_r	8 deg

2.5 - Lateral-directional Dynamics

The lateral-directional dynamics are more complicated than the longitudinal dynamics since the roll and yaw terms are coupled. Previous parafoil experiments that the author is aware of relied almost entirely on flight test data to determine the lateral-directional behavior. Since, at present, only limited flight observations have been made in our project, we make due with rough estimates of the lateral-directional stability derivatives with the expectation that they will be tuned in the future.

A parafoil's lateral-directional control comes from asymmetric deflections of the outer trailing edges. The difference in drag between the two sides produces a yawing moment, and a constant turn rate is achieved when the yawing moment is balanced by damping terms.

Assuming that the yaw moment due to asymmetric flap deflections comes from the additional drag, we approximate its value in eq. 2.9. When calculating the moment, the resultant drag force is assumed to act halfway between the outside edge and centerline of the parafoil.

$$C_{N,\delta a} \approx \frac{1}{2}\left[C_{D,\delta f} \cos(\alpha + \theta_r)\right] \tag{2-9}$$

As mentioned above, the vehicle's turn rate depends on the yaw damping terms. The larger the damping, the slower the steady-state turn will be. Flight tests produced turn rates of approximately 0.6 rad/s (completes full circle in about 10 seconds) with about 50% of asymmetric deflection. Using this turn rate for our model, and if we initially only consider the yaw due to asymmetric deflection and the damping term, we can balance the moments to approximate the value for $C_{N,r}$.

$$C_{N,Steady} \approx C_{N,\delta a}\delta_a - C_{N,r}\frac{b \cdot r}{2V_T} = 0 \tag{2-10}$$

The other lateral-directional terms, roll and side force, are more difficult to approximate without extensive flight testing. It was decided to use the aerodynamic derivatives observed in another parafoil project until our own flight data could become available. The only data that could be found came from a project dealing with a larger parafoil [5]. As an attempt to better fit the faster dynamics of our small parafoil, the lateral-directional

stability terms were scaled so as to be in proportion to the yaw moment and yaw damping terms previously estimated. Although this is a somewhat crude estimate for the dynamics, it does produce steady-state turn rates and bank angles that are in rough agreement with the observations made from flight tests.

The complete lateral-directional equations used in the simulation are given below. The moments are with respect to the system CG location. Table 2-2 lists the values for the various stability derivatives

$$C_Y = C_{Y,\beta}\beta + C_{Y,\delta a}\delta_a \tag{2-11}$$

$$C_l = C_{l,\beta}\beta + C_{l,\delta a}\delta_a + \frac{b \cdot p}{2V_T}C_{l,p} + \frac{b \cdot r}{2V_T}C_{l,r} \tag{2-12}$$

$$C_n = C_{n,\beta}\beta + C_{n,\delta a}\delta_a + \frac{b \cdot p}{2V_T}C_{n,p} + \frac{b \cdot r}{2V_T}C_{n,r} \tag{2-13}$$

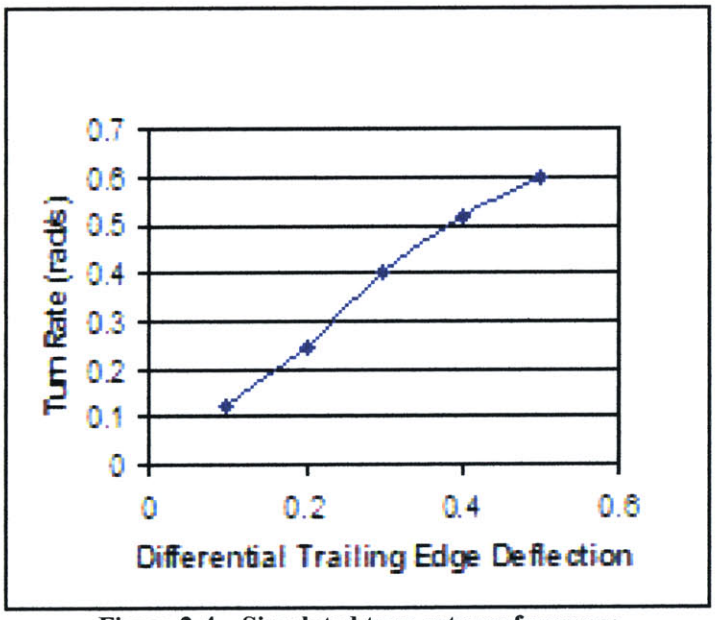


Figure 2-4 – Simulated turn rate performance

Table 2-2 Lateral-directional derivatives

$C_{Y,\beta}$	-0.24
$C_{Y,\delta a}$	-0.0096
$C_{l,\beta}$	-0.04
$C_{l,\delta a}$	-0.252
$C_{l,p}$	-4.5
$C_{l,r}$	0.8
$C_{n,\beta}$	0.16
$C_{n,\delta a}$	-.04
$C_{n,p}$	0.8
$C_{n,r}$	-.16

2.6 - Apparent Mass

A ram-air parachute's structure is very light, which means that it is heavily influenced by the air passing over and around it. The pendulum motion of a parafoil system is affected since the point of rotation moves away from the CG of the system and towards the parafoil. Furthermore, because the magnitude of the "apparent mass" effect depends on the axis of rotation, there may exist multiple points of rotation for the parafoil system.

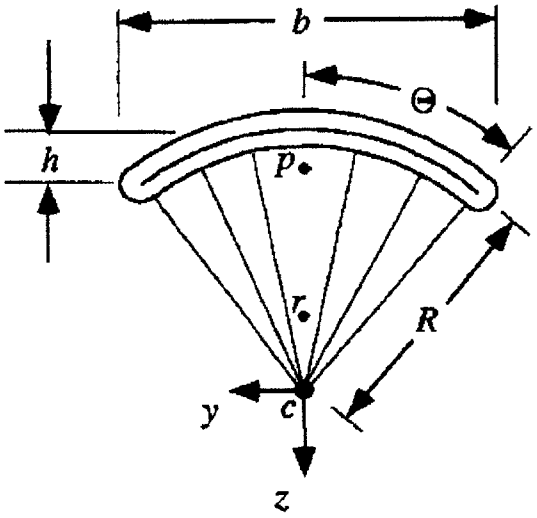


Figure 2-5 – Parafoil apparent mass geometry, centers of rotation [2]

We can describe the momentum of the system about an arbitrary point, 0, as

[2] Diagram from [7]

$$\begin{bmatrix} P \\ h_0 \end{bmatrix} = \begin{bmatrix} A_{0,R} + A_{0,A} \end{bmatrix} \begin{bmatrix} v_0 \\ \omega \end{bmatrix} \tag{2-14}$$

The linear and angular momentum terms, P and h_0, are related to the system's velocity by the 6x6 inertia matrix, which can be split up into real and apparent mass terms, $A_{0,R}$ and $A_{0,A}$.

If we place the origin, 0, at the center of the real mass, and if we assume that it lies on the same line as the pitch and roll centers (as shown in figure 2-5), the inertia matrices take the form:

$$A_{0,R} = \begin{bmatrix} M_R I_{3x3} & 0_{3x3} \\ 0_{3x3} & J_{0,R} \end{bmatrix} \qquad A_{0,A} = \rho \begin{bmatrix} m_{11} & 0 & 0 & 0 & -m_{11}a_{po} & 0 \\ 0 & m_{22} & 0 & m_{22}a_{ro} & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & m_{22}a_{ro} & 0 & J_{o11} & 0 & 0 \\ -m_{11}a_{po} & 0 & 0 & 0 & J_{o22} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{o33} \end{bmatrix}$$

where the apparent rotational inertia terms about the origin are found from the following equations:

$$J_{o11} = I_{11} + a_{ro}^2 m_{22}$$
$$J_{o22} = I_{22} + a_{po}^2 m_{11}$$
$$J_{o33} = I_{33}$$

M_R: real mass
$J_{o,R}$: Real rotational matrix about o

m_{ii}: Apparent mass terms
$J_{o,ii}$: Apparent rotational inertia terms
a_{po}: Distance from pitch center to origin
a_{ro}: Distance from roll center to origin
I_{11}: Rotational inertia about roll center
I_{22}: Rotational inertia about pitch center
I_{33}: Rotational inertia about yaw center

Three different apparent mass terms exist since the apparent mass effect is different depending on the direction of the airflow. For instance, a parafoil moving forward will

displace a greater amount of air than if it were moving in the sideslip direction. Values for the apparent mass parameters were obtained using the flat plate approximation methods described in [7]. They are based on the assumption that the apparent mass is related to the cross-sectional area perpendicular to the direction of the flow.

Finally, we need to estimate the roll and pitch centers, a_{ro} and a_{po}. The pitch center of rotation is assumed to be located at the centroid of the circular arc. Accounting for sideslip and rotation disturbances, the shift in the roll center from the confluence point is approximated by

$$a_{rc} = \frac{a_{pc} m_{f22}}{m_{f22} + I_{f11}/R^2} \tag{2-15}$$

a_{rc} is the distance from the roll center to the confluence point.

2.7 - Final Form of the Equations of Motion

Putting together the forces and moments with the inertia terms, we can write the complete equations of motion.

$$V = \begin{bmatrix} v_1 & v_2 & v_3 & \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$$

$$\dot{V}^T = \left(A_{oR} + A_{oA}\right)^{-1} \left\{ \begin{bmatrix} F_{3x1} \\ M_{3x1} \end{bmatrix} - \Sigma\left(A_{oR} + A_{oA}\right)V - U\left(A_{oR} + A_{oA}\right)V \right\} \tag{2-16}$$

$$\Sigma = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & 0 & 0 & 0 \\ \omega_3 & 0 & -\omega_1 & 0 & 0 & 0 \\ -\omega_2 & \omega_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_3 & \omega_2 \\ 0 & 0 & 0 & \omega_3 & 0 & -\omega_1 \\ 0 & 0 & 0 & -\omega_2 & \omega_1 & 0 \end{bmatrix} \qquad U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -v_3 & v_2 & 0 & 0 & 0 \\ v_3 & 0 & -v_1 & 0 & 0 & 0 \\ -v_2 & v_1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The aerodynamic forces and moments, along with gravity, make up F_{3x1} and M_{3x1}. The last two force terms in eq. 2.16 account for the non-linear effects due to the motion of the system.

Before we can move on to Chapter 3, we need to examine equation 2-16 a bit more closely. If we expand the section, $U(A_{oR} + A_{oA})V$, we get:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ -V_3((M_{22} + m_{22})V_2 + m_{22}a_{ro}\omega_1) + V_2((M_{33} + m_{33})V_3) \\ V_3((M_{11} + m_{11})V_1 + m_{11}a_{po}\omega_2) - V_1((M_{33} + m_{33})V_3) \\ -V_2((M_{11} + m_{11})V_1 + m_{11}a_{po}\omega_2) + V_1((M_{22} + m_{22})V_2 + m_{22}a_{ro}\omega_1) \end{bmatrix}$$

The portions that contain the real mass and velocity terms make up effects already accounted for in the stability derivatives. Therefore, to avoid double counting, equation 2-16 needs to be modified. We can leave equation in its present form if we note that setting the translational mass terms in the upper quadrant of the inertia matrices to zero removes the unwanted terms.

3. Trajectory Planning

One of the main advantages of the parafoil method of delivery is the longer flight time, which can increase mission flexibility. With a gliding capability that is unavailable for the ballistic configuration, a parafoil-guided drop could potentially be deployed miles away from the target. It could also use that extra time to spiral above the target, surveying the landing area. But the slow descent complicates other aspects of the delivery. Careful attention must be made to the path to target, and a trajectory for the vehicle must be planned in advance.

This chapter discusses a trajectory algorithm that may be used to guide the delivery vehicle during its descent. Alone, it cannot take the vehicle all the way to the target, but its purpose is to ensure that the vehicle will end up at a place in the sky where the vision system can take over and guide it in the rest of the way.

3.1 - Assumptions

It is advantageous for the vehicle to make its final approach to the target heading into the wind. This reduces the ground speed, essentially creating a bias in the horizontal direction that allows for steeper glide slope control. The lower ground speed is also useful since it lessens the impact during landing. Therefore, the approach point should be placed downwind from the landing zone whenever possible.

As long as control deflections are small, it is a reasonable assumption that the vehicle will be traveling at a constant airspeed and vertical velocity. This simplifies the computations when creating the trajectory since the vehicle's position can be predicted far in advance.

Knowledge of wind speed and direction is required before a successful trajectory can be computed. It is assumed that the UAV dropping the delivery vehicle can provide an initial wind estimate, possibly even deploying a wind measurement sensor to create a wind profile as a function of altitude. It is also very important that accurate estimates of the vehicle position are available, not only at the launch point, but throughout the trajectory. Winds can shift, or the vehicle may perform somewhat differently than what was initially assumed. Adjustments to the trajectory would be needed to ensure the vehicle reaches the final approach point.

3.2 - Methodology

To compute a workable trajectory, the algorithm takes in the following as inputs: delivery vehicle position, target position, horizontal and vertical trim speeds, wind speed/direction, and heading.

$$(\mathrm{Traj_N}, \mathrm{Traj_E}, \mathrm{Traj_Z}) = f(\mathrm{pN}, \mathrm{pE}, \mathrm{pD}, \mathrm{pN_{tar}}, pE_{tar}, pD_{tar}, V_{hor}, V_Z, wind_N, wind_E, \Psi)$$

The basic strategy of the algorithm is to take advantage of the constant vertical velocity assumption to determine the flight time. Given the vehicle's current altitude and trim vertical velocity, the time to arrive at the approach point can be calculated.

$$t_{traj} = \frac{(h_i - h_f)}{V_Z} \tag{3-1}$$

The path to target is comprised of three straight legs and two circular turns of a specified radius. An iterative procedure is used to find a trajectory that gets to the approach point. The algorithm begins with an initial path in 2D, meaning that it ignores the descent rate and only considers the north and east components of the flight path. Taking into account the wind conditions and the vehicle's airspeed, ground speed can be calculated throughout the path, and the trajectory time is determined by

$$t_{2D} = \int_{l_i}^{l_f} \frac{dl}{V_G(l)} \tag{3-2}$$

This time is compared to the time it takes to descend to the final altitude. The path can be lengthened iteratively until the two times, t_{traj} and t_{2D} equal one another.

Example trajectories are shown in figures 3-2 and 3-3. The dots represent the vehicle's planned position in 1 second intervals. Note that as the trajectory turns into the wind, the dots are closer together since the groundspeed is reduced. The second example (shown in figure 3-3) utilizes the same initial conditions as the first, only the starting altitude is raised by 40m. This increases the flight time, so the trajectory loops further away from the target.

Ground speed is calculated at every point along the trajectory, which allows the trajectory algorithm to accept different wind conditions at various altitudes. The examples in figures 3-2 and 3-3 use a single wind vector for all altitudes, but a wind profile with varying magnitude and direction may be used if such data is available.

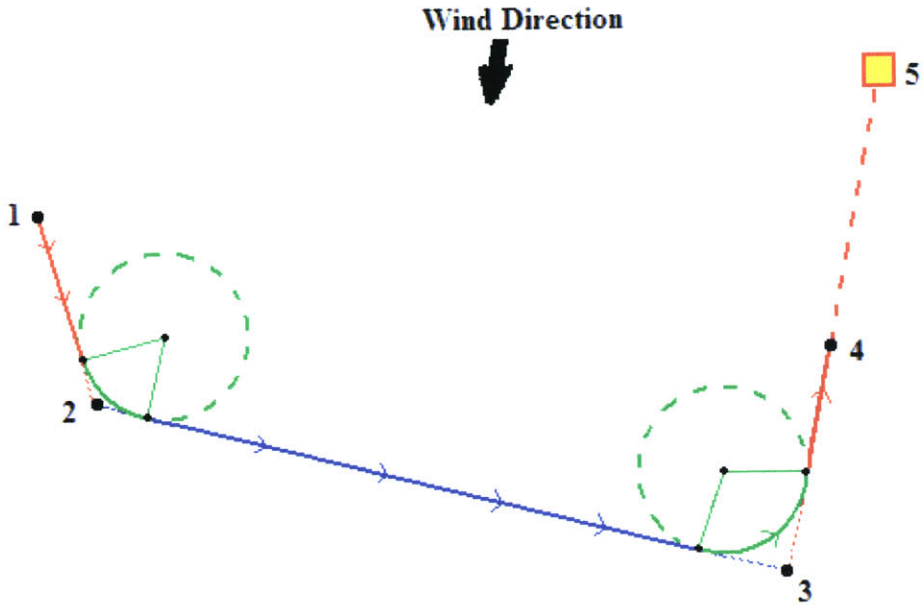


Figure 3-1 – Trajectory geometry

The following summarizes the method employed in computing a workable trajectory from the starting position to the approach point. Referring to figure 3-1:

2D Trajectory Computation

1) Trajectory starts at vehicle's initial position (1)
2) Trajectory ends at approach point (4), which is a specified distance from the target (5) in the direction of the wind.
3) The initial leg (1 - 2) is a straight path of a fixed length in the direction of the vehicle's initial ground speed.
4) The approach leg (3 - 4) is the **variable leg** – its length is increased incrementally during the iteration process. The leg is a straight path on the same line as the approach point and target.
5) The main leg (2 – 3) connects the initial and approach legs.
6) Circular turns of a specified radius modify the 3-legged trajectory to allow for gradual turns. The lengths of the arcs depend on the required turning angle, and the position of the arcs is found using geometric relations.

Iteration

7) Knowing the vehicle's trim airspeed, descent rate, and the wind speed and direction, the ground speed can be calculated at all points along the trajectory.
8) The 2D trajectory time is computed using eq. 3-2, which is compared to the required trajectory time (eq. 3-1).
9) The variable leg is increased incrementally, and a new 2D trajectory is computed.
10) The process is repeated until the 2D trajectory time is equal to required trajectory time.

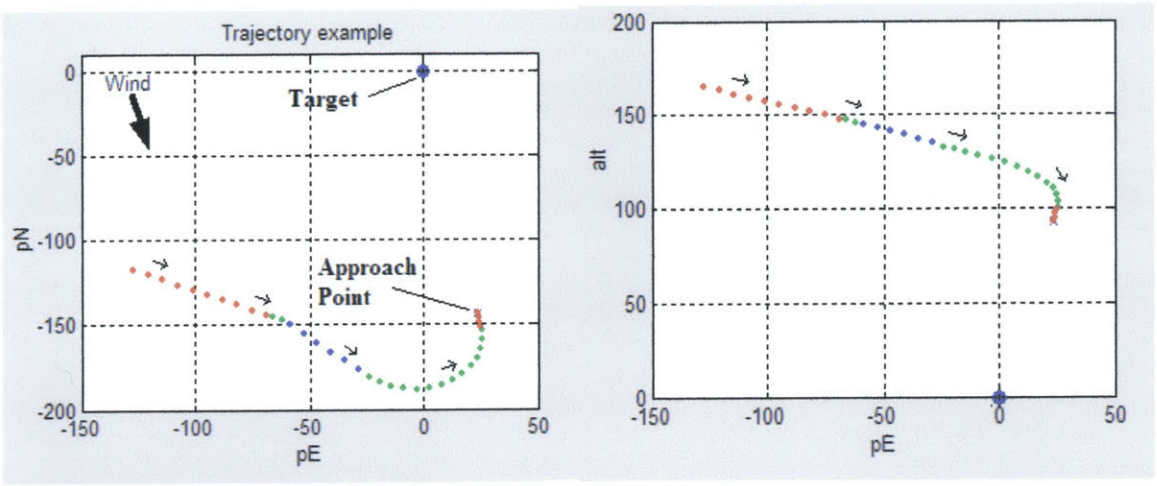


Figure 3-2 – Example trajectory – wind blowing from the NW – landing zone at (0,0)

The trajectory can be split up into 5 different segments.

-Initial straight leg (red dots) -- the vehicle continues along its initial heading for approximately 8 seconds
-Main leg (blue dots) – connects the two turning legs
-Turning legs (green dots) -- circular arcs of 30m radius
-Approach leg (red dots, bunched together) – Straight line to the approach point, heading into the wind. This is the variable leg in the iterative process

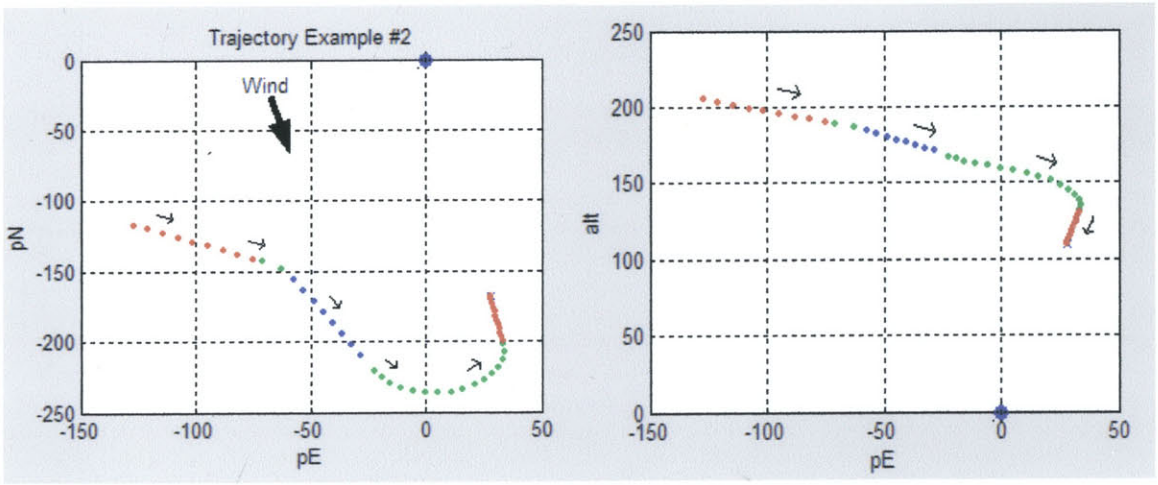


Figure 3-3 – Example trajectory where initial altitude is 40m higher than example 1

This trajectory methodology has its limitations. Normally, the higher the starting altitude, the longer the approach leg needs to be to accommodate the longer flight duration. At some point the computed trajectory becomes unreasonable since it wants to take the vehicle far away from the landing zone before bringing it back around. To avoid this, the delivery vehicle is initially instructed to circle around near the landing zone until it descends to a lower altitude. When an acceptable trajectory can be computed, the vehicle leaves the orbiting trajectory and follows the new path to the approach point. The energy management phase can serve other purposes as well, such as surveying the landing zone or taking wind measurements.

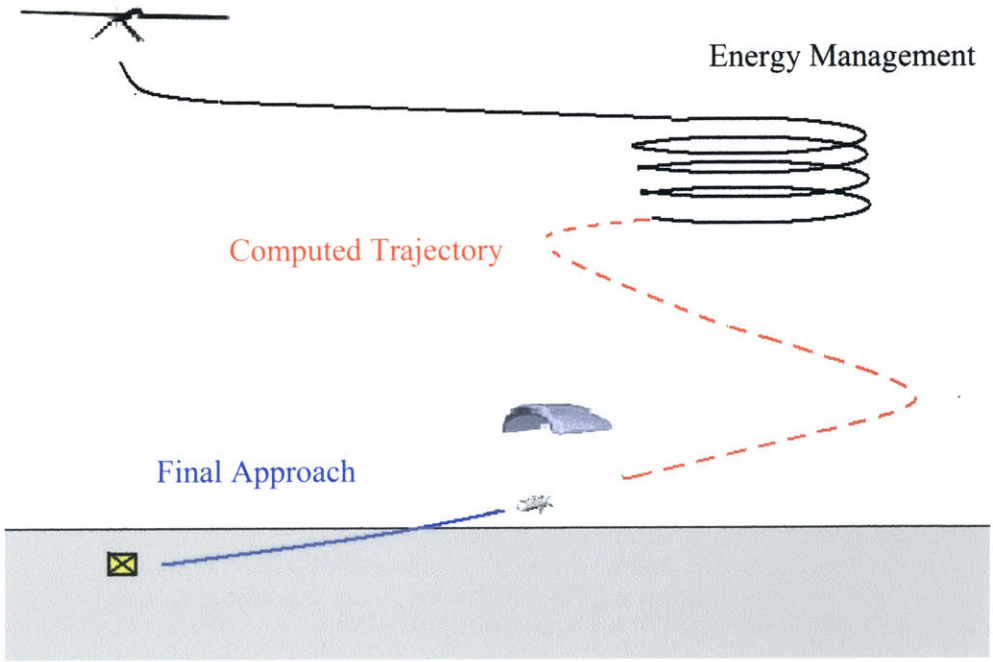


Figure 3-4 – Typical mission using an energy management phase before trajectory is computed

The transition between the energy management portion of the descent and the computed trajectory can occur in a number of ways. A simple method would be to continuously

compute trajectories as the vehicle descends using its present position and velocity as the initial conditions for the algorithm. The trigger logic for leaving the spiraling descent could be based on a maximum allowable distance from the target, a distance threshold, as shown in figure 3-5. Once the entirety of a computed trajectory is within the distance threshold, the guidance system begins following it.

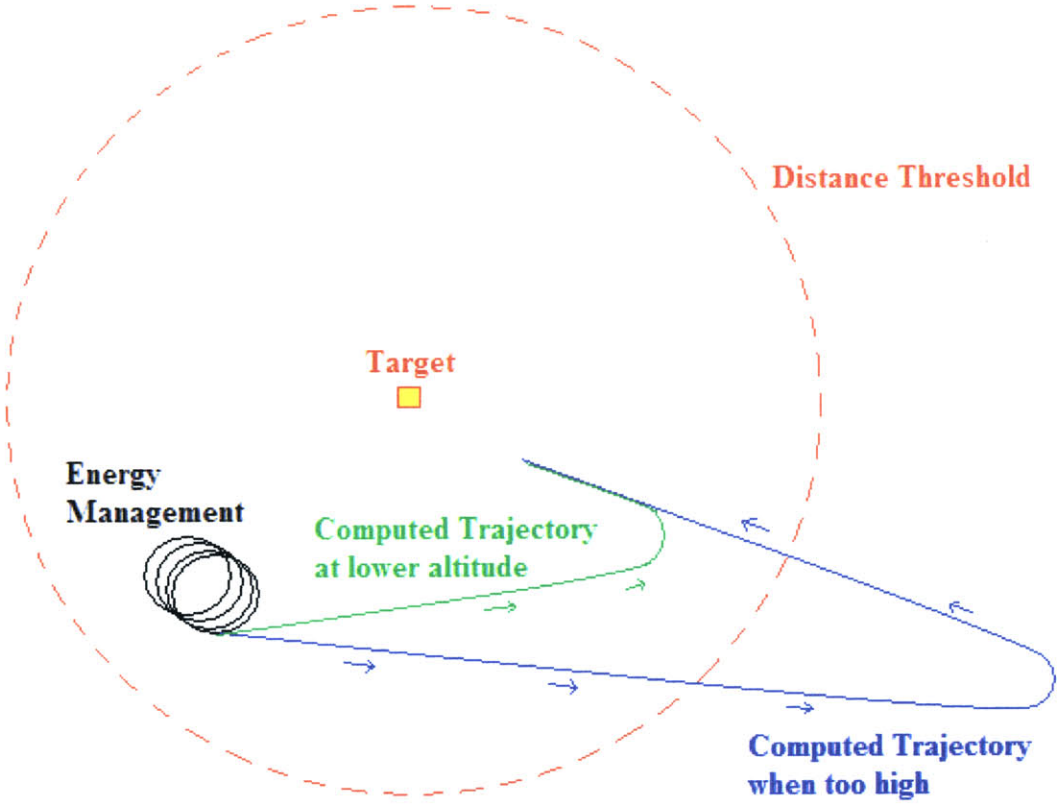


Figure 3-5 - Switching from energy management to computed trajectory

3.3 - Wind-reference Method

When testing the trajectory following capabilities of the guidance system (guidance system discussed in more detail in next chapter), it was noticed in simulation that the vehicle sometimes had a difficult time tracking the turning legs, especially when descending through a heavy crosswind.

As mentioned earlier, it was assumed that all turning legs of the trajectory would be circular arcs. However, as a vehicle follows the turning leg, its ground speed changes as it turns through the wind conditions. If the vehicle's ground speed decreases through the turn, the turn rate that is required to keep it on course also decreases. And likewise, if the vehicle is turning away from the wind, it speeds up as it turns, and therefore needs to turn harder towards the end of the leg to remain on the circular path. Due to the slow dynamics of our parafoil system, there is a noticeable time lag between the commanded turn rate and the vehicle's response. During a turn, if the required turn rate changes quickly, the guidance system may not have sufficient time to adjust.

The wind-relative method is a modification to the original trajectory algorithm. Rather than compute a trajectory in an Earth-relative frame and keep track of the ground speed, this trajectory is computed relative to the wind. In this reference frame, the target shifts from its actual location to where it would be if no wind were present. The advantage is that the vehicle's speed relative to the wind is always constant. A circular arc in this reference frame is not a circular arc in an Earth-relative frame, but since the airspeed is assumed to be unchanging, the turn rate is constant.

The modified algorithm works much the same way as the original method. Once the final approach point is determined, it is shifted by:

$$Shift_N = -wind_N \frac{z_i}{V_z} \qquad Shift_E = -wind_E \frac{z_i}{V_z} \tag{3-3}$$

$Shift_N$ and $shift_E$ represent the drift due to the wind during the descent, where a faster descent reduces the drifting. The approach point in the wind-relative reference frame becomes

$$p_{Ntar,W} = p_{Ntar} + Shift_N \qquad p_{Etar,W} = p_{Etar} + Shift_E$$

A trajectory is then computed using the shifted approach point and setting the wind magnitude to zero. Then it is transformed back into an Earth-relative reference:

$$traj_N(t) = \frac{wind_N[traj_Z(0) - traj_Z(t)]}{V_Z} \qquad (3\text{-}4)$$
$$traj_E(t) = \frac{wind_E[traj_Z(0) - traj_Z(t)]}{V_Z}$$

Figures 3-6 and 3-7 make the comparison between the wind-relative method and the original trajectory algorithm for the same initial conditions. When shifted back into navigation axes, the wind-relative trajectory no longer has a circular final turn leg. Instead, the turn is gradual when the ground speed is high and turns sharper as the groundspeed decreases. The varying turning radius compensates for the transition between groundspeeds to allow the vehicle to follow the trajectory curve at a constant turn rate.

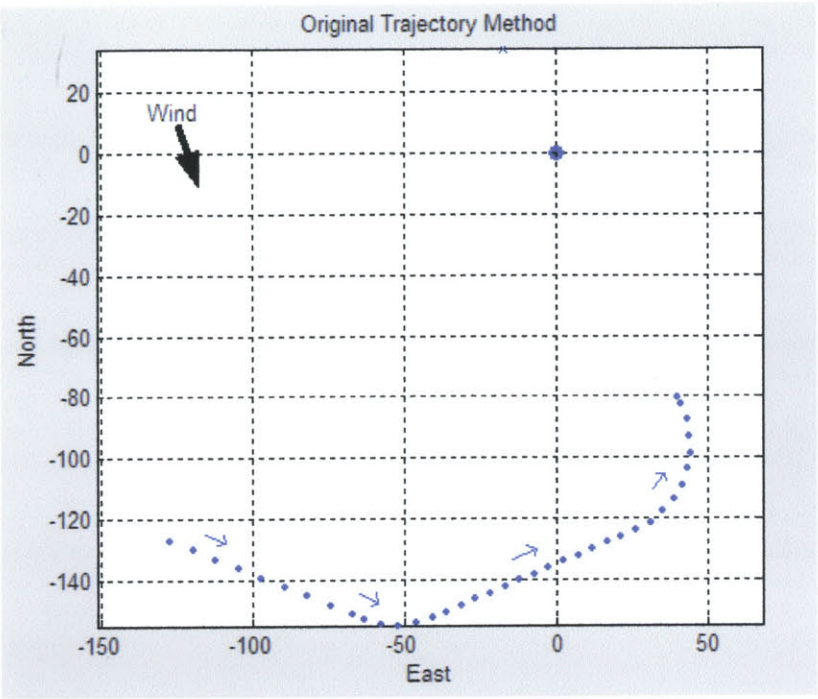
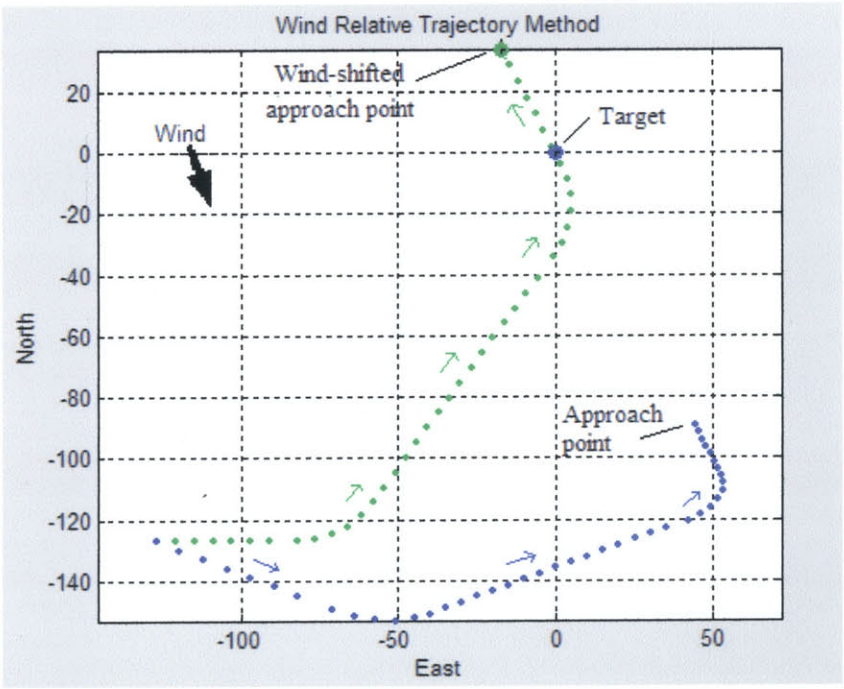


Figure 3-6 – Trajectory computed in Earth coordinates



The green dots represent the trajectory in the wind-relative frame.

The blue dots represent the trajectory when shifted back into N,E,D coordinates.

Figure 3-7 – Trajectory created in a wind-relative reference frame, then shifted back to Earth coordinates

Simulations were performed to compare the two methods. An improvement in the trajectory tracking was observed, especially when the turning angles were large. This will be discussed in more detail in the next chapter.

3.4 – Trajectory Update

An important component of the trajectory algorithm is its ability to adapt to changing conditions in real-time. As the parafoil delivery vehicle descends toward the ground, adjustments in its course may be necessary. The vehicle may have trouble following its planned trajectory, and if such deviations can be detected, it is helpful to modify the trajectory to give the best possible chance of reaching the approach point.

There are a variety of situations that may cause the vehicle to drift off its intended path. Many assumptions were made in the computation of the trajectory, most notably, that the wind speed and direction, vertical and horizontal velocity were known. Errors in those estimates will lead to a less than optimal trajectory. It is also possible that the assumptions are so wrong that the trajectory may be unattainable altogether. In addition, conditions may change mid-flight. Wind speed and direction can vary with both time and altitude – a single wind estimate obtained just before dropping the vehicle may not be sufficient. Various possible update methods have been considered, and there are advantages and disadvantages to each of them depending on the capabilities of the drop vehicle.

The first update option is to re-compute the trajectory at regular time intervals by replacing the vehicle's initial position with its current position, but leaving the wind parameters constant. This is the simplest method, as it does not pay attention to the vehicle's tracking ability. It is very easy to implement since it relies on a just a few inputs, but it is limited by the fact that it does not adjust to changing flight conditions. For instance, if the vehicle descends at higher rate than anticipated, all future trajectory updates would continue to overestimate the vehicle's gliding ability.

An enhancement to that update procedure is to adjust the key parameters used in the trajectory algorithm based on the available telemetry data. This method works well if those inputs ($wind_N$, $wind_E$, V_{hor},V_z) are accurately known, but there will always be uncertainty associated with each. For instance, when measuring wind speed, the airspeed component in the lateral direction is undetected, which disrupts the ability to accurately measure the wind component in the direction normal to the vehicle's heading.

A second method for updating the trajectories has been examined. This method eliminates the need for a separate wind estimator by relying solely on the drift position of the vehicle. Each time a trajectory is updated, the vehicle's current position is compared to the position anticipated by the present trajectory. The onboard navigation system looks at the North-East-Down components of the drift error and assumes that they reflect changes in wind conditions.

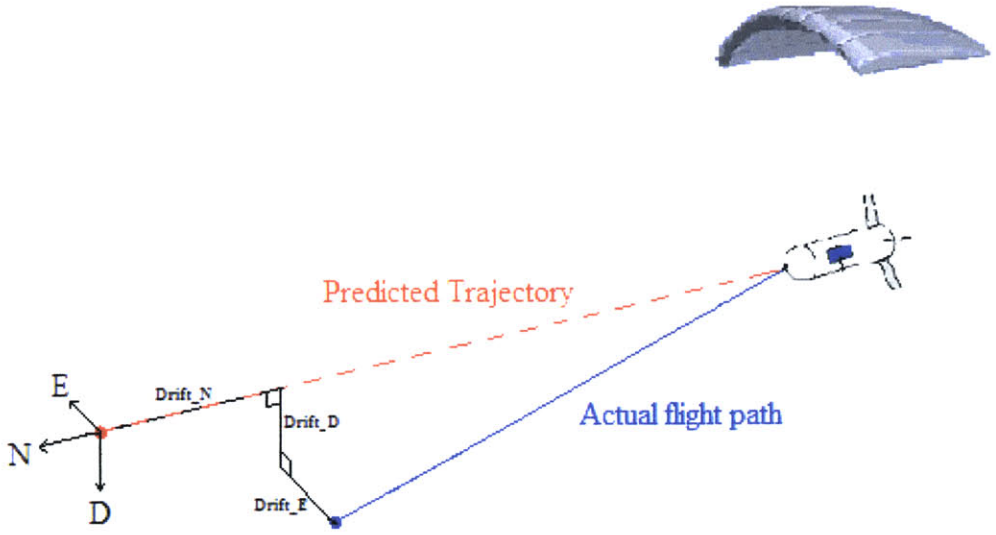


Figure 3-8 – Drift error

$$Drift_N = pN(t_f) - pN_{predicted}(t_f) \qquad Drift_E = pE(t_f) - pE_{predicted}(t_f) \tag{3-5}$$

The updated wind parameters are determined by,

$$Wind_N' = Wind_N + \frac{Drift_N}{\Delta t} \qquad Wind_E' = Wind_E + \frac{Drift_E}{\Delta t} \tag{3-6}$$

Similarly, the vehicle's descent rate parameter can be adjusted by comparing the actual altitude with the predicted altitude.

Figures 3-9 and 3-10 show how the trajectory update process works. In that example, the vehicle's gliding capability was overestimated when the initial trajectory was created. After 4 seconds, the vehicle had descended more than anticipated, so the update called for a shortened trajectory.

Since the updated parameters are based on drift error, they reflect the average change in conditions, so the update process tends to be more conservative than the first method. Therefore, it is less likely to overreact to temporary gusts or other disturbances.

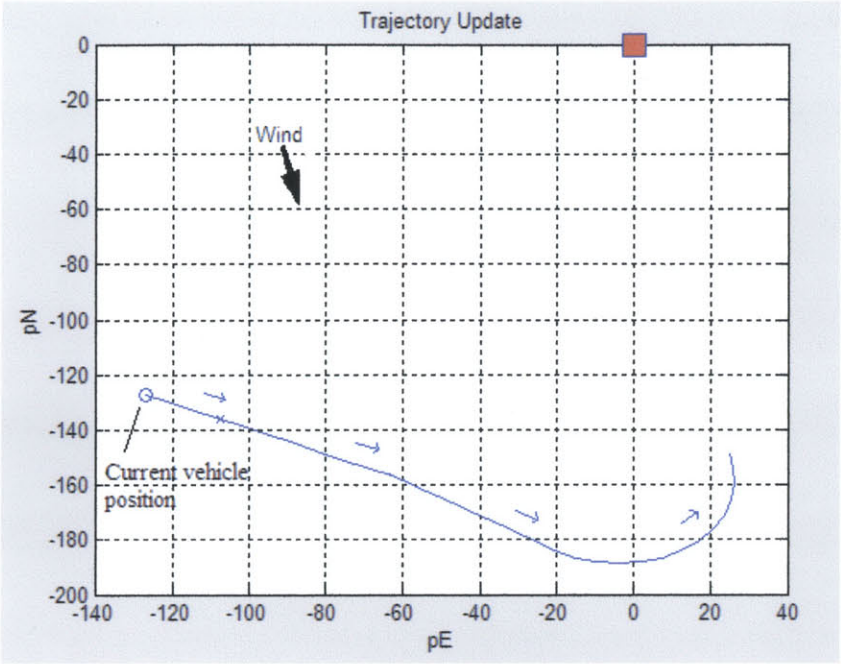


Figure 3-9 – Initial trajectory

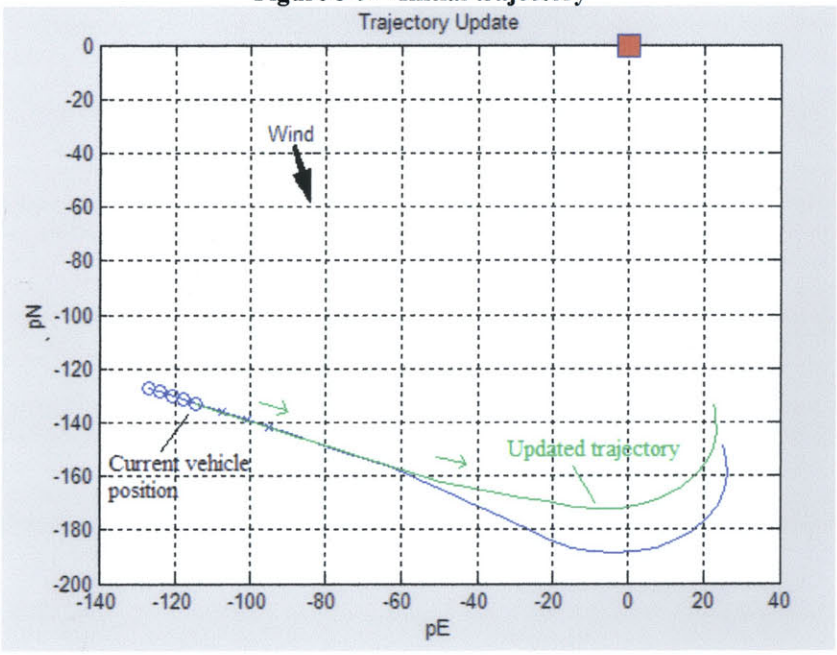


Figure 3-10 – Trajectory updated (in green) after 4 seconds using drift method

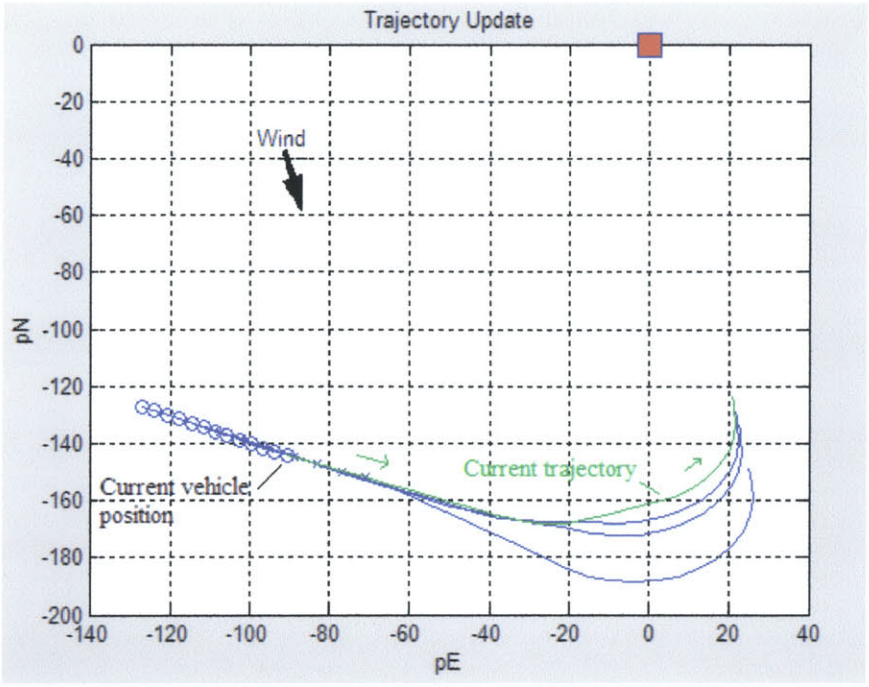


Figure 3-11 – Subsequent updates using drift method

There are limitations to the update process. For one, the drift errors only help to estimate wind conditions at the vehicle's present height altitude, therefore it is unclear how to update an entire wind profile. The simulation presented in figure 3-10 and 3-11 assumed that the same incremental wind shift can be applied at all altitudes, which is unlikely to accurately portray the actual wind conditions.

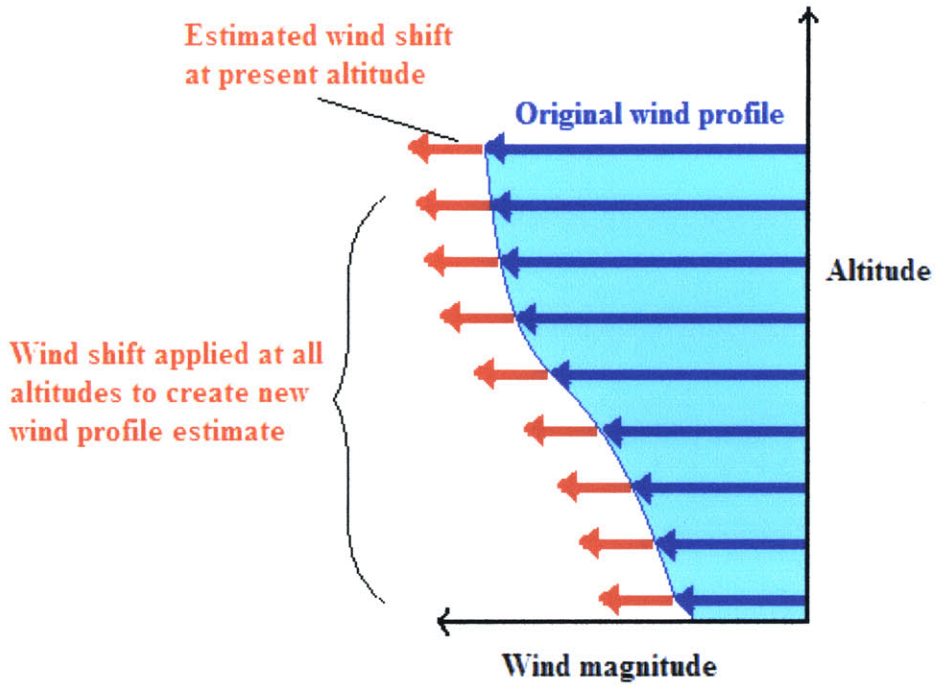


Figure 3-12 - Wind update based on drift error

Since the guidance system will attempt to follow the planned trajectory, the control system will limit the drifting, especially in the lateral direction. It therefore becomes difficult to measure wind changes in the direction perpendicular to the flight path. Although it was not performed in the simulations, it may be feasible to use a time history of flap commands to keep track of the controller's effort in maintaining the correct heading. That knowledge could be used to isolate the drift associated with the wind.

4. Trajectory Guidance and Control

There are issues that present a challenge to the guidance of our delivery vehicle. For one, the vehicle is a very light, and its parafoil makes it slow and very susceptible to wind and other disturbances. Keeping the vehicle on a planned trajectory becomes more difficult with the longer flight durations. Accurate guidance requires a decent ability to navigate throughout the descent.

4.1 - First Guidance Method

The first method is a simple guidance scheme that attempts to keep the vehicle's nose pointed at the desired waypoint. The equation that governs this guidance scheme is:

$$\dot{\Psi}_{Command} = K \cdot \Delta\Psi = K\left(\Psi - \Psi_{Command}\right) \tag{4-1}$$

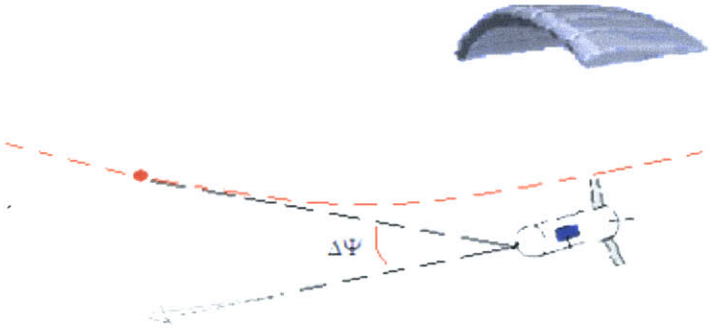


Figure 4-1 – First guidance law

$\Delta\psi$ is the angle between the direction to the target and the vehicle's current heading vector. A positive $\Delta\psi$ means the target is on the vehicle's right side, while a negative value for $\Delta\psi$ produces a left turn command. The gain value, K, reflects the

responsiveness of the guidance law. Higher K values produce larger turn rate commands but may lead to instability.

Asymmetric flap deflections are used to produce the desired turn rate. To help avoid stability problems, both roll and yaw rates are fed back in the inner control loop that determines the flap command.

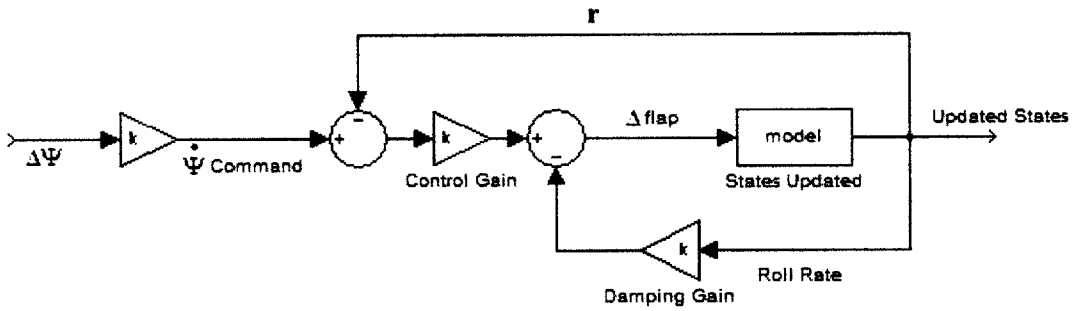


Figure 4-2 – Block diagram of 1st guidance law

In the presence of a crosswind, the vehicle's heading does not point in the direction it actually travels. The nose may be aimed at a waypoint, but a component of its ground speed is in the lateral direction. For fast traveling missiles, this isn't much of a problem since the wind's magnitude is small compared to the airspeed. However, our parafoil is strongly influenced by the wind and will gradually drift off the desired path, as is evident in the simulated run shown in figure 4-2. The crosswind causes the vehicle to belly out from its intended trajectory, and the vehicle ends up undershooting the approach point.

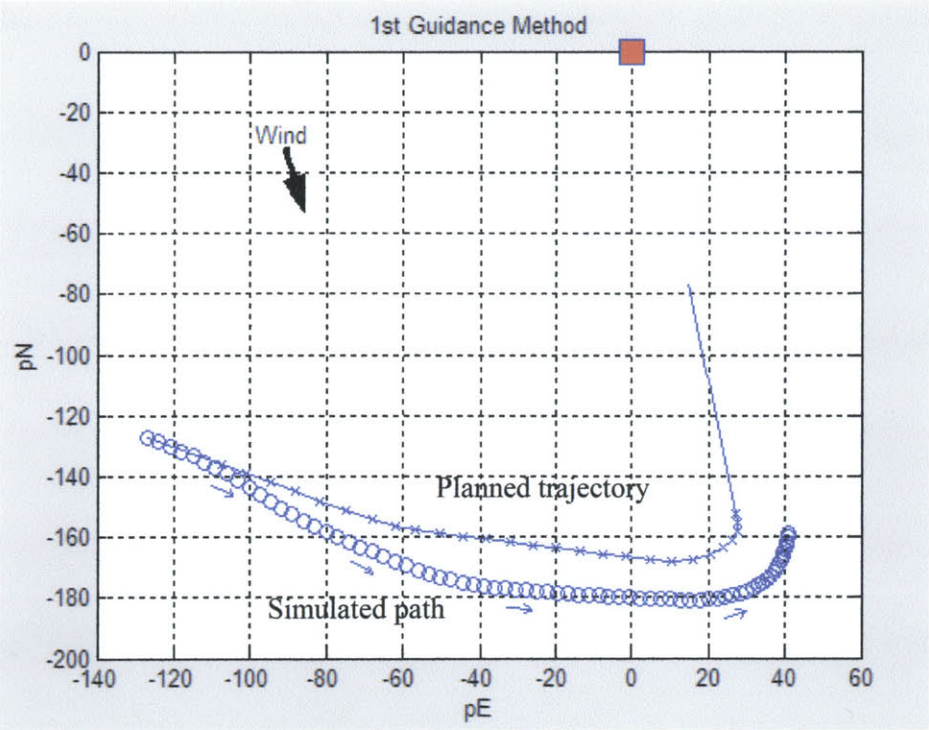


Figure 4-3 – Trajectory following using $\dot{\Psi}_{Command} = K \cdot \Delta\Psi$

4.2 - Wind Compensation

The problem due to wind drift may be addressed if the guidance law references the groundspeed direction instead of the heading direction. Just as an airliner adjusts its heading to keep the velocity vector pointing at its destination, the desired heading angle may be adjusted so that the vehicle's ground velocity points in the correct direction.

The tracking angle, ε, represents the difference between the heading and the velocity. η is the angle between the velocity vector and the aim point.

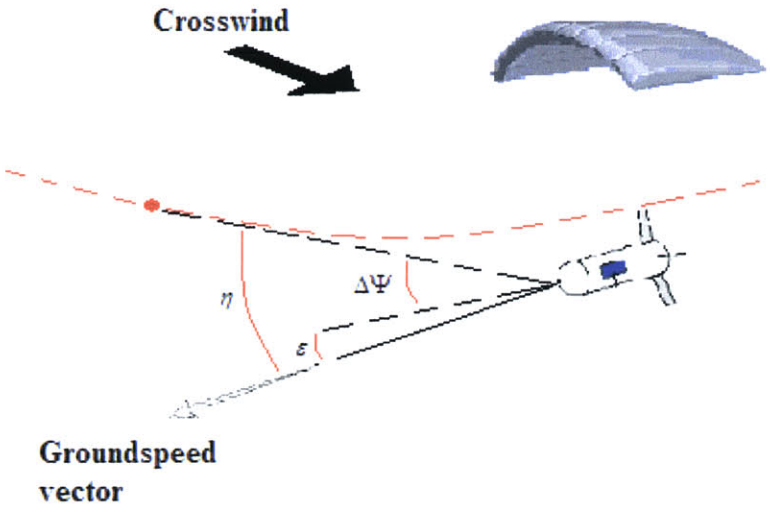


Figure 4-4 – Wind compensation based on ground speed

The guidance law becomes:

$$\dot{\Psi} = K(\Delta\Psi + \varepsilon) = K\eta \tag{4-2}$$

Guidance simulations implementing the wind compensation reveal a significant improvement in the ability to follow the trajectory.

4.3 - Nonlinear Method

A non-linear guidance law has also been tested and was developed by Sanghyuk Park for the PCUAV project at MIT. The commanded lateral acceleration is based on the centripetal acceleration needed to direct the vehicle to the waypoint in a circular arc. The governing guidance law is

$$A_C = \frac{V_G^2}{R}$$

where R is the radius of the instantaneous circular segment. η is the angle between the velocity vector and the aim point, and the distance to the aim point is related to η and R by

$$L = 2R\sin(\eta)$$

With this relation, the commanded acceleration can be rewritten as

$$A_C = \frac{2V_G^2}{L}\sin(\eta)$$

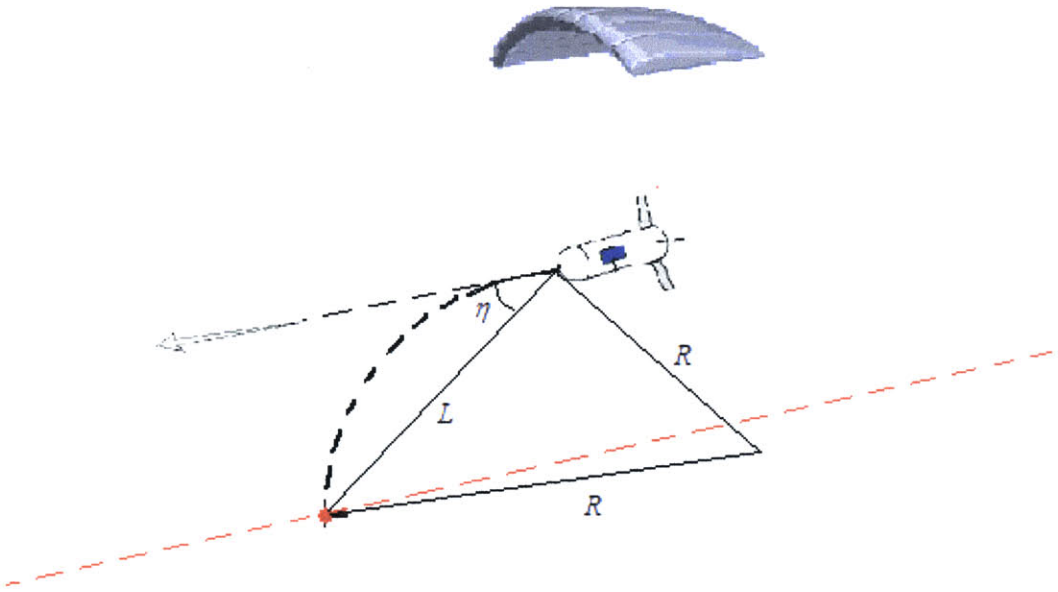


Figure 4-5 – Schematic of nonlinear guidance methodology

Just as with the previous guidance method, the wind conditions are already taken into account since their effects are captured in the ground speed. For instance, the commanded side force is less when the vehicle is turning into the wind – its ground speed is reduced, and therefore has more time to make the turn. When turning away from the wind, the ground speed is increased, and the same turn requires a higher control gain.

A conventional aircraft's bank angle directly corresponds to a specific lateral acceleration, so lateral acceleration commands are usually in the form of a bank angle command. For parafoil systems, it makes more sense to give acceleration commands in terms of turn rate, so if we use the relation:

$$A_C = \frac{V_G^2}{R} = \dot{\Psi}^2 R$$

and solving for turn rate yields,

$$\dot{\Psi} = \frac{2V_G}{L}\sin(\eta) \tag{4-3}$$

The commanded turn rate is executed with flap deflections, and yaw rate is measured to provide feedback.

4.4 - Choosing the Aim-point distance

In both guidance methods, the vehicle negotiates the trajectory by looking ahead a certain distance. Aiming for a waypoint that is too close would generate larger turn rate commands, which has a tendency to cause the vehicle to oscillate about its intended trajectory. Such a situation may lead to instability. Looking further ahead tends to smooth out the trajectory, but the vehicle's path may cut corners on turns. This is not desired since the vehicle's glide slope is assumed to be constant. A shortened path causes the vehicle to end up too high at the end of the trajectory.

A look-ahead distance was chosen that reflects a compromise, and it is constant throughout the descent. A distance of 30m appeared to produce the best tracking ability

in simulations. At every time update, a new aim point is chosen based on the vehicle's current position.

4.6 - Trajectory Comparisons

In the previous chapter two different types of trajectories were presented. One trajectory method computed everything in an Earth coordinate system, while the other created a wind-relative trajectory. The reasoning behind the second scheme is that it creates turning legs that the vehicle is able to negotiate at a constant turn rate. It was hoped that this would make it easier for the guidance systems to track the trajectory during the turns.

The linear guidance law, $\dot{\Psi} = K\eta$, was tested against both trajectory schemes. The guidance law faired poorly while negotiating turns created with the earth-fixed trajectory method. Simulations demonstrated that the vehicle had a tendency to temporarily slide off course during the beginning of the final turn. When the controller attempts to correct itself, it often overcompensates. As a result, this scenario can lead to oscillations around the nominal trajectory, which lengthens the vehicle's overall trajectory. As was mentioned previously, longer than expected trajectories lead to a lower altitude at then end, possibly resulting in the vehicle falling short of the target.

The observed difficulty is a consequence of the fact that the parafoil has a low trim airspeed. Its velocity with respect to the ground is heavily influenced by the wind conditions. On a turn, the shift from tailwind to headwind can cause a rapid change in groundspeed. Since the original trajectory methodology called for a circular arc with

respect to the ground, the turn rate required to follow this turn would not be constant. The initial portion of the curve would require a large turn rate, and as the vehicle transitions into the wind, the turn rate would reduce. This change from a high to low turn rate presents problems for the slow dynamics of a parafoil. The lags in the system make it difficult for the controller to react to changing turn rates.

But when following the wind-relative trajectory, the linear guidance law fairs much better. Since the basis of the wind-relative method is to create turns that can be followed with a constant turn-rate, it is ideally suited for the linear law. The vehicle is no longer expected to react to changing turn-rates throughout its turn. Furthermore, since the estimate of the vehicle's trim airspeed is known, the required turn rate will be known beforehand. If, for instance, the trajectory was created with a 30 m turning radius with respect to the wind axis, the vehicle's required turn rate through the turn will be

$$\dot{\Psi} \cong \frac{V_T}{30}$$

The controller gains can then be tuned with the expected turn rate in mind, and the controller can quickly jump to the corresponding flap setting without having to wait for the vehicle's response. This alleviates some of the problems associated with the vehicle's slow dynamics and actuator lag.

Figure 4-8 presents a comparison between the two trajectory methods using the linear guidance law. The solid lines are the desired trajectories, while the series of circles represent the vehicles actual flight path. The plots demonstrate how the linear guidance

law had trouble following the earth-relative trajectory during the turn, while it performed much better using the wind-relative trajectory.

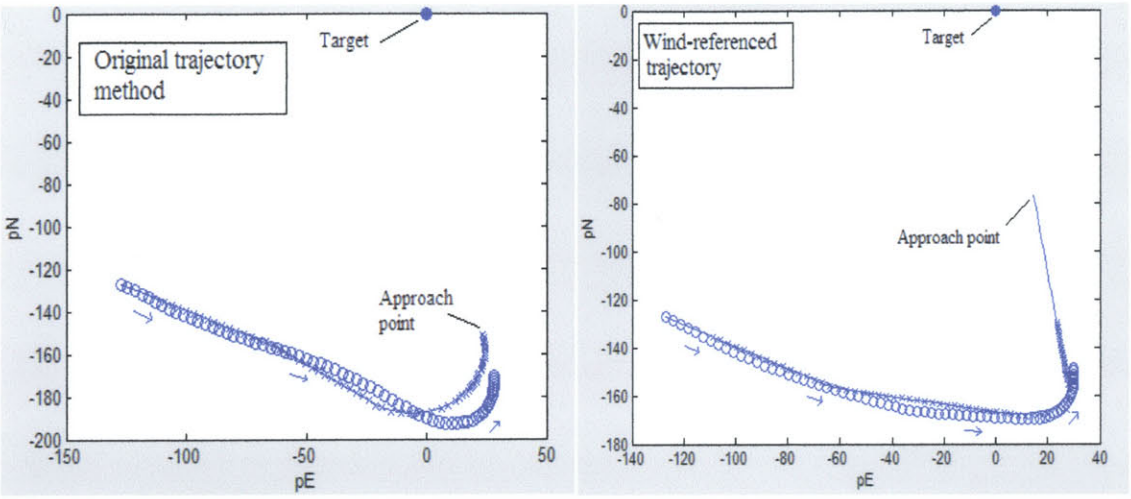


Figure 4-6 – Trajectory comparison using $\dot{\Psi} = K\eta$ guidance law

It is interesting to note that when using the nonlinear guidance system, the performance varied little between the two trajectory types. The control gain is directly related to the ground speed, so it is better suited to adapt to the changing groundspeeds during a turn. And since the guidance method was originally intended to keep the vehicle in a circular pattern ($A_C = \frac{V_G^2}{R} = \dot{\Psi}^2 R$), it makes sense that it can successfully negotiate circular turns.

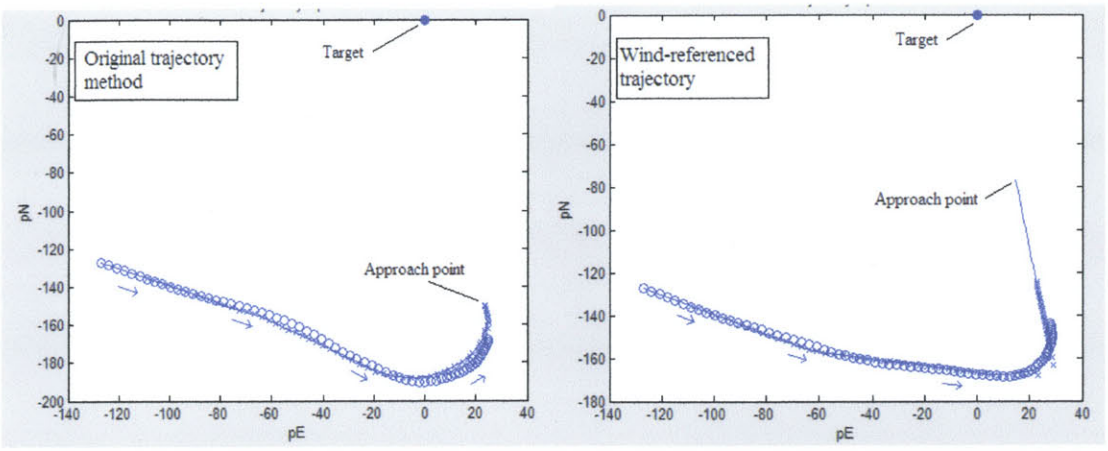


Figure 4-7 – Trajectory comparison using nonlinear guidance law

As shown in figures 4-8, 4-9, both guidance laws seem to perform equally well when following the wind-referenced trajectory. The advantages of the non-linear method are not needed since the turn-rate along the final curve is constant, therefore, it should make little difference which method is used when following the wind-referenced trajectory.

Nonlinear law $\dot{\Psi} = \frac{2V_G}{L}\sin(\eta) = cons\tan t$

Linear law $\dot{\Psi} = K(\Delta\Psi + \varepsilon) = K\eta = cons\tan t$

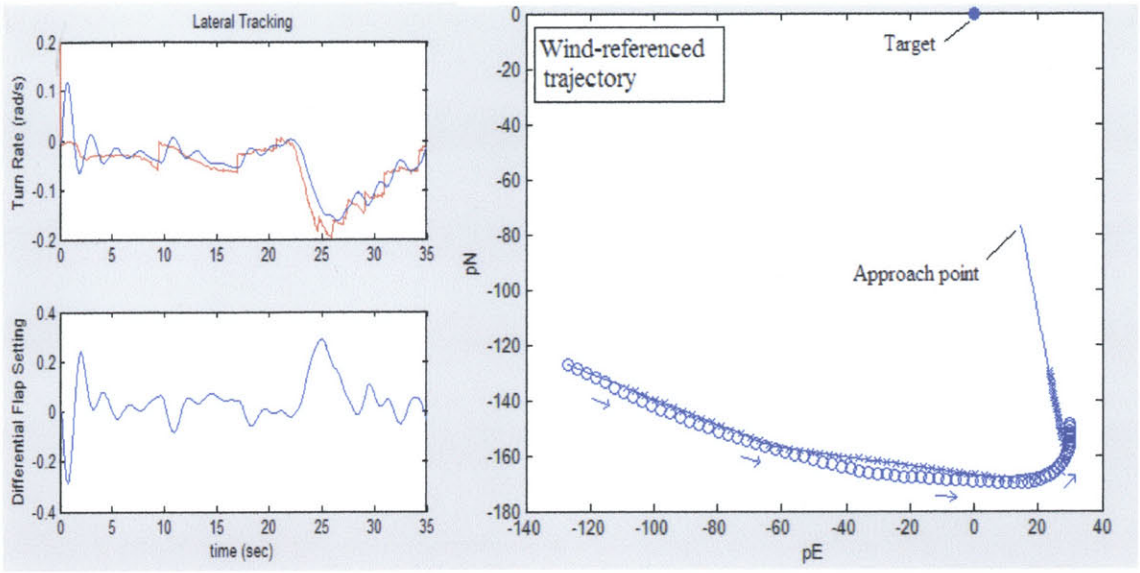


Figure 4-8 – Trajectory following using $\dot{\Psi} = K\eta$ guidance law

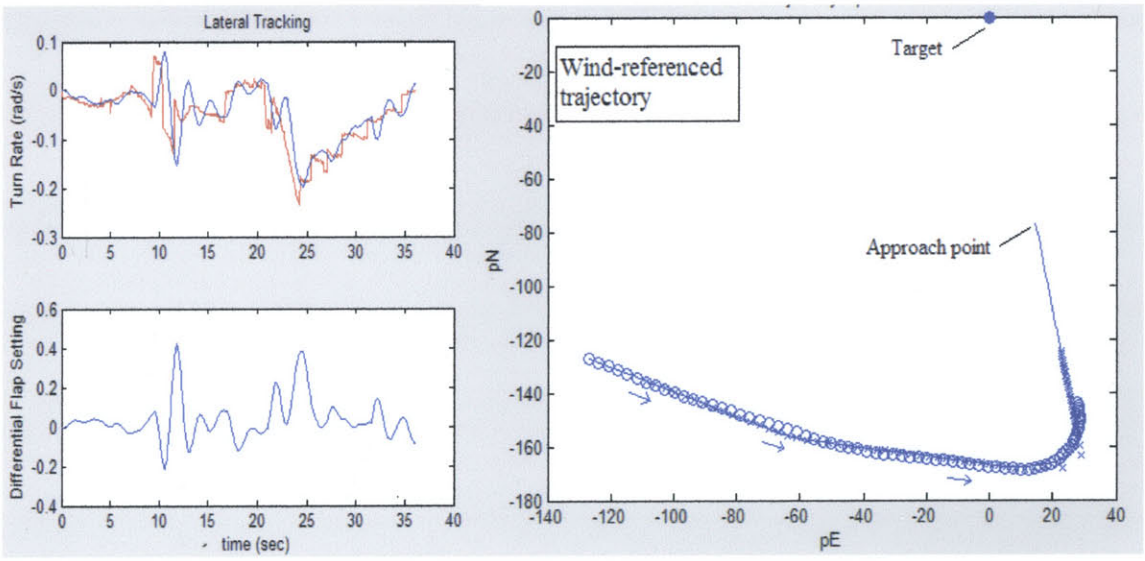


Figure 4-9 – Trajectory following using nonlinear guidance law

5. Final Approach

5.1 – Introduction

The final approach phase begins when the drop vehicle is facing the target, heading into the wind. The camera sends video of the landing site back to the ground station. A human operator is then able to designate the target on the screen, sending back pixel information to the vehicle. With the target's position updated, small corrections to the flight path are made to guide the vehicle to the target.

The guided drop must be able to achieve the desired 2-5m accuracy under a variety of conditions. Parafoils are heavily influenced by the wind - a late wind gust might blow the vehicle off course – so it is desirous for the control system to be able to react quickly to disturbances. This chapter describes the control methods that can be used to fulfill the accuracy requirements and attempts to quantify the magnitude of wind disturbances that can be tolerated.

5.2 - Glide Slope Control

The delivery vehicle must have the ability to adjust its flight path angle if it is to reach the target. This glide slope angle depends on the L/D ratio and can be manipulated mid-flight with symmetric flap deflections. However, changes in flap setting increase the lift coefficient as well as the drag, resulting in only a slight decrease in L/D. The parafoil of this project has an L/D range of 2.1 – 2.6, which corresponds to a glide slope angle range of only 10 deg. Alone, this does not leave much room for error when guiding the vehicle

to its target. But in the presence of a headwind, flap deflections change the groundspeed more than the airspeed, and the glide slope range relative to the ground (and therefore relative to the target) is more effective. For instance, a headwind of 5mph produces a glide slope range of 20 deg, double the range achieved in still conditions.

When two objects are on a collision course, the angle between them remains constant. The same is true for the glide slope angle as the vehicle approaches the target, so a proportional control law is used to keep the angle-to-target constant during the final approach. Symmetric flaps are deflected to manipulate the glide slope angle.

$$\delta_f = K \cdot \dot{\lambda} \tag{5-1}$$

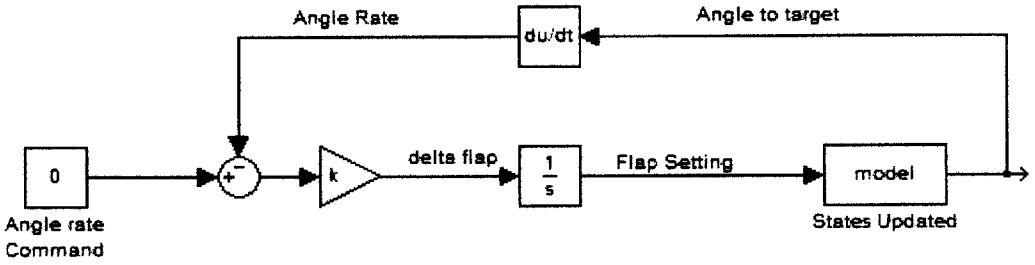


Figure 5-1 – Glide slope control

A positive angle rate corresponds to a target overshoot and calls for an increase in flap deflection, while a negative angle rate requires an increase in L/D by reducing the flaps. The incremental flap increase is proportional to $\dot{\lambda}$, which is then integrated so that the flap change is added to the current flap setting. When the angle rate is stabilized, the flaps remain at that setting.

Flaps can be deflected only so much before the parafoil stalls. To avoid situations where the canopy collapses, glide slope control is limited by the allowed flap deflection range.

Situations may arise where the angle rate cannot be stabilized, and other control methods must be used to avoid overshooting the target.

Figure 5-2 shows a simulated approach in 2D. The dashed lines are the paths the vehicle would take with either full or no flaps and therefore represent the boundaries of the reachable landing area from that starting point.

The angle-to-target, λ, is kept relatively constant throughout the descent. It does begin to diverge right before the landing since any small offset produces a large angle change as the target distance goes to zero. However, this is of little concern, since the vehicle can only make small adjustments, and it would already be so close to the target that it wouldn't have time to miss. In fact, the angle divergence might actually be useful since it could serve to signal that the ground is approaching. A large increase in the angle rate could be used to trigger a flared landing.

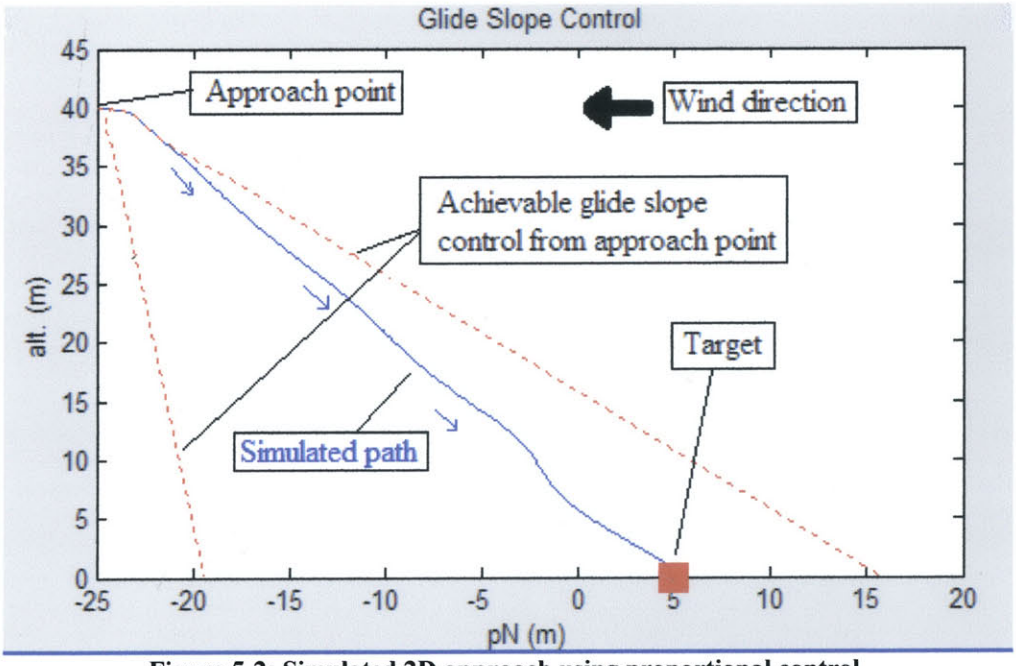


Figure 5-2: Simulated 2D approach using proportional control.

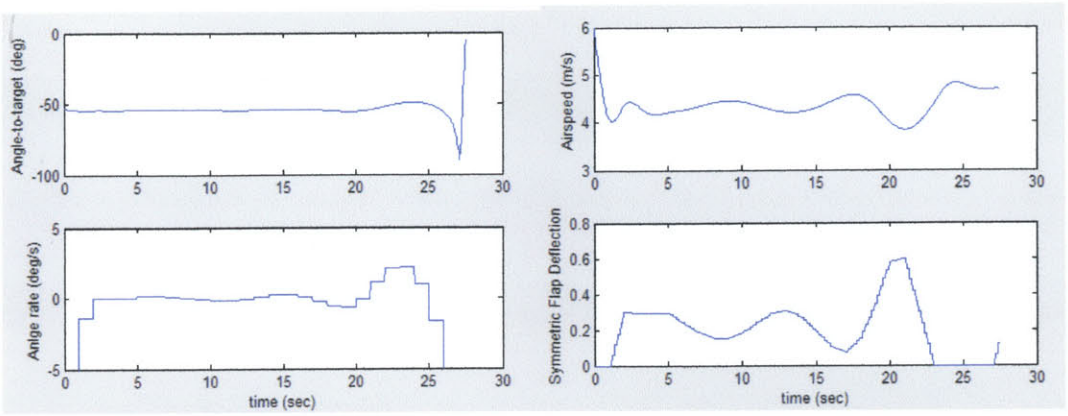


Figure 5-3: Angle-to-target and flap settings during 2D approach

5.3 - Lateral/directional Control

Apart from glide slope control, reaching the target requires lateral steering control. Two lateral/directional control methods are examined for the final approach. Much like the glide slope control, the first method used gains proportional to the lateral angle-to-target. Roll and sideslip angles are assumed to be small during the final approach, so the angle information can be taken directly from the camera pixel offsets.

$$\delta_a = K \cdot \dot{Pixel}_X \tag{5-2}$$

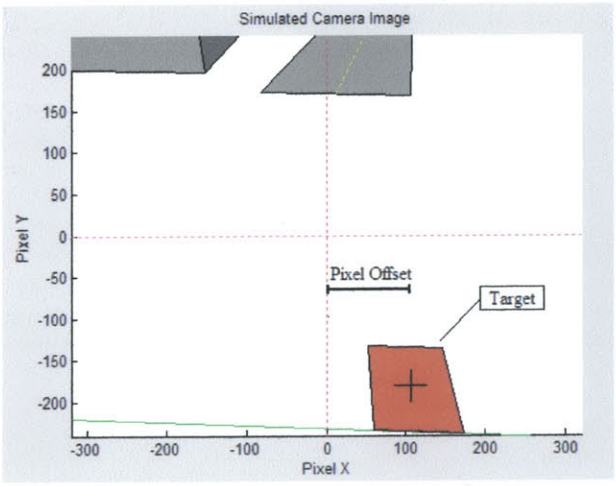


Figure 5-4: Camera image used in simulation. In this case, the target (red square) is offset by 100 pixels

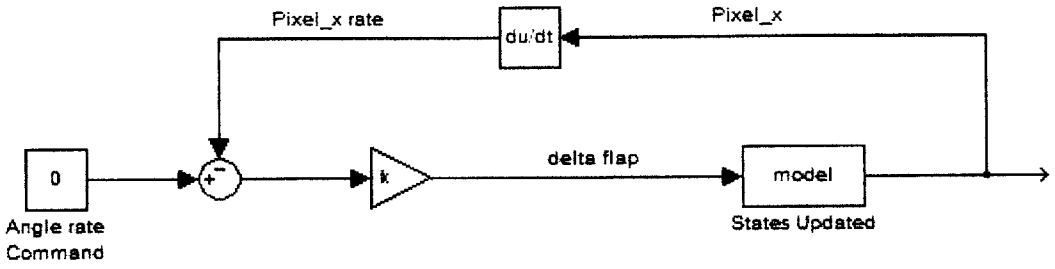


Figure 5-5: Diagram for lateral/directional proportional control using eq. 2

Using direct camera information eliminates the need to feedback yaw and roll rates for stability, as was the case for the guidance during the trajectory following. Figure 5-5 shows a simulated approach where the vehicle is initially offset from the target. The proportional controller guides the vehicle to the target successfully, but if an integrator is added, the response time can be improved. The guidance law becomes

$$\delta_a = K_1 \cdot \dot{Pixel}_X + K_2 \cdot Pixel_X \tag{5-3}$$

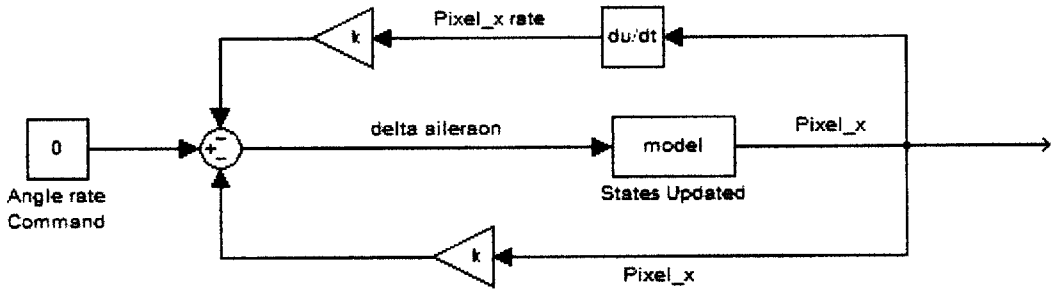


Figure 5-6: Lateral/directional control using eq. 3

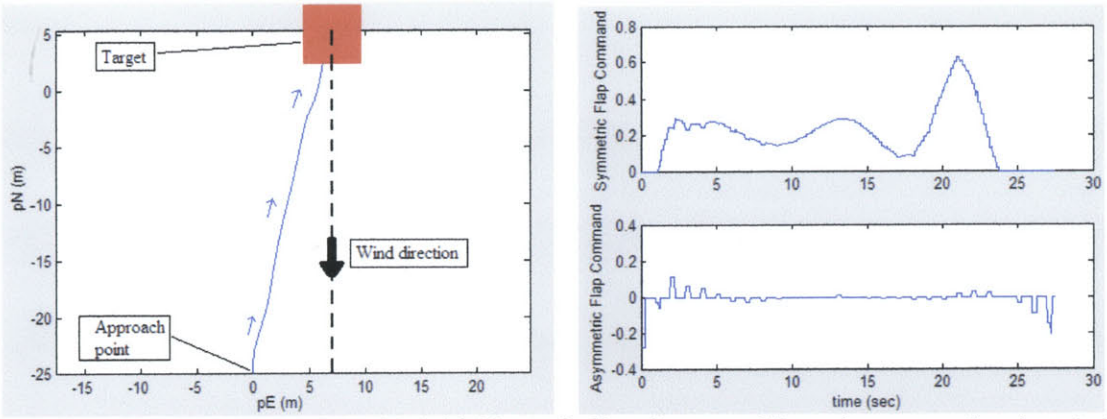


Figure 5-7: Simulated approach with initial lateral offset distance of 7m using proportional control

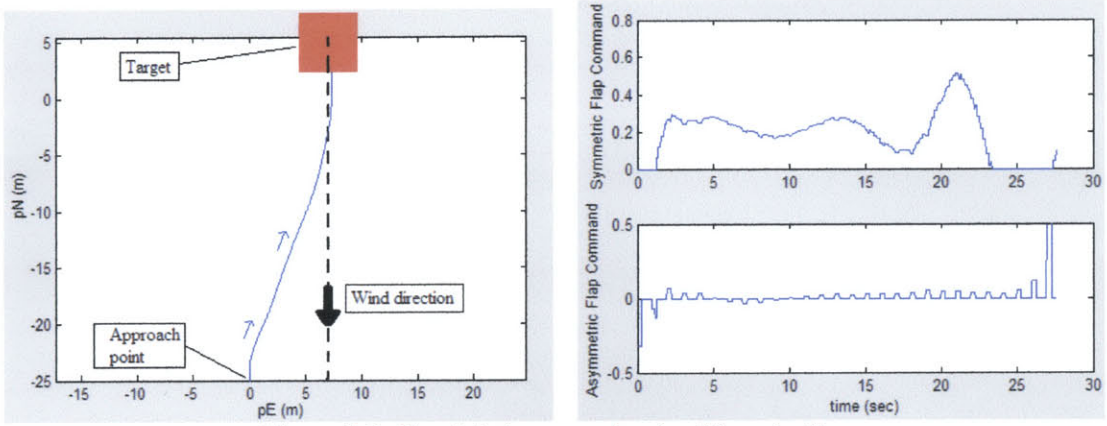


Figure 5-8: Simulated approach using PI controller

Using the same starting conditions, the simulation was run once more (figure 5-8), this time using the PI controller. Commanding the pixel offset to zero helps keep the vehicle pointed towards the target, which gives the controller a weathercock effect. The resulting response time is significantly improved. Also, by keeping the vehicle heading into the wind, the glide slope control is more effective.

With the slow dynamics and limited turn rates of parafoil systems, the faster controller is helpful when making flight path corrections. A major concern with the parafoil delivery system is that a strong gust disturbance could knock the vehicle off course late in its flight and not allow enough time to get back on target. Figure 5-9 shows a simulated

approach where a sudden wind change occurs about 8 seconds before landing. The presence of the integrator enables the vehicle to react faster to the wind change. This means it can handle disturbances later in flight and still have time to correct for them, making it more robust than the simple proportional controller.

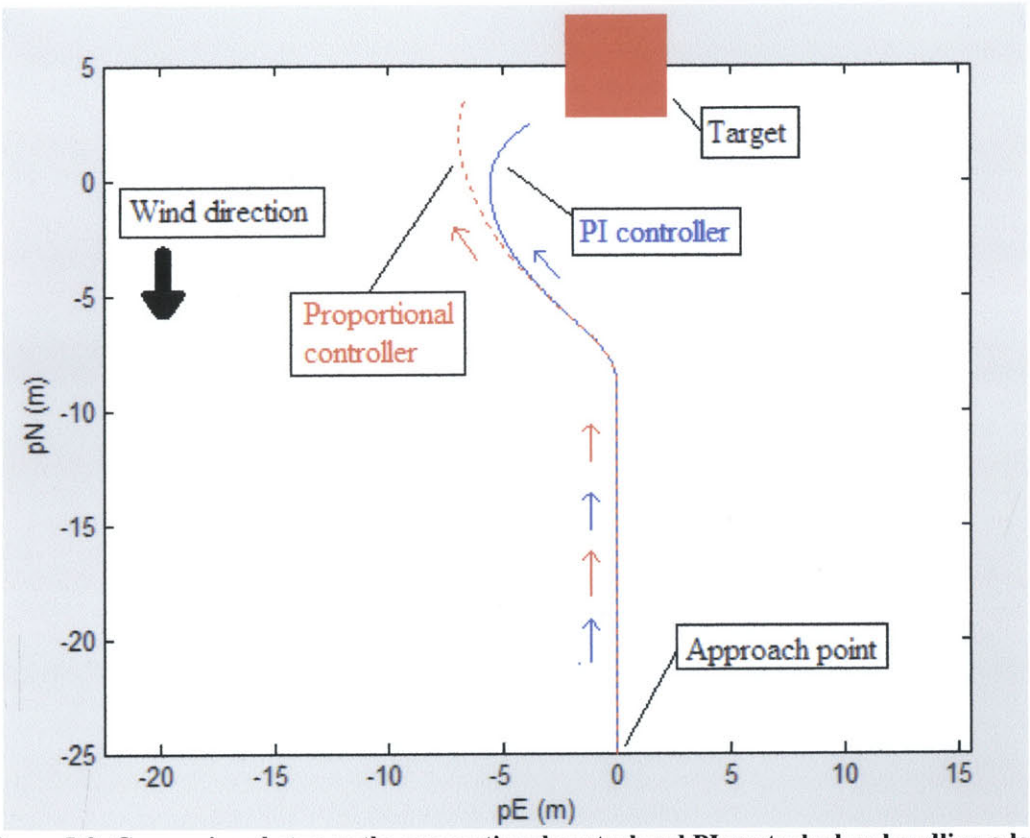


Figure 5-9: Comparison between the proportional control and PI control when handling a late disturbance

5.4 - Spiral Path

If the vehicle is ever in a situation where $\dot{\lambda}$ continues to increase after flaps are fully deflected, a modification can be made to the lateral/directional PI controller to prevent overshooting the target. Rather than keep the target centered, a commanded pixel offset is given. The result is a path that bellies out before approaching the target, thereby increasing the distance the vehicle travels to allow more time to bleed off altitude.

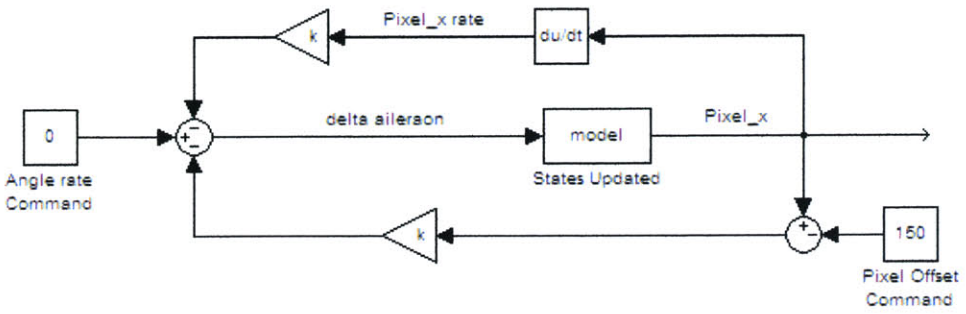


Figure 5-10: PI controller with nonzero pixel offset command

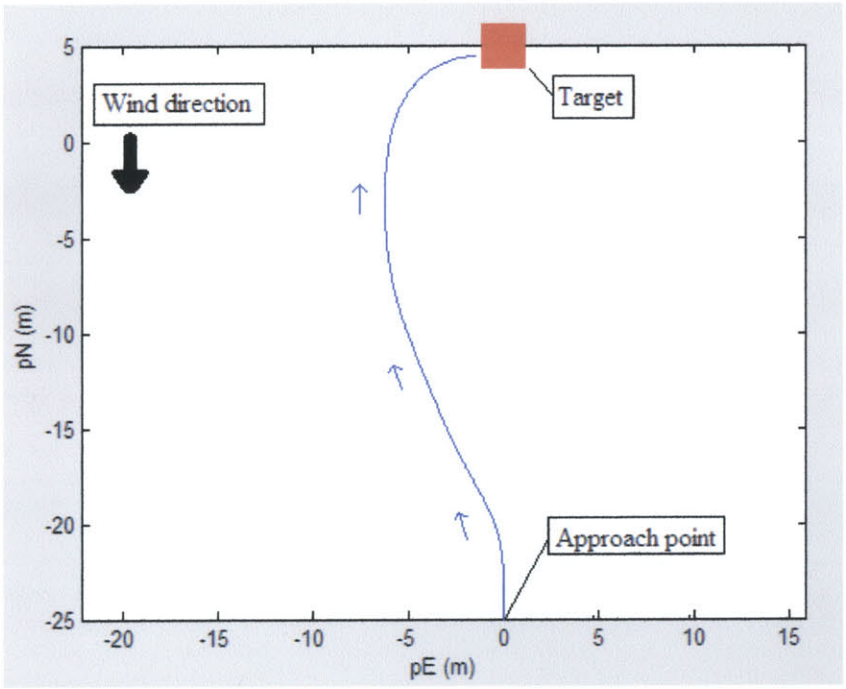


Figure 5-11: "Bellied" approach using controller shown in figure 5-10

Additional simulated approaches were performed with the starting altitude too high for the glide slope controller to keep λ constant. The lateral controller was modified to give a pixel offset command if the flaps were fully deflected and the angle rate had not stabilized. As the vehicle veered away from target, the target closing speed was reduced, along with $\dot{\lambda}$. When λ became constant, the controller could reduce the lateral offset angle and head more directly to the target.

It is hoped that a spiraled trajectory can be developed that reaches the target no matter how high the initial altitude is. If this were possible, the requirements on the trajectory guidance would be relaxed. Creating a controller that can spiral to a target accurately has proven difficult, however. As the vehicle approaches the target, the spiral gets tighter, requiring higher turning rates. In many cases, the turn rate needed to remain on the spiral exceeds the turning capabilities of the parafoil.

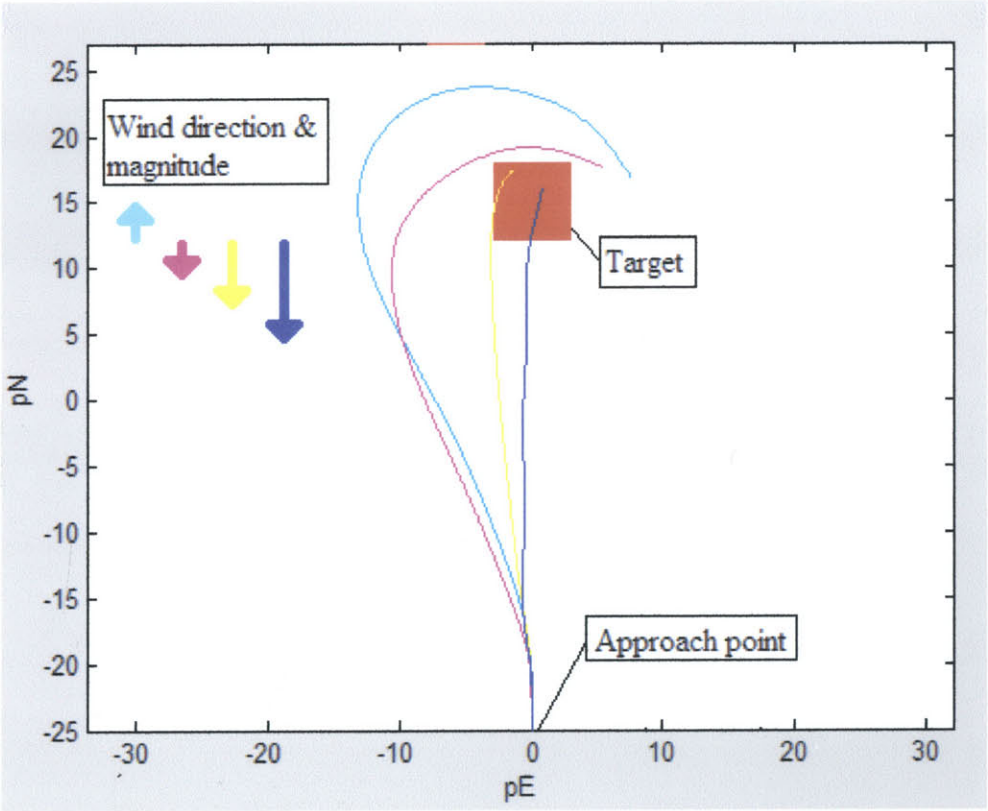


Figure 5-12: Simulated spiral trajectories for different wind conditions using guidance law described in eq. 4

Tests have shown there is promise for this spiral method. The delivery vehicle may not always arrive within 5m of the target, but its landing accuracy is far better than if the parafoil had simply overshot the target. Figure 5-12 shows examples of a spiral method for varying wind speeds. As the headwind is reduced, the glide slope control is degraded

and a larger commanded offset angle is required to avoid an overshoot. In this case, the commanded offset angle is proportional to $\dot{\lambda}$. The guidance law used for the simulations in figure 5-12 is

$$\delta_a = K_1 \cdot \dot{Pixel}_X + K_2\left(Pixel_X - K_3\dot{\lambda} \right) \tag{5-4}$$

where the commanded offset angle is represented by the $K_3\dot{\lambda}$ term.

Further improvements may be possible. Scheduling an increase in gains as the vehicle approaches the target would create faster responses to better handle the tightening spiral. And a wider camera viewing angle would enable the offset angle to be larger, which would allow for wider, more gradual spirals.

There are other methods worth considering. A "snaking" approach to the target would bleed away height much like the spiral method, and the vehicle would still be heading into the wind after the correct glide slope is reached. Another possible solution would be to calculate a new approach point and revert back to a trajectory planning algorithm. Both of these methods would not have to rely on the camera imagery, but they would require sufficient time before landing to be useful. Perhaps a combination of these error handling methods can be developed to handle a variety of circumstances.

6. Avionics and Estimation

The previous chapters discussed possible guidance and control methods, but in all cases, it was assumed that accurate telemetry data would be available. This section evaluates the sensors that are available for the delivery vehicle, and it also discusses their effectiveness when used in together to estimate the vehicle's states.

6.1 - Driving Factors

Many factors are considered when selecting the avionics package for the mission. The main priority is to obtain sensors sufficiently accurate to meet the demands of the precision delivery, but minimizing cost and complexity is also important. Establishing a ground network with many nodes requires numerous delivery vehicles, so the final avionics package should be as simple and affordable as possible for the system to be viable. And because there are two possible deployment configurations (ballistic and parafoil), compatibility in the avionics would help to reduce costs. Therefore, sensors should be shared between the two configurations whenever possible.

6.2 - Avionics components

6.2.1 - Vision-based System

Most autonomous drops using parafoils rely on GPS as the primary guidance sensor. However, the landing zones in theses instances are usually the size of football fields. To achieve our landing accuracy goal of 2-5m, far better telemetry is required than is available with GPS.

The limitations of a GPS-guided drop are due to the inherent errors in the vehicle position, as well as the target location. A vision system uses the relative position between the target and vehicle, essentially removing the bias errors associated with their absolute position. This greatly enhances the accuracy of the measurements, especially as the vehicle gets closer to the target. While the objective system may eventually call for a system with frame grabbing capabilities that can automatically track the target, MDPP is initially assuming that a human-in-the-loop vision system will be implemented. With video imagery sent down to the ground station, a human operator would have the ability to designate the target and give low frequency updates of its position.

As described in the previous chapter, the controllers used during the final approach relied heavily on the vision system. Without the camera's ability to identify the target, it would be impossible for the controller to steer the vehicle in the right direction.

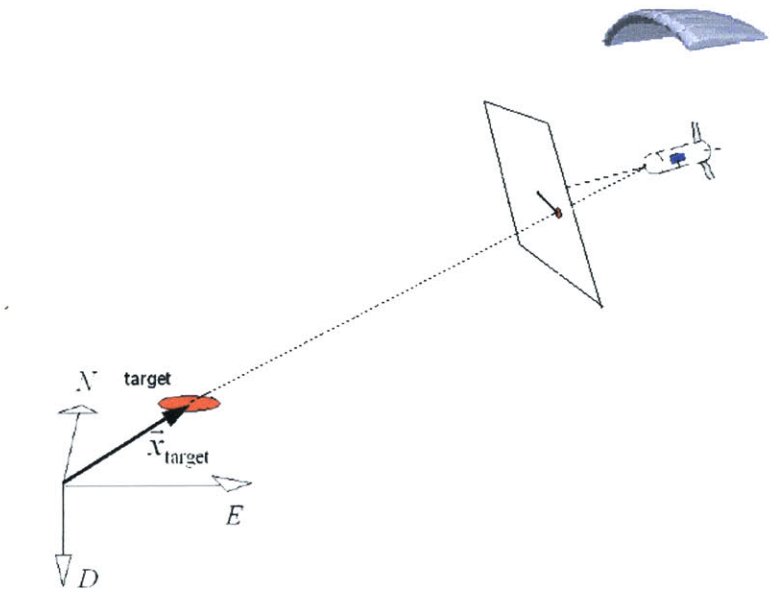


Figure 6-1 – Vision system

6.2.2 - IMU

The IMU is made up of 3 accelerometers and 3 rate gyros. Integrating the accelerations and angular rates can be used to estimate the vehicle's velocity and orientation, but estimation errors grow over time. An IMU is rarely used alone since state estimates would quickly become unreasonable for most applications. Other sensors are required to keep errors from growing. But an IMU is an important navigational aid; its high frequency measurements nicely compliment the lower frequency updates provided by other sensors.

6.2.3 - GPS

Alone, GPS cannot provide the required landing accuracy of 2-5m, but it will still be an important component of the avionics package. It is needed to update the vehicle position and velocity when the target is either far away or not in the camera's view. Following planned trajectories like the ones described in chapter 3 would be impossible without GPS, and the delivery system would be limited to simple drops where the target is always in view.

GPS will also be helpful during final approach since it can improve the angle-to-target rate estimates with velocity updates. These groundspeed measurements can also be compared to the pitot-static data to provide estimations of the current wind conditions.

6.2.4 - Magnetometer

A magnetometer is a 3-axis attitude measuring device. It works by sensing the magnetic field direction and comparing that with the known field strength at that location. Just as the GPS velocity information helps to correct for the drift errors associated with the IMU accelerometers, the magnetometer allows for accurate estimation of the rate gyro biases.

6.3 - Estimation Method

The continuous states estimated by the filter are the same ones propagated during the ballistic drop. With most of the avionics package shared between the two configurations, much of the filter algorithm is also the same. The differences lie in the measurement update portion of the filter, and only because GPS is required for the parafoil vehicle.

The filter used in this project keeps track of 19 states.

$$x = [pN \quad pE \quad pD \quad U \quad V \quad W \quad q_0 \quad q_1 \quad q_2 \quad q_3 \quad bias_P \quad bias_Q \quad bias_R \quad bias_{Ax} \quad bias_{Ay} \quad bias_{Az} \quad target_N \quad target_E \quad target_D]^T$$

The position states are estimated (pN, pE, pD), and the inertial velocities are with respect to the body axis. Quaternions are used instead of Euler angles to avoid the singularity that occurs when the vehicle is in a nose-down orientation. It is not really an issue with our parafoil system, but it will occur with the ballistic option.

The target position is also measured, but this is important only when the camera system is activated. Before that time, the target's position estimate is not updated and remains at its initial value before launch.

The remaining states are the biases from the IMU's accelerometers and rate gyros.

Table 6-1 General Extended Kalman Filter

$\dot{x} = f(x,t) + w \qquad w \sim N(0,Q)$ $z_k = h_k(x(t_k)) + v \qquad v \sim N(0,R_k)$
State Propagation $\dot{x} = f(\hat{x}(t),t)$ $\dot{P} = F(\hat{x}(t),t)P(t) + P(t)F(\hat{x}(t),t)^T + Q(t)$ $F(\hat{x}(t),t) = \left.\frac{\partial f(x(t),t)}{\partial x}\right\|_{x=\hat{x}(t)}$
Update Process $K = P(-)H^T(\hat{x}(-))\left[H(\hat{x}(-))P(-)H^T(\hat{x}(-)) + R\right]^{-1}$ $\hat{x}(+) = \hat{x}(-) + K(z - h(\hat{x}))$ $P(+) = (I - KH)P(-)(I - KH)^T + KRK^T$ $H(\hat{x}_k) = \left.\frac{\partial h(x(t_K))}{\partial x(t_K)}\right\|_{x(t_K)=\hat{x}_K}$

A complimentary filter is useful when combining the high data from the IMU with the other, lower frequency measurements. The accelerations and angular rates, along with their variances, are used to propagate the state estimates between the measurement updates from the other sensors. Because the states and measurements are not linearly related, an extended Kalman filter is used in the estimation process.

The following are the propagation relations for our 19 states.

$$\frac{d}{dt}\begin{bmatrix} p_N \\ p_E \\ p_D \end{bmatrix} = \begin{bmatrix} & C_{B2E} & \end{bmatrix}\begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} A_X + bias_{Ax} \\ A_Y + bias_{Ay} \\ A_Z + bias_{Az} \end{bmatrix} + \begin{bmatrix} & C_{E2B} & \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \begin{bmatrix} 0 & -(R+bias_R) & (Q+bias_Q) \\ (R+bias_R) & 0 & -(P+bias_P) \\ -(Q+bias_Q) & (P+bias_P) & 0 \end{bmatrix}\begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = -0.5\begin{bmatrix} 0 & (P+bias_P) & (Q+bias_Q) & (R+bias_R) \\ -(P+bias_P) & 0 & -(R+bias_R) & (Q+bias_Q) \\ -(Q+bias_Q) & (R+bias_R) & 0 & -(P+bias_P) \\ -(R+bias_R) & -(Q+bias_Q) & (P+bias_P) & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} bias_{A,3x1} \\ bias_{\Omega,3x1} \\ target_{p,3x1} \end{bmatrix} = \begin{bmatrix} 0_{3x1} \\ 0_{3x1} \\ 0_{3x1} \end{bmatrix}$$

The derivatives of the IMU biases and target position are zero since they can only be changed during the measurement update processes.

In table 1, $h(\hat{x})$ is the measurement matrix that relates the sensor data to the state vector. $h(\hat{x})$ is nonlinear in general, so the gradient with respect to the state vector must be computed to determine the measurement update gain. $h(\hat{x})$ for each of the three sensors is described in table 2.

Table 6-2 Sensor equations

Magnetometer	
$z_{Magnetometer} = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$ $\hat{z}_{Magnetometer} = h_{Magnetometer}(\hat{x}(t_K)) = [C_{e2b}] \begin{bmatrix} b_N \\ b_E \\ b_D \end{bmatrix}$	The values for b_N, b_E, and b_D are constants specific to the vehicle's location on the planet.
GPS	
$z_{GPS} = \begin{bmatrix} pN_{GPS} & pE_{GPS} & pD_{GPS} & vN_{GPS} & vE_{GPS} & vD_{GPS} \end{bmatrix}^T$ $\hat{z}_{GPS} = h_{GPS}(\hat{x}(t_K)) = \begin{bmatrix} I_{3x3} & 0_{3x3} \\ 0_{3x3} & C_{B2N} \end{bmatrix} \begin{bmatrix} \hat{p}_N \\ \hat{p}_E \\ \hat{p}_D \\ \hat{U} \\ \hat{V} \\ \hat{W} \end{bmatrix}$	
Camera	
$z_{Camera} = \begin{bmatrix} Pixel_X & Pixel_Y \end{bmatrix}^T$ $\begin{bmatrix} x_{Camera} \\ y_{Camera} \\ z_{Camera} \end{bmatrix} = [C_{N2B}] \begin{bmatrix} \hat{t\arg et}_N - \hat{p}_N \\ \hat{t\arg et}_E - \hat{p}_E \\ \hat{t\arg et}_D - \hat{p}_D \end{bmatrix} - \begin{bmatrix} r_{camera} \\ 0 \\ 0 \end{bmatrix}$ $\hat{z}_{Camera} = \begin{bmatrix} \hat{Pixel}_X \\ \hat{Pixel}_Y \end{bmatrix} = \begin{bmatrix} (y_{Camera}/x_{Camera})(640/2\tan(70/57.3)) \\ (z_{Camera}/x_{Camera})(480/2\tan(50/57.3)) \end{bmatrix}$	r_{camera} is the distance from the camera to the IMU unit. Assuming that both the camera and IMU lie on the center line of the vehicle, there is only one term for r_{camera}. The pixel resolution assumed to be 640x480, and the camera's horizontal and vertical field of view set at 70deg and 50deg respectively

6.4 - Estimation during Final Approach

Simulations were performed with different combinations of sensors during final approach. Without the camera, the angle-to-target measurements were poorly estimated and quickly became meaningless as the vehicle approached the landing area. When the camera system was used, the target was located, and the angle-to-target estimates were improved significantly.

Figure 6-2 shows the angle-to-target estimates during a simulation. Relying on these estimated angles, the vehicle was able to come within 5m of the target using the control methods described in chapter 5.

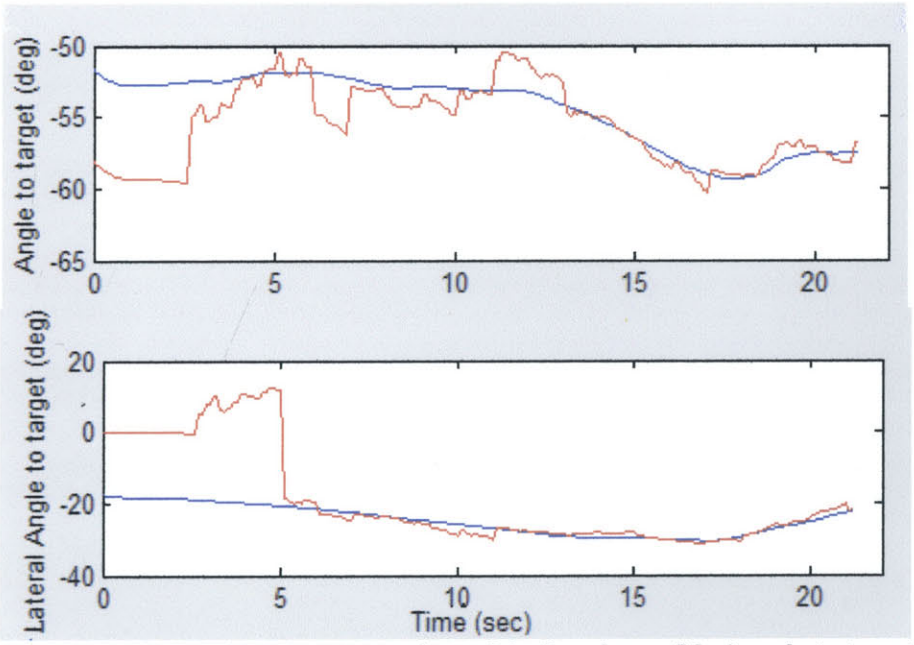


Figure 6-2: Comparisons between estimated (red) and true (blue) angle-to-target

The camera was activated after 5 seconds. Before that time, only the GPS and magnetometer were used (both assumed to be operating at 5Hz), and the angle estimates were not very accurate since the target position was not well known. When the camera measurements became available, there was an immediate improvement in the estimate. The errors in the measurements continued to decrease the closer the vehicle got to the target, which agrees with our intuition. By the end of the descent, the filter relied almost exclusively on the camera and ignored the GPS position measurements.

The figures below show how the position, velocity and attitude states were tracked during the simulated approach. There was significant error in the vehicle's position estimate during the descent, even after the camera was activated. It is interesting to note in figure 6-3 that the position errors lie mostly outside the 1σ lines (calculated as the square root of the diagonal terms in the covariance matrix). The 1σ lines are a measure of confidence in the estimates, therefore the errors should lie within those boundaries 67% of the time. The errors are with respect to the vehicle's true position, but as the camera's contribution increases, the 1σ lines begin to reflect the confidence in the relative position with respect to the target. Rather than update the vehicle's position with respect to a fixed target, the camera pixel measurements shifted the target so that the relative position agreed with the GPS measurements. In other words, the camera contributed little to the absolute position estimates, causing larger than expected errors for both the vehicle and target positions, yet was still able to accurately obtain the angle-to-target measurements.

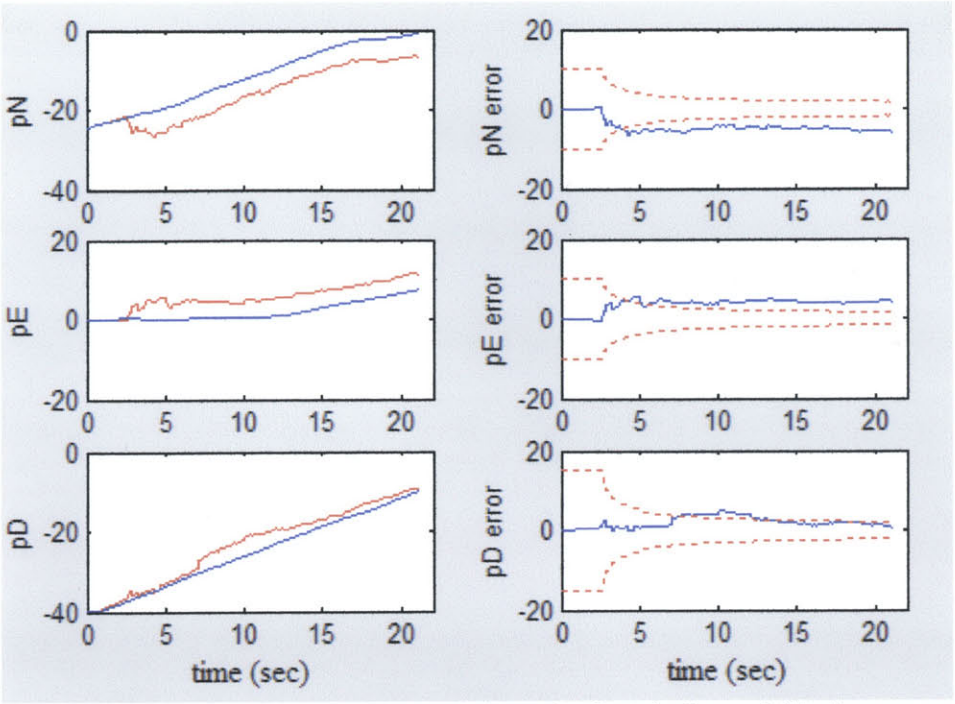


Figure 6-3: Vehicle position – estimated values (red), truth(blue)

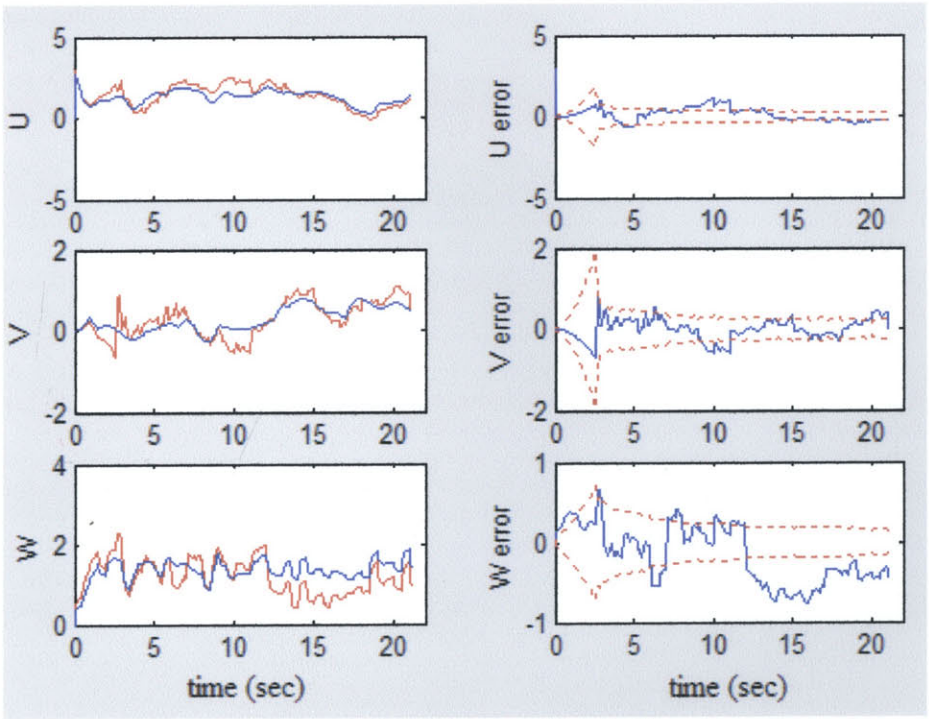


Figure 6-4: Vehicle inertial velocity

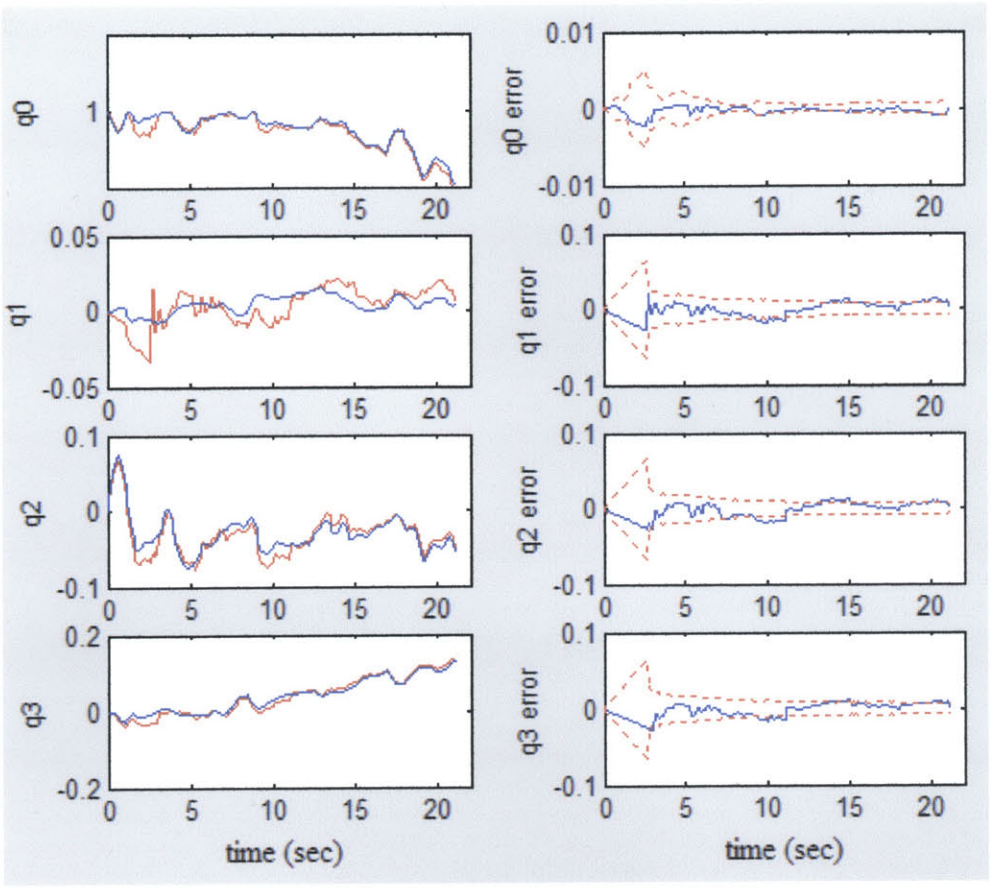


Figure 6-5: Vehicle attitude

7. Prototype PDV

7.1 – Prototype PDV

It is hoped that demonstrations will be conducted in the near future to validate the work outlined in this report. The next step is to build a prototype vehicle, however there will be many mechanical issues to address. A critical design challenge involves dealing with the high opening forces associated with the canopy unfurling. The parachute lines and their attachment points must be reinforced, and steps should be taken to avoid damaging the actuator linkages. It will also likely be necessary to employ reefing techniques to reduce the forces. Parachute canopies may be packaged so they inflate in stages so as to slow the opening process. By spreading the deceleration over a longer period, the stress on the lines is reduced. Another technique is to wrap rubber bands around the loops of line to help absorb energy. When the line is pulled, the rubber bands resist some tension before snapping off and allowing the main lines to full extend. And for prototype demonstrations, the forces may be further reduced by dropping the vehicle from lower altitudes and deploying the chute immediately.

Another important mechanical question involves the line actuation. There are two different types of servos, flapping and spooling, that can be used to deflect the trailing edge of the parafoil. Spooling servos allow an unlimited deflection range, and because the lines simply wind around a central spool, there is less risk in damaging the servo during the canopy opening. Flapping servos are likely to be faster and are simpler to employ, yet may require a larger space within the PDV.

The prototype PDV will hopefully be able to validate the aerodynamic model. As was discussed in chapter 2, the parafoil's dynamics are represented by stability derivatives. Their values were either derived analytically or were based on empirical data collected in previous parafoil experiments. However, most of the earlier experiments involved larger parafoil systems - little data was available on the behaviors of small parafoils. Improvements can therefore be made to the simulator by comparing the model with the actual flight data.

An output error methodology is described in figure 7-1. Commanded control inputs are recorded during flight tests, along with the vehicle's states. The same inputs are then run through the simulator so that the generated responses may be compared to the actual states. The stability parameters can be tuned to minimize the output error.

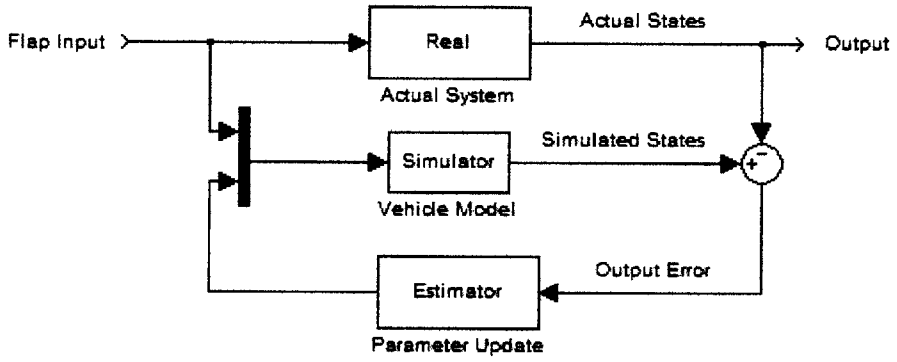


Figure 7-1 – Parameter estimation scheme

An iterative method has been suggested by [8] that utilizes maximum likelihood principles to update the parameters. By minimizing the negative log-likelihood function, a maximum likelihood estimate of the parameter vector can be obtained. The unknown

parameter vector can be updated by applying the Gauss-Newton method, iterating until it can no longer converge. Table 3 summarizes the relevant equations.

Table 7-1 Parameter Estimation Methodology

$\dot{x}(t) = f[x(t), u(t), \beta]$, $x(t_0) = x_0$ $y(t) = g[x(t), u(t), \beta]$ $z(t_k) = y(t_k) + v(t_k)$	x: state vector u: input vector y: observation vector z: output vector β:unknown system parameters
$\theta = [\beta^T; x_0{}^T; \Delta z^T; \Delta u^T]$	Unknown parameter vector
$J(\theta, R) = \frac{1}{2}\sum_{k=1}^{N}[z(t_k) - y(t_k)]^T R^{-1}[z(t_k) - y(t_k)] + \frac{N}{2}\ln\lvert R\rvert$	Negative log-likelihood function
$\theta_{i+1} = \theta_i + \Delta\theta$ $\Delta\theta = \left\{\sum_{k=1}^{N}[\frac{\partial y(t_k)}{\partial\theta}]^T R^{-1}[\frac{\partial y(t_k)}{\partial\theta}]\right\}^{-1}\left\{\sum_{k=1}^{N}[\frac{\partial y(t_k)}{\partial\theta}]^T R^{-1}[z(t_k) - y(t_k)]\right\}$	Iteration process

Demonstrating the precision delivery experiment will most likely be accomplished in incremental stages. "Dummy" drops testing the deployment and parachute unfurling mechanisms may be performed initially. R/C controlled parafoils can then be utilized to verify the vehicle's maneuvering capabilities.

Once the hardware is ready for implementation, the various control methods can be tested. Non-GPS versions can be used as long as the target is initially in view. From a low altitude, the drops will only demonstrate the final approach. The system will be less complicated without worrying about the trajectory planning or guidance. Only after the final approach system is verified should GPS be added to the avionics package. At that point the trajectory planning and guidance systems can be tested.

7.2 – Conclusion

The goal of this study was to determine the feasibility of using a small guided parafoil for precision delivery. Various issues associated with an autonomous parafoil were explored through the use of simulation. A 6-DOF aerodynamic model was created to provide an understanding of the parafoil dynamics and to help test the control methods. Different trajectory planning strategies were investigated, along with the guidance laws needed to follow them. To evaluate state estimation abilities, Kalman filters were implemented with an assortment of sensor packages. Final approach simulations were performed using the estimated states and consistently demonstrated landing accuracies within five meters.

It has been concluded that the parafoil PDV has the potential to develop into a viable option for precision delivery. While it may be more complicated than the ballistic method, the challenges are worth overcoming. Combined with the ballistic vehicle's capabilities, the potential benefits of the parafoil PDV may increase the effectiveness and flexibility of the precision delivery system.

References

[1] Frank L. Lewis, Brian L. Stevens. *Aircraft Control and Simulation*, A. Wiley-Interscience Publication

[2] Theodore W. Knack. *Parachute Recovery Systems Design Manual.* Para Publishing, Santa Barbara, California, 1992.

[3] C. Iacomini, C.J. Cerimele. *Longitudinal Aerodynamics from a Large Scale Parafoil Test Program.* AIAA, 1999

[4] C. Iacomini, C.J. Cerimele. *Lateral-directional Aerodynamics from a Large Scale Parafoil Test Program*, AIAA, 1999

[5] P. Mortaloni, O. Yakimenko, V. Dobrokhodov, R. Howard. *On the Development of a Six-degree-of-freedom Model of a Low Aspect-ratio Parafoil Delivery System*, AIAA, 2003

[6] C. Matos, R. Mahalingam, G. Ottinger, J. Klapper, R. Funk, N. Komerath. *Wind Tunnel Measurements of Parafoil Geometry and Aerodynamics.* AIAA, 1998

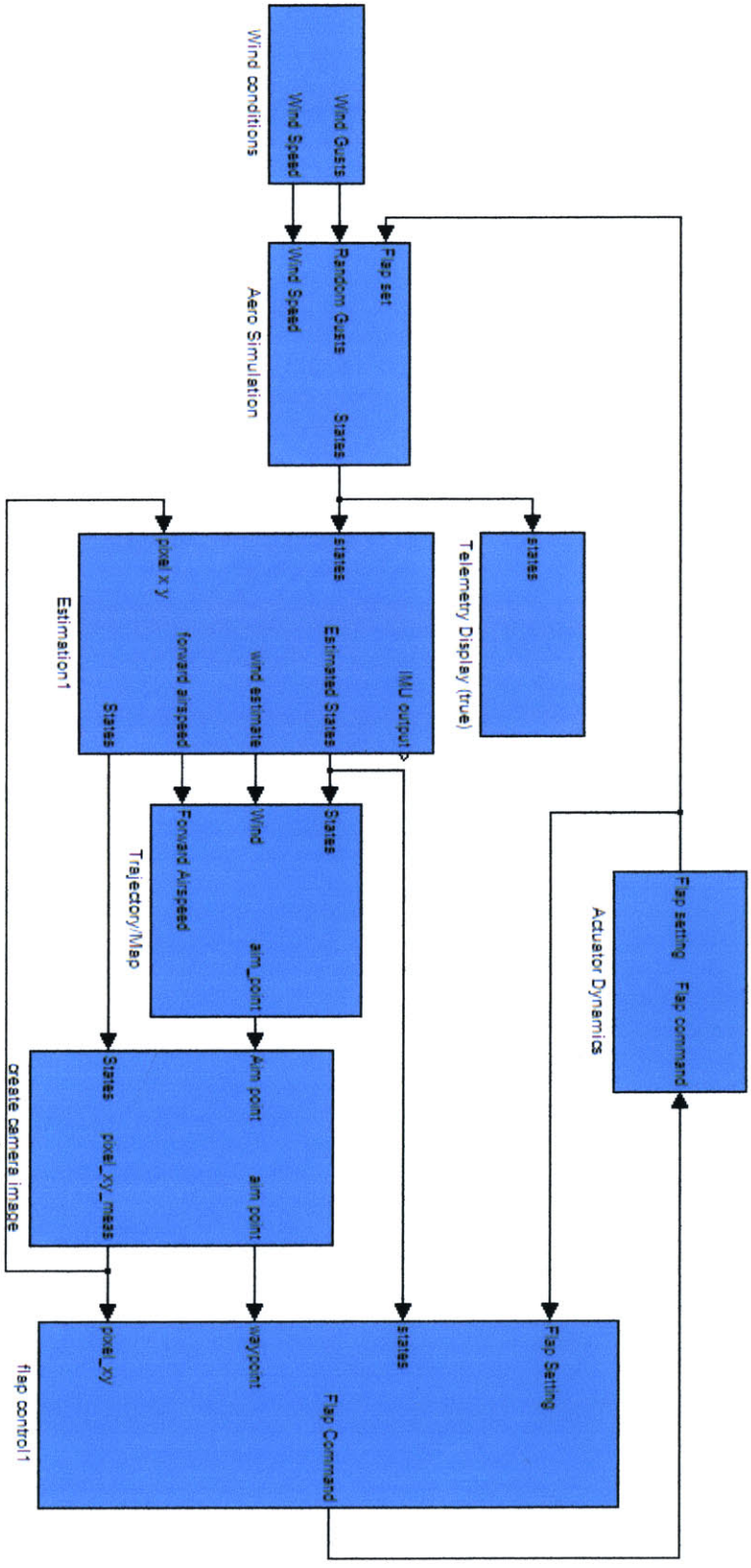
[7] T. Barrows. *Apparent Mass of Parafoils with Spanwise Camber*, AIAA, 2001

[8] R. V. Jategaonkar, F. Thielecke. *Aircraft Parameter Estimation – A Tool for the Development of Aerodynamic Databases.* Sadhana, Vol. 25, Part 2, April 2000, pp. 119-135

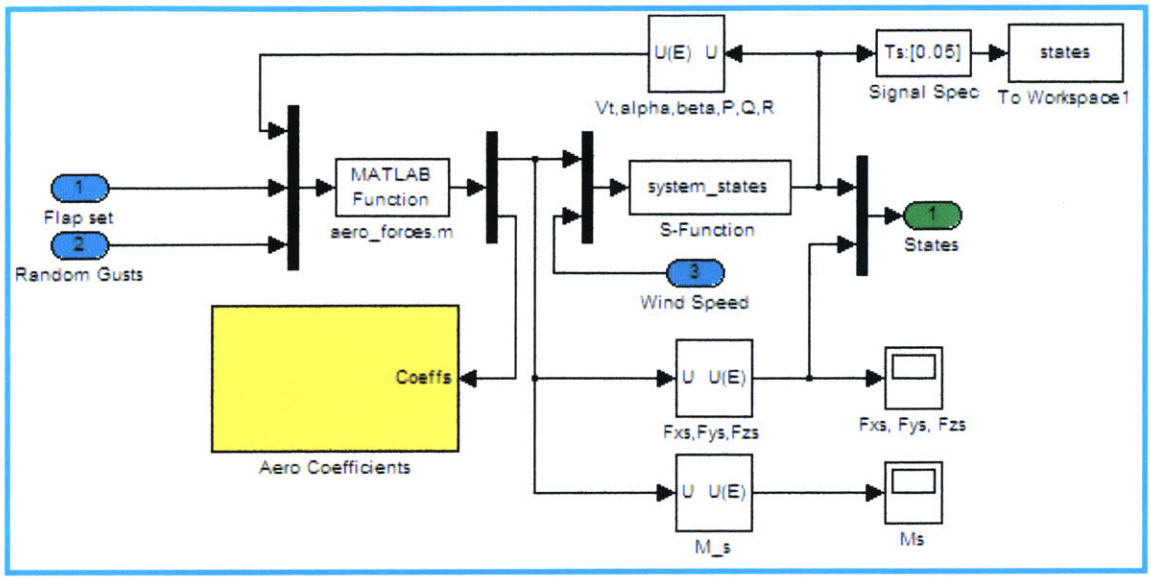
[9] Sanghyuk Park. Integration of Parent-Child Unmanned Air Vehicle Focusing on Control System Development, February 2001

[10] Lingard, S., "The Aerodynamics of Gliding Parachutes" ,AIAA-86-2427-CP, 9th Aerodynamic Decelerator and Balloon Technology Conference, October 1986

Appendix - Simulink Model



Aero Simulation



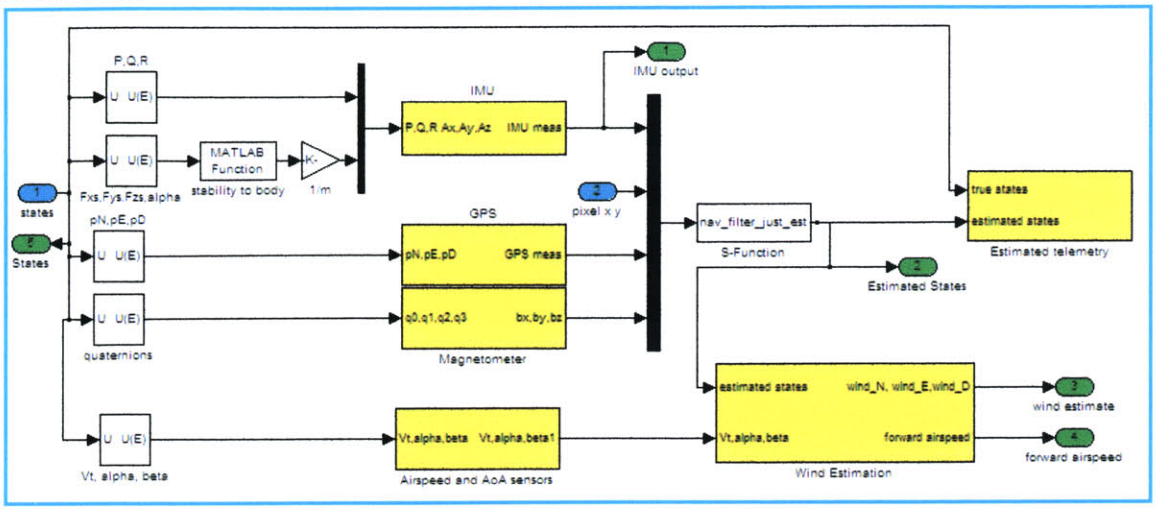
Summary: This module computes the aerodynamic forces based on the vehicle's current states, the flap settings, and the wind conditions. It then applies the forces to an S-function, which calculates the vehicle's response and updates the states. The updated state vector is then passed on as an output of the aero simulation block so it can be used by other modules. Some of the states (V_T, α, β, p, q, r) are fed back within the aero module for force calculation

Inputs:
Wind gusts – random noise defined by the user (3x1)
Wind speed – constant bias defined by the user (3x1)
Flap setting – Right and left flap setting. 0=no flaps, 1=full flaps. (2x1)

Outputs:
States vector (V_T, α, β, q0, q1, q2, q3, p, q, r, pN, pE, pD)
time, Fxs, Fys, Fzs are added to the output vector

Functions
aero_forces.m – Takes in V_T, α, β, p, q, r , left flap, right flap, wind gusts and computes forces in stability axis and moments about vehicle's CG. It uses stability derivative method and is based on the force and moment equations given in Chapter 1 of this report
system_states.m – Takes in the 6x1 force/moment vector and the wind speed(3x1) to compute accelerations based on the vehicle's inertia and current state. The derivative of the state vector is calculated, then integrated to get the update.

Estimation



Summary – The estimation module contains models for the various sensors available to the PDV. The state vector is fed into the sensor models along with bias and random errors, to create simulated sensor measurements. These sensor measurements are used by the navigation filter to estimate the state vector.

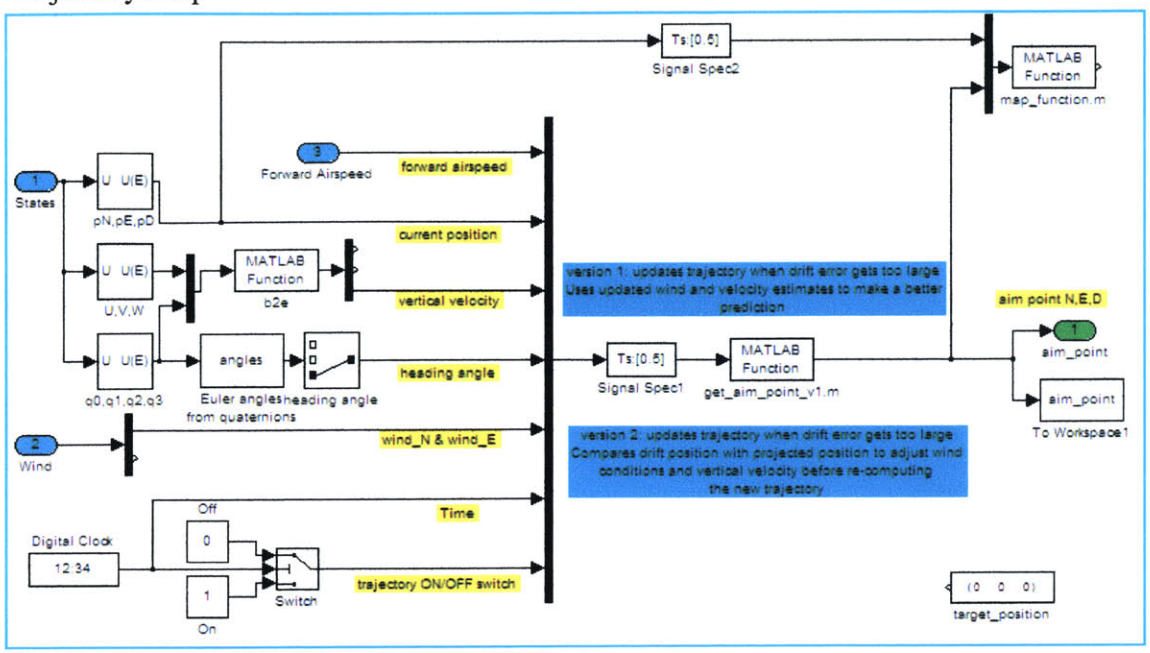
Inputs – State Vector, camera pixel measurements

Outputs – Estimated state vector (pN,pE,pD,u,v,w,q0,q2,q3,q4, accel biases, ang. Biases, target position)

Functions
IMU, GPS, magnetometer – they take in true telemetry, add error and noise based on the sensors' accuracy, and create simulated sensor data
estimation_filter.m – the extended Kalman filter that estimates the PDV's position, velocity and orientation. It also measures the target position. The high frequency IMU data is part of the continuous, state propagation portion of the filter, while the measurements from the other sensors make up the state update segment. Contained within the filter is the initial state estimate vector and covariance.
17 Inputs – IMU accelerations (6), camera pixel information(2), GPS position and velocity (6), magnetometer(3)
Output – Estimated state vector, diagonal terms of covariance vector

Note: While the true state vector keeps track of airspeed, AoA, and sideslip, the estimated state vector measures the inertial velocities u, v, w.

Trajectory/Map



Summary – The trajectory planning algorithm is contained within this module. The inputs are the PDV's position and velocity, the target location, and the wind profile. When turned on, a trajectory is created to guide the PDV to the approach point. It also computes the aim point – the point along the trajectory that the vehicle is currently aiming for – to be handed off to the guidance system. Also contained in this block is a map-like visualization.

Inputs – pN, pE, pD, U, V, W, q0, q1, q2, q3, wind_N, wind_E

Outputs – aim point coordinates(3x1)

Functions

get_aim_point_v1.m – This function takes in the vehicle's state information and the wind conditions to create a trajectory to follow. It updates the trajectory at regular time intervals, and it also picks the aim point for the guidance system.

get_aim_point_v2.m – Similar to the 1st version, except that it updates the trajectory parameters based on the PDV's drift error.

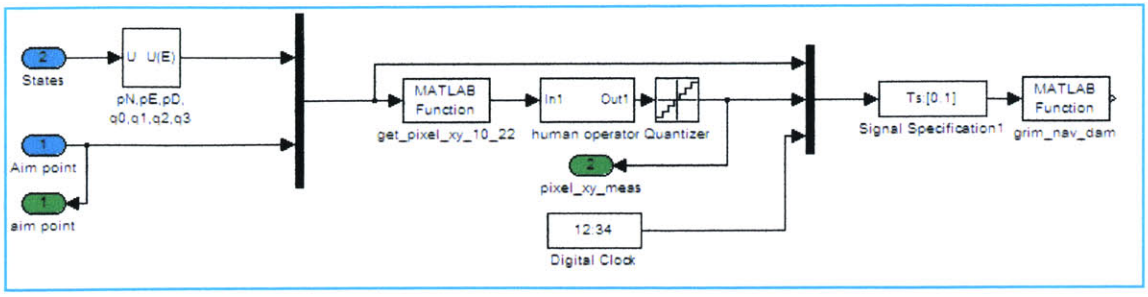
get_trajectory.m and get_trajectory_wind.m – a function called by get_aim_point that houses the actual trajectory algorithm. Two versions exist depending on the chosen trajectory type (earth-relative or wind-relative)

Notes:

- Currently, a simple timer switch is used to toggle between TRAJECTORY ON and TRAJECTORY OFF

- The trajectory update process can only occur for the first few seconds of the descent, before the PDV begins the second turning leg

Create Camera Image



Summary – This block is responsible for simulating the camera image. It models the human operator designating the target, and it obtains pixel information of the target location to be used by the GNC systems. This camera model was created by Sanghyuk Park for the ballistic PDV.

Inputs – pN, pE, pD, q0, q1, q2, q3, target coordinates(3x1)

Outputs – pixel_x, pixel_y

Functions
get_pixel_xy.m – Takes in the PDV's position and orientation, along with the target's position to calculate where the target would appear on the camera's field of view.
grim_nav_dam.m – Creates the actual visualization for the simulation. It paints the target as a red square.

Notes:
The visualization was designed for the ballistic drop, where the PDV is pointed at the target the whole time. In the parafoil's case, there are times when the target and other objects leave the FOV of the camera. This sometimes causes problems with the visualization. To avoid this issue, the camera visualization should only be used when the PDV is on final approach and the target is in view. The map display can be used during the trajectory following phase.
The map and camera visualizations cannot be run at the same time

Flap Control

Summary – Both the trajectory guidance and the final approach controllers are contained in this module. Depending on the phase of the descent, either the trajectory or final approach guidance is turned on.

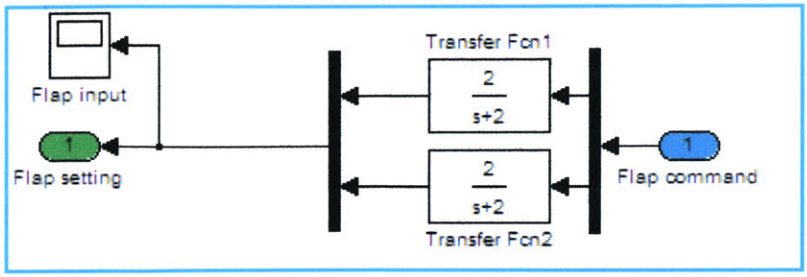
Inputs
The trajectory guidance portion requires the PDV's state vector and the current aim point along the trajectory.
The glide slope and lateral controllers for the final approach require the angle-to-target information

Output – Commanded flap setting (Lflap, Rflap)

Glide Slope Control – Uses angle-to-target rate to determine the necessary change in flaps. When the angle rate is zero, the flaps are left at their current position. Glide slope control is only used during final approach – a simple timer switch turns it on.
Lateral/directional Control – Lateral pixel offset from center is the only input. The lateral gains are proportional to the pixel offset rate. An integrator can also be added by commanding the lateral offset to zero.

Trajectory Guidance – The PDV's state vector and the aim point coordinates are used to determine the angle to target. Depending on the guidance method (linear or non-linear) lateral gains are determined and correspond to a commanded turn rate. Yaw and roll rates are fed back to provide damping.

Actuator Dynamics



Summary – This block models the actuator dynamics of the parafoil's control lines. Because of the non-rigid nature of the parafoil, it is not as responsive to control surface deflections as an aircraft would be. For the small parafoil modeled in this project, actuator time constants of 0.5 seconds are used.

Inputs: Commanded flap setting

Outputs: Simulated flap setting