ADAPTIVE CONTROL OF AUTONOMOUS AIRDROP SYSTEMS IN DEGRADED CONDITIONS

A Dissertation
Presented to
The Academic Faculty

by

Martin R. Cacan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2016

ADAPTIVE CONTROL OF AUTONOMOUS AIRDROP SYSTEMS IN DEGRADED CONDITIONS

Approved by:

Prof. Mark Costello, Advisor
School of Aerospace Engineering
School of Mechanical Engineering
Georgia Institute of Technology

Prof. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Prof. William Singhose
School of Mechanical Engineering
Georgia Institute of Technology

Prof. Eric Johnson
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Gregory Noetscher
Natick Soldier Research, Development
& Engineering Center
U.S. Army

Date Approved: 3 November 2016

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Dr. Mark Costello for consistent support and guidance. The values, goals, and leadership he provides to his research group have helped develop valuable engineering and life skills in both myself and lab mates. I would also like to thank my thesis committee for helping question my approach and offer direction on ways to make this the best work it could be. Additionally, I would like to acknowledge the funding provided by the Natick Soldier Research, Development, and Engineering Center and the interaction and support of Dr. Keith Bergeron, Dr. Gregory Noetscher, and Michael Shurtliff. It was extremely motivating to be connected to the people most interested in the development and impact of this technology.

I must also acknowledge that I wouldn't have completed this document without the support of many friends at Georgia Tech. I have learned that many graduate students have an innate bond formed by the desire to share information and learn new things, ideally over free food. I became a well-rounded engineer by engaging in the research of my peers and in return was aided significantly in the success of my own. In particular, I want to thank Edward Scheuermann and Michael Ward for the assistance they provided towards the development of this work. Additionally, I want to thank the current and former members of Dr. Costello's research group who have provided guidance and helped me convert series of rough ideas into a successful project.

I would like to thank the support of my closest friend and family members. Theresa Wilks helped ensure my mental sanity while constantly pushing me to be productive and graduate - all while pursuing her own PhD at Georgia Tech. I am grateful to my

parents who have provided endless forms of support. From helping me pay for airfare to go home to sound life advice, they have been pivotal in my academic success. The same is true for my two loving sisters who have provided encouragement and comic relief my whole life. Last but not least, I want to thank my niece Paityn whose interest in science allowed an enthusiastic uncle to drag her through coding projects and science experiments that resulted in successes and failures, both of which are important things to learn.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Autonomous airdrop systems exhibit significant improvements over their unguided counterparts due to the addition of control mechanisms and real-time feedback. Increased landing accuracy comes with a high dependence on accurate sensor measurements (primarily GPS) and a consistent nominal system and environment. Unfortunately, airdrop systems represent a highly uncertain class of aerial vehicles where operation at off-nominal, degraded conditions is the norm and not the exception. Degraded conditions are caused by numerous effects including: parafoil rigging changes caused by canopy opening shock, parafoil damage during canopy inflation, parafoil canopy collapse, payload mass and inertial loading imbalances, human rigging errors, etc. Additionally, the loss of GPS feedback due to environmental causes (urban or canyon environments blocking the signal or causing multi-pathing issues) and active GPS denial by tech savvy adversaries present a second classification of degraded flight. These issues must be addressed to aid the continued expansion of the parafoil technology for delivery and re-entry purposes.

This work applies two solutions to overcome the problems of physical degraded flight conditions and loss of GPS feedback. A highly adaptive control law is embedded at the core of the new guidance, navigation and control (GNC) algorithm to identify system dynamics and control sensitivity through the use of a discrete nonlinear Hammerstein dynamic model. All model parameters are estimated in-flight to maintain a high level of landing accuracy under a large range of degraded flight conditions. In addition, an analysis into the feasibility of radio frequency (RF) beacons as a redundant feedback system to GPS is analyzed. Novel GNC algorithms are introduced to handle the limited feedback that RF beacons provide in comparison to the rich data embedded in GPS. Performance of the proposed autonomous algorithms are tested through rigorous simulation of a validated flight dynamic model and experimental testing on a small scale autonomous parafoil and payload system.

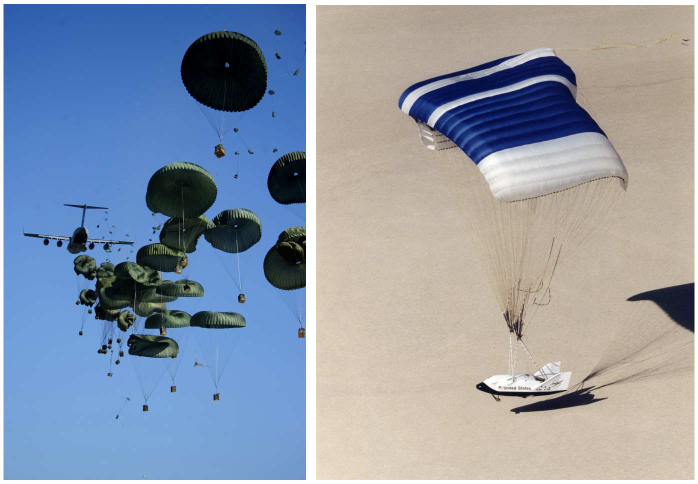
CHAPTER I

INTRODUCTION

Airdrop technology has revolutionized the delivery of supplies for both military and humanitarian operations. Dropped from carrier aircraft above the desired target, supplies can be delivered to disaster locations, contested war zones, or generally difficult to reach locations in a short period of time. The bulk of cargo delivered in theater uses ballistic (unguided) parachutes dropped en masse from low flying C-17 and C-130 aircraft. The calculated aerial release point (CARP) is based on estimated atmospheric wind conditions from the ground to aircraft, carrier vehicle velocity, and nominal descent rate of the airdrop system [6]. Accuracy is primarily ensured through low altitude releases as round parachutes are drag dominant devices that are susceptible to being blown off target due to errors in the atmospheric wind estimate. Figure 1.1(a) provides an example of an unguided airdrop delivery where the U.S. Air Force supplied nearly 70,000 lbs of food and water to victims of the earthquake in Haiti in 2010 [1]. In military operations, there is great risk involved with flying low enough to ensure accurate delivery of unguided systems. The development of autonomously guided and controlled airdrop vehicles addresses the need for both high release altitude and high landing precision capabilities.

1.1 Overview of Precision Airdrop

The next era of airdrop systems use sensor feedback and onboard actuators in conjunction with guidance, navigation, and control (GNC) algorithms to actively steer the aerial vehicle to the desired impact point (IP). The field of guided airdrop systems was originally studied by Kane in the early 1960s using a modified ballistic parachute [7] and then in the late 1960s by Knapp and Barton who used a directional



(a) (b)

Figure 1.1: Examples of different airdrop technologies: (a) Mass low altitude deployment of ballistic airdrop systems [1]; (b) NASA X-38 test vehicle [2].

radio frequency beacon for the recovery of a 150 pound sounding rocket nose cone [8]. In the 1970s, Goodrick et al. [9–11] proposed an improved homing technique and analyzed current state of the art hardware to enable a field implementable guided system. When military grade GPS signal quality was made available to public and private enterprises in 2000, more advanced GNC schemes could be developed given position, velocity, and time (PVT) data available for feedback control. GNC development was accelerated starting in 1997 with the precision airdrop system (PADS) sponsored by the U.S. Army and Air Force. The goal of the project (and that of its successor [12]) was aimed at enabling high altitude, accurate delivery of cargo up to 10,000 lbs [13]. Similar research was conducted by the National Aeronautics and Space Administration (NASA) for an emergency crew re-entry vehicle for the international space station. Development of the X-38 vehicle shown in Figure 1.1(b) ran from 1999 to 2003, and among several things, was used to study the construction and safety of canopies ranging in size from 750 ft^2 to 7500 ft^2 [14–16]. These government projects initiated a spark in GPS guided parafoils that has led to many advances in GNC algorithms and system controllability for improved landing accuracy [17–23].

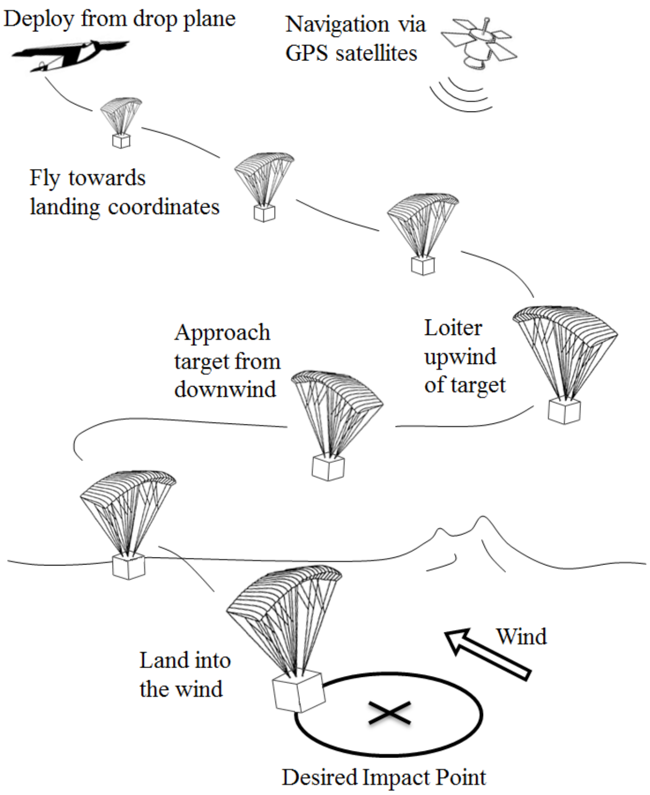

Deploy from drop plane
Navigation via GPS satellites
Fly towards landing coordinates
Approach target from downwind
Loiter upwind of target
Wind
Land into the wind
Desired Impact Point

Figure 1.2: Typical flight profile for modern precision airdrop systems.

While details of each approach differ, each follows the general methodology outlined in Figure 1.2. Using similar CARP estimation methods as unguided systems, guided airdrop vehicles are released upwind of the target and use its control authority to overcome errors in the *a priori* atmospheric wind estimates to accurately reach the IP. After deployment, the vehicle flies towards the target and maintains local proximity to the target upwind of the target until sufficient altitude has been lost to land at the IP.

Current airdrop research focuses on advancing one or more of the following elements: accuracy, cargo survivability, and cost. While some projects approach these three elements through the integration of control authority and GNC algorithms into conventionally ballistic round parachutes [24, 25], most research focuses on ram-air parafoils which generate both lift and drag. Ram-air parafoils use openings at the front of the airfoil to inflate with static pressure to form an arc anhedral wing with

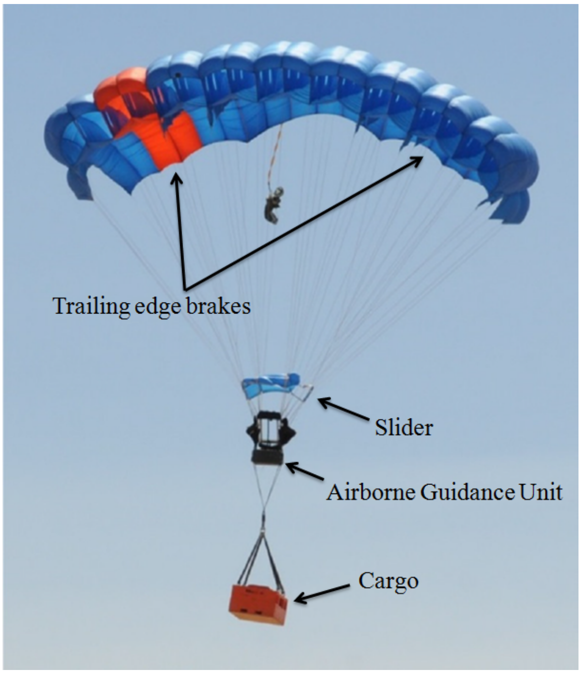


Figure 1.3: Modern guided, ram-air parafoil.

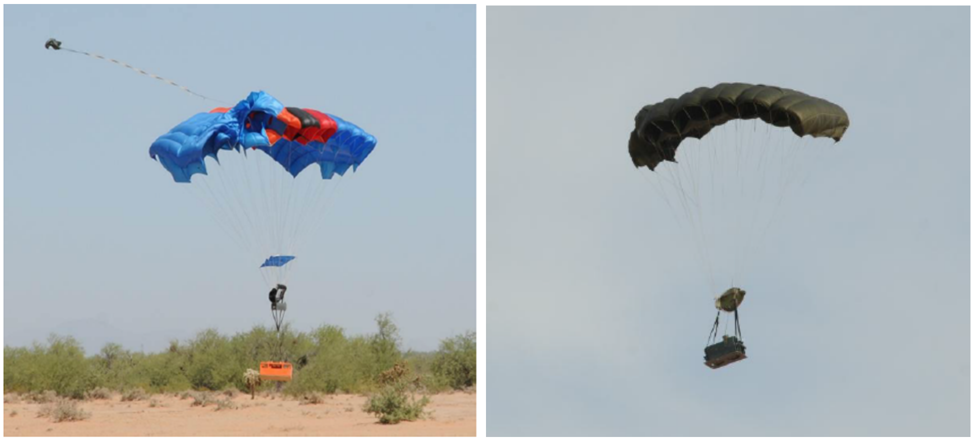
a classic airfoil cross section design. While more expensive to manufacture than ballistic round chutes, parafoils benefit strongly from their ability to penetrate the wind in addition to both lateral and longitudinal control authority. Longitudinal control for parafoil and payload systems implies the ability to change the ratio of forward airspeed to descent rate, termed glide slope, and reaches 4:1 for some fielded systems. Figure 1.3 shows the configuration of a modern precision airdrop vehicle. Actuators, sensors, and processing power are located in the airborne guidance unit (AGU). General rigging lines connect the canopy to the AGU with control lines connecting the actuators to the trailing edge brakes. The payload is suspended below the AGU through a gimbal joint to ensure rotation motion is not passed to the rest of the structure. Inflation of the low porosity rip-stop nylon canopy is controlled by the addition of a mini drag chute (slider) attached to the riser lines. The slider passively forces the canopy to inflate in a center outward fashion, slowing the inflation event and mitigating high shock openings of the canopy.

Lateral and longitudinal control authority for precision airdrop systems is conventionally achieved through symmetric and asymmetric deflection of the trailing edge brakes [23, 26], but weight shift [27] and upper surface canopy spoilers [28] have also been analyzed. Pure longitudinal control can be achieved through variable canopy incidence angle by changing the aerodynamic angle of attack [22, 29–31]. This method is more effective than symmetric brake control but difficult to implement on large scale systems due to the required actuation force.

1.2 Susceptibility to Degraded Conditions

Almost all guided airdrop systems rely on GPS and a variety of onboard sensors for feedback for control algorithms to steer the system to the target. These control algorithms are designed to control mathematically derived dynamic models which predict the actual flight characteristics of the system. Development of these guided systems require extensive system identification in which entire flights are conducted with open loop commands chosen to characterize aerodynamic parameters and control sensitivity mappings. However, as a result of the very nature of parafoils as a flexible and redeployable system, they are severely prone to changes in flight dynamics which decreases the ability of the control algorithm to steer the system accurately. Two examples of guided parafoil systems which had canopy cells rip during deployment are presented in Figure 1.4 and classify just one of the failure modes experienced by airdrop systems.

In the initial development of guided parafoil technology for the NASA X-38 project, damage mitigation was a priority concern as human life was in the balance. However, as the technology transitioned to military applications for cargo delivery, damage mitigation was scaled back to reduce manufacturing cost for mass production [15]. While the aforementioned case of torn canopy cells appears to be a relatively



(a) (b)

Figure 1.4: Autonomous airdrop system with ripped (a) upper surface canopy and (b) right canopy cells caused by high shock loading during inflation [3].

extreme failure, the aerial vehicle is still controllable but exhibits flight dynamics significantly different from the modeled behavior which causes the GNC algorithm to be unable to steer the vehicle to the target. Other identified failure modes include the slider getting caught preventing full inflation of the canopy, tension knots in the canopy risers changing the rigging geometry, human induced errors, and changes in fabric porosity due to age, use, and ultraviolet light exposure. In this context, human error is primarily concerned with the rigging and connection of the parafoil to the AGU. It is not uncommon for the control lines to get switched causing a control reversal or be spooled backwards onto the actuator resulting in a significantly different control mappings. These failure modes can cause slight to extreme variation of the actual system dynamics from the expected model dynamics rendering non-adaptive control algorithms ineffective at controlling the guided airdrop system.

There is significant work in the literature relating to the compensation of unmodelled dynamics to the control architecture of many aerial vehicles or flexible structures. Adaptive neural networks are commonly used to characterize and reject the error between measured and expected dynamic response of a vehicle [19, 20, 32, 33].

Using the trained neural network, control effort is modified to improve system performance. Additionally, several articles have studied onboard estimation of vehicle airspeed [22, 34–36]. An incorrect vehicle airspeed can easily destabilize the navigation state estimator as airspeed, horizontal atmospheric winds, and heading are all closely coupled. Changes in airspeed can be induced by variations in canopy construction alone and more severely impacted by canopy damage or incorrect cargo weight. Adaptive turn rate control was studied [3, 19, 20] in order to improve the tracking ability of the lateral controller or overcome significant turn rate biases exhibited by the physical system. However, the current state of the art algorithms are still unable to handle the high level of dynamic uncertainty that these vehicles exhibit. A summary of key deviations in nominal flight dynamics and current solutions in the airdrop literature are outlined in Table 1.1. Potential sources of physical degradation that could induce such flight perturbations are also characterized and listed in the legend at the bottom of Table 1.1. Note that while several modes have solutions, they are not necessarily generic enough to combine to generate a large scale solution to all forms of damage.

An additional dilemma for guided airdrop systems is their extreme dependence on GPS information for feedback control. This can limit airdrop applications as a robust and reliable GPS signal is not always available in certain operational scenarios. Urban and natural canyons can induce multipath propagation and obscured satellite issues that can degrade GPS signal quality and deny the ability of a receiver to achieve GPS lock. Additionally, technology that allows adversaries to overpower the extremely low level signal strength (L1, L2, and L5 signals employ received signal strengths of -157.7 dBw, -160 dBw and -154 dBw, respectively [37]) by swamping a GPS receiver in noise is becoming more common. As a result, future missions have a significant need for a secondary or redundant feedback source to reduce dependence on GPS information. To accomplish this, the airdrop system must be augmented with additional sensor

Table 1.1: Examples of how deviations in nominal flight dynamics manifest from modes of physical degradation.

Deviations in Flight Dynamics	Possible Source of Deviation	Solutions Proposed in Literature
Change in vehicle airspeed	A,B,C,D,F,G	[22, 34–36]
Change in control sensitivity		
(including turn bias)	A,B,C,D,E,G	[3, 19, 20]
Unidirectional Control Authority	A,D,E	—
Control reversal	D	—

Legend:
A: Torn canopy cells
B: Partially collapsed canopy, not fully inflated canopy
C: Change in canopy geometry due to tension knots or broken rigging lines
D: Human rigging error
E: Faulty actuator, broken control line
F: Aged canopy material
G: Incorrect payload weight

types (e.g., vision, digital terrain data, star mapping, etc.) to provide the typical types of data expected of a GPS receiver.

Research in the field of aerospace guidance systems has identified multiple solutions to the GPS blackout problem, summarized in Table 1.2. The most commonly implemented is vision based navigation which uses landmark detection, image recognition, or field of view motion to construct an estimate of the vehicle position and velocity [38–43]. Vision algorithms are often run in conjunction with dead reckoning methods based on the integration of inertial measurement units (IMUs) [44]. The fusing of visual and IMU data requires careful calibration of sensors making self-calibration and correction methods an active area of research [45]. Additionally, aerial image matching compares real time images with a bank of preloaded satellite images of the drop zone for position estimates [46]. While these technologies are viable for use in guided airdrop, several key limitations currently exist that prevent field implementation. Vision feedback is limited to daytime operations with relatively

Table 1.2: Potential feedback sources for GPS denied airdrop systems

Sensor	Main Advantages	Main Disadvantages
Commercial IMU	Low cost	Poor signal quality
Tactical IMU	Low signal to noise ratio	Weight, cost
Barometer	Accurate altitude measurements, low cost	Requires calibration
Vision System	Advanced algorithms enable position and velocity feedback	Requires high quality camera, computationally intensive, typically requires with IMU
RF Beacon (Range)	Range and radial velocity enables better observability in comparison to directional beacon	Requires setup at IP, broadcasts location, cost
RF Beacon (Radiolocation)	Low cost	Requires setup at IP, broadcasts location

minimal clouds, smoke, or fog in the airspace above the drop zone (DZ). Additionally, the performance limitations of commercial grade IMUs and the weight and cost limitations of tactical grade IMUs make their integration infeasible.

An alternative method is through radio frequency (RF) beacons which can record range measurements based on time of flight [47–49] or relative heading to the beacon via radiolocation methods [50, 51]. This field has seen a large amount of work associated with low cost and short range sensors for localization of robots in indoor environments [52, 53]. These strategies often focus on using a large set of beacons and advanced estimation methods to handle poor accuracy or limited range of the beacons. Long range beacons like the original LORAN system [54] provide global coverage but very poor resolution. Recent advances in beacon technology provide medium range beacon systems that provide high precision range and velocity measurements over 1km [55, 56]. Additionally, beacons placed at the target can transmit signals at sufficient power to decrease their susceptibility to signal jamming. This provides an avenue to reanalyze and expand beyond early guided airdrop solutions presented by Kane et al. [7], Barton et al. [8], Goodrick et al. [9–11].

Autonomous Underwater Vehicles (AUV) have made particularly strong use of beacons. Potential solutions exist for the marine application problem through use of a range measurements from a single location [40, 47, 49, 57]. Radio beacon signals are commonly blended with other sensor data streams (such as a barometric pressure sensor for altitude measurements) to calculate a navigation solution [58]. The analogy of UAV position estimation in an unknown underwater current and guided airdrop is striking giving RF beacon technology strong promise to become relevant to the airdrop community again.

1.3 Contributions of the Thesis

The overarching goal of this thesis is to expand upon the achievable flight conditions in which a precision airdrop system can be used to accurately deliver supplies to a location. The field currently shows a high dependence on nominal flight conditions and GPS lock for feedback which are far from guaranteed in real world applications. Many failure modes and various forms of GPS loss or denial exposes a general lack of robustness to the conditions in which airdrop systems can accurately reach the target. This thesis proposes a two part method to overcome these issues: an adaptive control algorithm and the inclusion of a radio frequency beacon to airdrop operations.

1. Adaptive Control: Development of a highly adaptive GNC algorithm capable of in-flight system identification to guarantee accurate landing capabilities of a controllable precision airdrop system subject to degraded flight conditions.

2. Radio Frequency Beacon: Integration of a single radio frequency beacon broadcasting from the desired impact point to provide a navigation solution to GPS denied vehicles and ensure landing accuracy given limited feedback.

In order to satisfy the above objectives, a simulation and experimental based approach is taken. A sophisticated simulation environment was developed including a 6 degree of freedom parafoil and payload model and a high fidelity wind field model that captures relevant low and high frequency variations in the air mass. Details of these models are presented in Chapter II. A conventional, GPS based GNC is presented in Chapter III from which portions of the GNC algorithms proposed in this work are based upon. The simulation model is based on a small-scale, custom built parafoil and payload system discussed in Chapter IV. Chapters V and VI present the in-flight system identification and compares the effect of damage on the landing accuracy several adaptive GNC algorithms. Methods using RF beacon technology to overcome GPS blackout are presented in Chapter VII. Conclusions and future work are presented in Chapter VIII.

CHAPTER II

PARAFOIL FLIGHT DYNAMIC MODEL

This section details the simulation model created to develop and validate guidance, navigation, and control algorithms prior to experimental flight testing. Many dynamic models for airdrop systems have been presented in the literature from 3 degree of freedom (DOF) models which decouple lateral and longitudinal motion [59–61] to 9 DOF models which account for different orientations of the payload with respect to the canopy [21, 62, 63]. This work considers a 6 DOF parafoil and payload configuration as it is the simplest model that can accurately model observed behavior of experimental systems. Typically, higher DOF models are used for non-conventional rigging configurations or when the vehicle is tethered to a second system. An comparison and overview of higher DOF models is presented by Gorman and Slegers [64].

The remainder of this section details the derivation of the flight dynamic model using a Newton-Euler approach. Additionally, a sensor and atmospheric model are presented which can be used in conjunction to generate noise and turbulence observed in experiential flight testing. The combination of all three models is commonly referred to as the complete simulation environment.

2.1 Nomenclature

In this work, standard aerospace nomenclature is used including the use of right handed North (I) - East (J) - Down (K) (NED) reference frame orientation. Position vectors from point a to b are denoted by $\bar{r}_{a\rightarrow b}$ and the velocity of point b with respect to frame A is defined as $\bar{V}_{b/A}$. Measure numbers of vectors are enclosed with curly brackets $\{\cdot\}$ while matrix expressions are enclosed with square brackets $\left[\cdot\right]$. Transformation matrices are represented as $\left[T_{\alpha\beta}\right]$ which represents the conversion from the

β to the α frame. Single axis rotational transformations about axis I and angle η, $R\left(\bar{I},\eta\right)$.

The measure number operators $\mathbb{C}_A\left(\cdot\right)$ extracts measure numbers from a vector and the skew symmetric cross product operator $\mathbb{S}_A\big[\,\cdot\,\big]$ is used to represent the cross product of two vectors through matrix multiplication. Given $\bar{a}$ and $\bar{b}$ expressed in frame A, the cross product can be defined as:

$$\begin{aligned}
\bar{a} &= a_x\bar{I}_A + a_y\bar{J}_A + a_z\bar{K}_A \\
\bar{b} &= b_x\bar{I}_A + b_y\bar{J}_A + b_z\bar{K}_A \\
\mathbb{C}_A\left(\bar{a}\times\bar{b}\right) &= \mathbb{S}_A\big[\bar{a}\big]\bar{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix}
\end{aligned} \tag{2.1}$$

Last, several shorthand notations are used. CG defines the vehicle center of gravity and CP defines the aerodynamic center of pressure. Trigonometric functions are also abbreviated as follows.

$$\begin{aligned}
s_\alpha &\triangleq \sin(\alpha) \\
c_\alpha &\triangleq \cos(\alpha) \\
t_\alpha &\triangleq \tan(\alpha)
\end{aligned} \tag{2.2}$$

2.2 Dynamic Model of a Parafoil and Payload System

This work considers a 6 DOF model prescribing inertial translation of the CG (x, y, z) and body orientation described by Euler angles (ϕ, θ, ψ). A cross section representation of a rigid body parafoil and payload system is presented in Figure 2.1. The body reference frame (B) is attached to the vehicle based on its orientation with respect to the North-East-Down (NED) Inertial reference frame. The canopy reference frame (C) is rotated from frame B through angle Γ, $[T_{CB}] = R\left(\bar{J}_B,\Gamma\right)$. This 6 DOF model assumes that once the canopy is inflated it behaves like a rigid body implying no relative motion between the payload and canopy. Deflection of the

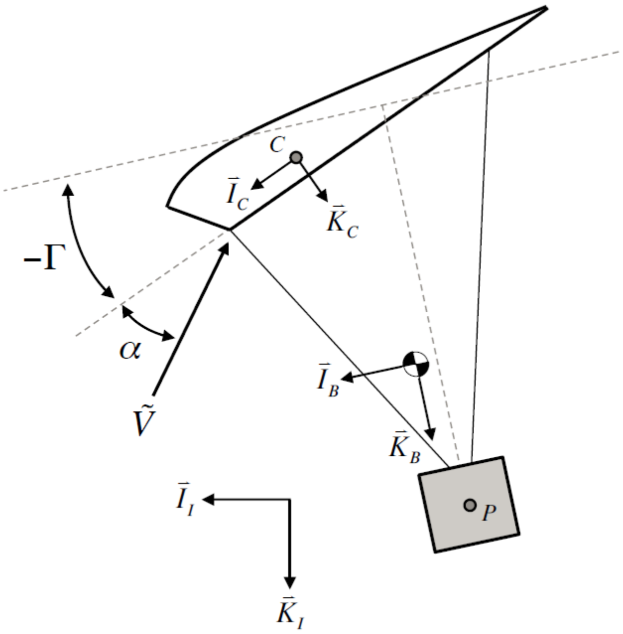


Figure 2.1: 2D representation of a parafoil and payload system

control surfaces, termed trailing edge brakes, are considered as a superposition onto the rigid canopy model.

2.2.1 Kinematics

Kinematics prescribe the relationship between the derivatives of the inertial position and orientation with respect to the body velocity and angular rate. For translation, the velocity of the center of mass, $\bar{V}_{cg/I}$ is expressed in both the inertial and body frame.

$$\bar{V}_{cg/I} = \dot{x}\bar{I}_I + \dot{y}\bar{J}_I + \dot{z}\bar{K}_I = u\bar{I}_B + v\bar{J}_B + w\bar{K}_B \tag{2.3}$$

The transformation from inertial to body frame is conducted through a series of body fixed 3-2-1 Euler angle rotations depicted graphically in Figure 2.2. The vehicle carry (V) and no-roll (N) frames are found through single axis rotations . Based on the Euler rotations, all terms of Equation (2.3) can be converted into the inertial frame to identify the total transformation matrix $[T_{IB}]$.

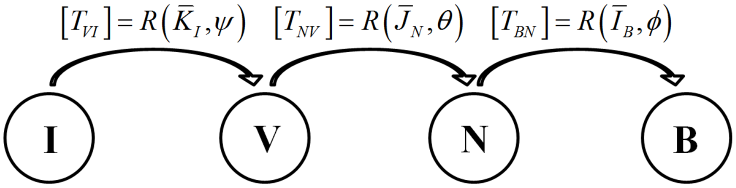

$[T_{VI}] = R(\bar{K}_I, \psi)$
$[T_{NV}] = R(\bar{J}_N, \theta)$
$[T_{BN}] = R(\bar{I}_B, \phi)$
I
V
N
B

Figure 2.2: Graphic representation of the 3-2-1 Euler transformation from Inertial to Body reference frames.

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \left[T_{IB}\right] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{2.4}$$

An identical approach is taken for the Euler angle derivatives. Here the Euler derivatives are expressed in a mixed frame format based on the graphic in Figure 2.2.

$$\bar{\omega}_{B/I} = \dot{\phi}\bar{I}_B + \dot{\theta}\bar{J}_N + \dot{\psi}\bar{K}_I = p\bar{I}_B + q\bar{J}_B + r\bar{K}_B \tag{2.5}$$

Using frame relations to convert all reference frames in Equation (2.5) to the body frame, we get the following relationship.

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\theta \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{2.6}$$

Note that as a result of having a mixed frame definition of the derivative of Euler angles, the transformation matrix in Equation (2.6) is not orthonormal, aka $T^{-1} \neq T^T$.

2.2.2 Dynamics

The impact of aerodynamic loads and gravity on the flight characteristics of the parafoil and payload system are considered through the application of the Newton-Euler method. Inertial derivatives of $\bar{V}_{cg/I}$ and $\bar{\omega}_{B/I}$ expressed in body reference frame are computed using the vector derivative transport theorem to yield the sum of the forces and moments equations below.

$$m\bar{a}_{cg/I} = \Sigma\bar{F} \rightarrow \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} + \mathbb{S}_B\left[\bar{\omega}_{B/I}\right] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \frac{1}{m} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \tag{2.7}$$

$$\frac{d\bar{H}_{B/I}}{dt} = \Sigma\bar{M} \rightarrow \left[I_B\right] \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} + \mathbb{S}_B\left[\bar{\omega}_{B/I}\right]\left[I_B\right] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} \tag{2.8}$$

In the above equations, m is the total vehicle mass and $\left[I_B\right]$ is the inertia matrix about the center of mass. $\{X, Y, Z\}^T$ and $\{L, M, N\}^T$ represent the sum of the forces along and moments about the $\{\bar{I}_B, \bar{J}_B, \bar{K}_B\}^T$ axes. The forces and moments considered in this work consist of gravity (W), payload aerodynamics (PA), canopy aerodynamics (CA), and apparent mass (AM). The last term is an aerodynamic phenomenon that adds additional forces and moments onto a body due to the acceleration of the fluid (air) in which a body traverses [26, 65, 66]. For systems such as parafoils that exhibit low mass and inertia compared to their volume, these effects can significantly impact the flight and turn rate dynamics.

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} X_W \\ Y_W \\ Z_W \end{Bmatrix} + \begin{Bmatrix} X_{PA} \\ Y_{PA} \\ Z_{PA} \end{Bmatrix} + \begin{Bmatrix} X_{CA} \\ Y_{CA} \\ Z_{CA} \end{Bmatrix} + \begin{Bmatrix} X_{AM} \\ Y_{AM} \\ Z_{AM} \end{Bmatrix} \tag{2.9}$$

$$\begin{aligned} \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = \mathbb{S}_B\left[\bar{r}_{cg\rightarrow p}\right] \begin{Bmatrix} X_{PA} \\ Y_{PA} \\ Z_{PA} \end{Bmatrix} + \begin{Bmatrix} L_{CA} \\ M_{CA} \\ N_{CA} \end{Bmatrix} + \mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right] \begin{Bmatrix} X_{CA} \\ Y_{CA} \\ Z_{CA} \end{Bmatrix} \\ + \begin{Bmatrix} L_{AM} \\ M_{AM} \\ N_{AM} \end{Bmatrix} + \mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right] \begin{Bmatrix} X_{AM} \\ Y_{AM} \\ Z_{AM} \end{Bmatrix} \end{aligned} \tag{2.10}$$

The remainder of this section details how these individual forces are computed and finally, how all elements of Equations (2.7) and (2.8) can be combined to solve for the derivatives of body velocity and angular rate.

Gravitational forces act upon the CG of the combined parafoil and payload system, given by Equation (2.11).

$$\begin{Bmatrix} X_W \\ Y_W \\ Z_W \end{Bmatrix} = mg \begin{Bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{Bmatrix} \tag{2.11}$$

For the payload, only aerodynamic drag is considered. This calculation requires the velocity of point P to be given with respect to the atmosphere, $\bar{V}_{P/A}$. The aerodynamic velocity of point P is found using the velocity of the CG and angular velocity of the body from which the elements of the wind are subtracted. As discussed in Section 2.4 below, the atmospheric wind is comprised of both slowly varying translational flows and stochastic disturbances due to high frequency turbulence. $\{V_{Wx}, V_{Wy}, V_{Wz}\}^T$ denotes the total atmospheric wind velocity along the inertial axes.

$$\bar{V}_{P/A} = \tilde{u}_P \bar{I}_B + \tilde{v}_P \bar{J}_B + \tilde{w}_P \bar{K}_B \tag{2.12}$$

$$\mathbb{C}_B\left(\bar{V}_{P/A}\right) = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} - \mathbb{S}_B\left[\bar{r}_{cg\to P}\right] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} - \left[T_{BI}\right] \begin{Bmatrix} V_{Wx} \\ V_{Wy} \\ V_{Wz} \end{Bmatrix} \tag{2.13}$$

Given the aerodynamic velocity, the payload drag is calculated based on Equation (2.14) where ρ is the atmospheric density, S_p is the surface area, and $C_{D,p}$ is the drag coefficient.

$$\begin{Bmatrix} X_{PA} \\ Y_{PA} \\ Z_{PA} \end{Bmatrix} = -\frac{1}{2}\rho S_p C_{D,p} \|\bar{V}_{P/A}\| \begin{Bmatrix} \tilde{u}_P \\ \tilde{v}_P \\ \tilde{w}_P \end{Bmatrix} \tag{2.14}$$

Aerodynamic lift and drag on the canopy is computed at point C (see Figure

2.1) which represents the mean center of pressure for the canopy. Similar to payload aerodynamics, the aerodynamic velocity of point C is required and transformed into the canopy frame (aligned with the airfoil) in order to calculate the angle of attack and side slip angle.

$$\bar{V}_{C/A} = \tilde{u}_C \bar{I}_C + \tilde{v}_C \bar{J}_C + \tilde{w}_C \bar{K}_C \tag{2.15}$$

$$\mathbb{C}_C\left(\bar{V}_{C/A}\right) = \left[T_{CB}\right]\left(\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} - \mathbb{S}_B\left[\bar{r}_{cg\to C}\right]\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} - \left[T_{BI}\right]\begin{Bmatrix} V_{Wx} \\ V_{Wy} \\ V_{Wz} \end{Bmatrix}\right) \tag{2.16}$$

Based on these equations velocity, the canopy speed $\tilde{V}$, angle of attack α, and side slip β can be computed.

$$\tilde{V} = \|\bar{V}_{C/A}\| \tag{2.17}$$

$$\alpha = \arctan(\tilde{w}_C, \tilde{u}_C) \tag{2.18}$$

$$\beta = \arcsin(\tilde{v}_C/\tilde{V}) \tag{2.19}$$

Provided canopy angle of attack in addition to left (δl) and right (δr) trailing edge brake deflections, the aerodynamic lift and drag coefficients are defined based on Equations (2.20) - (2.21). The individual left and right brake deflections are mapped to asymmetric (δa) and symmetric (δb) brake metrics to decouple the effect of the control surfaces into the lateral and longitudinal channels, respectively. Definition of these mappings are provided in Equations (2.22) - (2.23).

$$C_L = C_{L,0} + C_{L,\alpha}\alpha + C_{L,\alpha 3}\alpha^3 + C_{L,\delta b}\delta b + C_{L,\alpha\delta b}\alpha\delta b \tag{2.20}$$

$$C_D = C_{D,0} + C_{D,\alpha 2}\alpha^2 + C_{D,\delta b}\delta b + C_{D,\alpha 2\delta b}\alpha^2\delta b \tag{2.21}$$

$$\delta a = \delta r - \delta l \tag{2.22}$$

$$\delta b = 0.5(\delta r + \delta l) \tag{2.23}$$

Additionally, canopy incidence angle Γ is considered as a longitudinal control input as it directly effects the canopy angle of attack. Nominal incidence angle is varied across the total actuator range, $\Delta\Gamma$, by a nondimensional control parameter $\delta\Gamma \in [0, 1]$ as defined below.

$$\Gamma = \Gamma_{nom} + \delta\Gamma\Delta\Gamma \tag{2.24}$$

Aerodynamic forces acting upon the CP are prescribed in Equation 2.25 where lift, drag, and side forces are transformed from the aerodynamic frame (one aligned with the aerodynamic velocity) to the canopy frame through a single axis rotation, $\left[T_{CA}\right] \triangleq R(\bar{J}_C, \alpha)$. Side force is based on the model parameter $C_{Y\beta}$, and the canopy surface area is given by S_c. Additionally, the offset nature of these forces from the CG enduces a moment and is captured in Equation (2.10).

$$\begin{Bmatrix} X_{CA} \\ Y_{CA} \\ Z_{CA} \end{Bmatrix} = \frac{1}{2}\rho\tilde{V}^2 S_c \left[T_{BC}\right]\left[T_{CA}\right] \begin{Bmatrix} -C_D \\ C_{Y\beta}\beta \\ -C_L \end{Bmatrix} \tag{2.25}$$

Since all aerodynamic forces are only considered to act at the CP, the effects of a 3-dimensional, dihedral wing are accounted for through canopy aerodynamic moments. These stability moments mainly provide angular rate damping to match experimental flight testing. These moments are dependent on the angular velocity of the canopy with respect to the atmospheric winds expressed in the canopy frame, given by Equation (2.26).

$$\bar{\omega}_{B/A} = \tilde{p}\bar{I}_C + \tilde{q}\bar{J}_C + \tilde{r}\bar{K}_C \tag{2.26}$$

$$\mathbb{C}_C\left(\bar{\omega}_{B/A}\right) = \left[T_{CB}\right] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{2.27}$$

Aerodynamic roll C_l, pitch C_m, and yaw C_n moment coefficients are defined based on the canopy angular rates and the effect of both symmetric and asymmetric brake. Equations (2.28) - (2.30) define the typical yaw-roll coupling and the impact that asymmetric brake has on the turn rate behavior of the vehicle.

$$C_l = C_{l\beta}\beta + \frac{b}{2\tilde{V}}\left(C_{lp}\tilde{p} + C_{lr}\tilde{r}\right) + C_{l\delta a}\delta a \tag{2.28}$$

$$C_m = \frac{\bar{c}}{2\tilde{V}}C_{mq}\tilde{q} + C_{m\delta b}\delta b \tag{2.29}$$

$$C_n = C_{n\beta}\beta + \frac{b}{2\tilde{V}}\left(C_{np}\tilde{p} + C_{nr}\tilde{r}\right) + C_{n\delta a}\delta a + C_{n\delta a2}\delta a^2 \tag{2.30}$$

The canopy aerodynamic moments can be calculated in Equation (2.31) using the moment coefficients, canopy span b, and nominal chord length $\bar{c}$.

$$\begin{Bmatrix} L_{CA} \\ M_{CA} \\ N_{CA} \end{Bmatrix} = \frac{1}{2}\rho\tilde{V}^2 S_c\left[T_{BC}\right]\begin{Bmatrix} b\,C_l \\ \bar{c}\,C_m \\ b\,C_n \end{Bmatrix} \tag{2.31}$$

Finally, the apparent mass effects of the canopy are addressed. Apparent mass forces stem from the constant acceleration of displaced air in which the vehicle moves. It has been characterized that these forces have non-negligible effects on low weight to volume systems which can significantly complicate the dynamics [26,65,66]. However, these references also show that the displaced fluid can be represented by manipulating the mass matrix of the vehicle. This is defined as the apparent mass $\left[I_{AM}\right] = \text{diag}(A, B, C)$ and apparent inertia $\left[I_{AI}\right] = \text{diag}(I_A, I_B, I_C)$ of the vehicle. In this work, apparent mass is considered to act on the CP of the canopy with principal axes aligned with the canopy reference frame. Apparent mass effects, defined in Equations (2.32) - (2.33), are based on the acceleration of the CP and angular velocity of the canopy with respect to the atmosphere. Note that atmospheric parameters are assumed to vary slowly such that their time derivatives can be neglected and the apparent mass and inertia matrices must be rotated into the body frame based on Equations (2.34) - (2.35).

$$\begin{Bmatrix} X_{AM} \\ Y_{AM} \\ Z_{AM} \end{Bmatrix} = -\left[I_{AM}\right]'\left(\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} - \mathbb{S}_B\left[\bar{r}_{cg\to C}\right]\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} - \left[T_{BI}\right]\begin{Bmatrix} \dot{V}_{Wx} \\ \dot{V}_{Wy} \\ \dot{V}_{Wz} \end{Bmatrix}\right) \tag{2.32}$$

$$\begin{Bmatrix} L_{AM} \\ M_{AM} \\ N_{AM} \end{Bmatrix} = -\left[I_{AI}\right]' \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} \tag{2.33}$$

$$\left[I_{AM}\right]' = \left[T_{BC}\right]\left[I_{AM}\right]\left[T_{BC}\right]^T \tag{2.34}$$

$$\left[I_{AI}\right]' = \left[T_{BC}\right]\left[I_{AI}\right]\left[T_{BC}\right]^T \tag{2.35}$$

By substituting the equations associated with gravity, payload aerodynamics, canopy aerodynamics, and apparent mass into Equations (2.7) - (2.8), the governing equation of motion is given. Note that all apparent mass effects are captured in the modified mass matrix and result in a coupling between the translational and rotational dynamics. Additionally, $\left[I_{3x3}\right]$ is a 3x3 identity matrix and the total force and moments effects are given in Equations (2.37) - (2.38).

$$\begin{bmatrix} m\left[I_{3x3}\right] + \left[I_{AM}\right]' & -\left[I_{AM}\right]'\mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right] \\ \mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right]\left[I_{AM}\right]' & \left[I_B\right] + \left[I_{AI}\right]' - \mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right]\left[I_{AM}\right]'\mathbb{S}_B\left[\bar{r}_{cg\rightarrow C}\right] \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \cdots \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = \begin{Bmatrix} B_1 \\ \cdots \\ B_2 \end{Bmatrix} \tag{2.36}$$

$$B_1 = -m\mathbb{S}_B\left[\bar{\omega}_{B/I}\right] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + \begin{Bmatrix} X_W \\ Y_W \\ Z_W \end{Bmatrix} + \begin{Bmatrix} X_{PA} \\ Y_{PA} \\ Z_{PA} \end{Bmatrix} + \begin{Bmatrix} X_{CA} \\ Y_{CA} \\ Z_{CA} \end{Bmatrix} \tag{2.37}$$

$$B_2 = -\mathbb{S}_B\left[\bar{\omega}_{B/I}\right]\left[I_B\right]\begin{Bmatrix} p \\ q \\ r \end{Bmatrix} + \mathbb{S}_B\left[\bar{r}_{cg\to p}\right]\begin{Bmatrix} X_{PA} \\ Y_{PA} \\ Z_{PA} \end{Bmatrix} + \begin{Bmatrix} L_{CA} \\ M_{CA} \\ N_{CA} \end{Bmatrix} + \mathbb{S}_B\left[\bar{r}_{cg\to c}\right]\begin{Bmatrix} X_{CA} \\ Y_{CA} \\ Z_{CA} \end{Bmatrix} \tag{2.38}$$

2.3 Sensor Model

To make informed control decisions and adapt to changing flight conditions, accurate feedback must be available. Sensors considered in this work are GPS, barometric altimeter, and a canopy mounted MEMS IMU to provide heading rate needed by the adaptive control algorithm. Relevant beacon signals including spherical range, R_B, radial rate, $\dot{R}_B$, and relative heading to beacon, ψ_B, are generated by manipulating aforementioned sensor data to focus on GNC development instead of beacon hardware integration. Realistic measurements are generated in Equation (2.39) by corrupting state parameters, y_k, with exponentially correlated Gaussian noise, n_k.

$$\begin{aligned} v_k &= y_k + n_k \\ n_k &= e^{-\Delta t/\tau} n_{k-1} + \epsilon_k \sqrt{1 - e^{-2\Delta t/\tau}} \\ \epsilon_k &\sim N(0, \sigma_n) \end{aligned} \tag{2.39}$$

Time constant τ and standard deviation σ_n are chosen based on commercial off-the-shelf sensors and listed for each sensor in Table 2.1.

2.4 Atmospheric Wind Model

Atmospheric wind is modeled by superimposing an altitude varying mean wind field with a high frequency turbulence model to generate a realistic wind field. This combined method simply and effectively captures low altitude wind fields identified during experimental flight testing. Low frequency, large scale wind features are generated

Table 2.1: Sensor error parameters

Sensor: Measurement	Standard Deviation	Time Constant
GPS: Horizontal Position	2.0 m	20 s
GPS: Horizontal Velocity	0.2 m/s	20 s
Altimeter: Vertical Position	3.0 m	1 s
Altimeter: Vertical Velocity	0.2 m/s	1 s
Canopy IMU: Heading Rate	5.0 deg/s	0.5 s

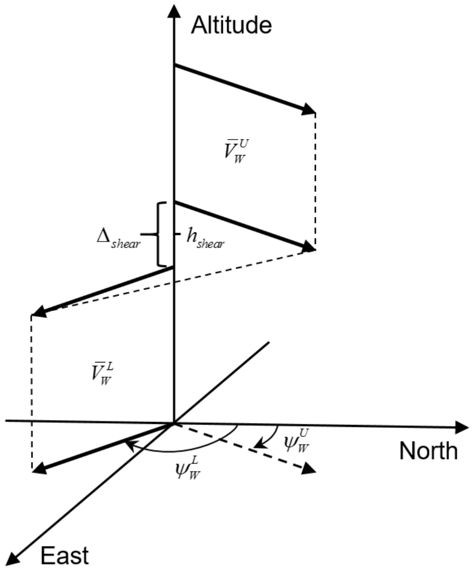
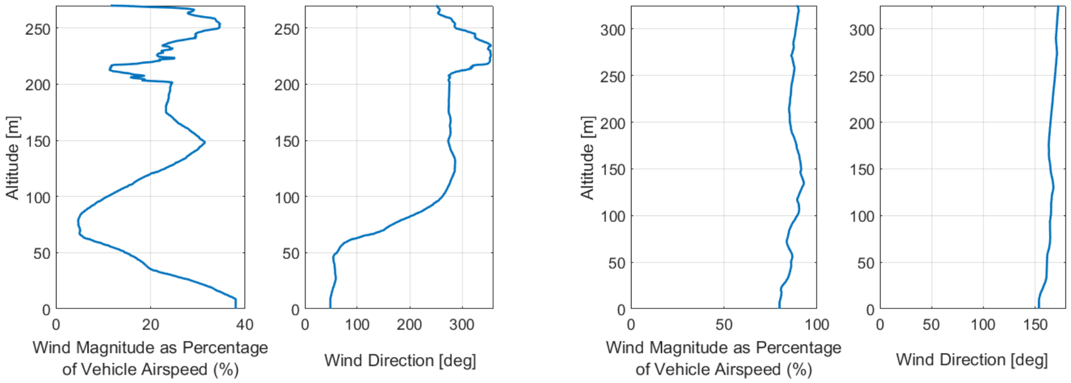


Figure 2.3: Horizontal wind shear profile.

by a horizontal wind shear model. This altitude dependent wind field is presented in Figure 2.3 where parameters are variable between flights and constant during a particular flight. By statistically varying the magnitude and direction of both the upper and lower wind vectors, a rich variety of physical scenarios can be constructed. Note that parameter variation is slightly constrained to match experimental data collection which exhibits a correlation between wind speed magnitude and the direction change of the wind shear. Large wind magnitudes tend to have smaller shear direction changes while low wind magnitudes are prone to large direction changes.



(a)

(b)

Figure 2.4: (a) Low wind magnitude wind field presenting high levels of direction change; (b) High wind magnitude with low wind shear direction change.

This phenomena is presented in Figure 2.4 through two experimentally collected wind fields. Additionally, due to the horizontal spatial variability of thermals and sinks, the vertical winds are not modeled.

To capture high frequency, small scale components of the wind, turbulence is generated according to the Dryden turbulence spectrum given by MIL-STD-1797A [67] (See Table 54). High frequency changes of the wind acting across the control volume of the vehicle induce rotational gusts. Also note that several conventions exist with separate altitude domains [68]. This work focuses on low altitude turbulence modeling (below 1000 ft by definition) which is the primary domain of experimental flight test operations. Gust velocities are computed along the body axes and rotated to align with the inertial NED coordinate system. Turbulence parameters are calculated by driving discrete filters with unit-variance and independent white noise signals, $\eta_i \sim N(0,1)$.

$$u_g(k+1) = \left(1 - \Delta t \frac{V_0}{L_u}\right) u_g(k) + \sqrt{2\Delta t \frac{V_0}{L_u}} \sigma_u \eta_1 \tag{2.40}$$

$$v_g(k+1) = \left(1 - \Delta t \frac{V_0}{L_v}\right) v_g(k) + \sqrt{2\Delta t \frac{V_0}{L_v}} \sigma_v \eta_2 \tag{2.41}$$

$$w_g(k+1) = \left(1 - \Delta t \frac{V_0}{L_w}\right) w_g(k) + \sqrt{2\Delta t \frac{V_0}{L_w}} \sigma_w \eta_3 \tag{2.42}$$

$$p_g(k+1) = \left(1 - \Delta t \frac{2.6}{\sqrt{2L_w b}}\right) p_g(k) + \sqrt{2\Delta t \frac{2.6}{\sqrt{2L_w b}}} \frac{1.9}{\sqrt{2L_w b}} \sigma_w \eta_4 \tag{2.43}$$

$$q_g(k+1) = \left(1 - \Delta t \frac{\pi V_0}{4b}\right) q_g(k) + \frac{\pi}{4b} \left(w_g(k+1) - w_g(k)\right) \tag{2.44}$$

$$r_g(k+1) = \left(1 - \Delta t \frac{\pi V_0}{3b}\right) r_g(k) + \frac{\pi}{3b} \left(v_g(k+1) - v_g(k)\right) \tag{2.45}$$

In Equations (2.40) - (2.45), V_0 is the vehicle airspeed and L_u, L_v, and L_w define the turbulence length scales which are altitude dependent. At higher altitudes, large length scales imply long time constants for the discrete filter resulting in slow variations in the turbulence. Near ground level, the length scales are shortened and time constants are reduced resulting in higher levels of variance. These length scale parameters are defined in MIL-STD-1797A [67] (Figure 266) for altitudes below 1200 ft and given in Equation (2.46). Note that scaling parameters are defined for SI units resulting in SI turbulence velocities and angular rates, though given in the reference material in Imperial units.

$$\begin{aligned} L_u = 2L_v &= \frac{h}{(0.177 + 0.0027h)^{1.2}} \\ L_w &= 0.5h \end{aligned} \tag{2.46}$$

Additionally, the standard deviation of the horizontal components of the Dryden turbulence model vary with respect to altitude while the vertical standard deviation is held constant and left to the user as an input parameter.

$$\sigma_u = \sigma_v = \frac{\sigma_w}{(0.177 + 0.0027h)^{0.4}} \tag{2.47}$$

An example simulated wind field is presented in Figure 2.5 which shows the underlying wind shear model (dashed line) in addition to the total wind field which includes

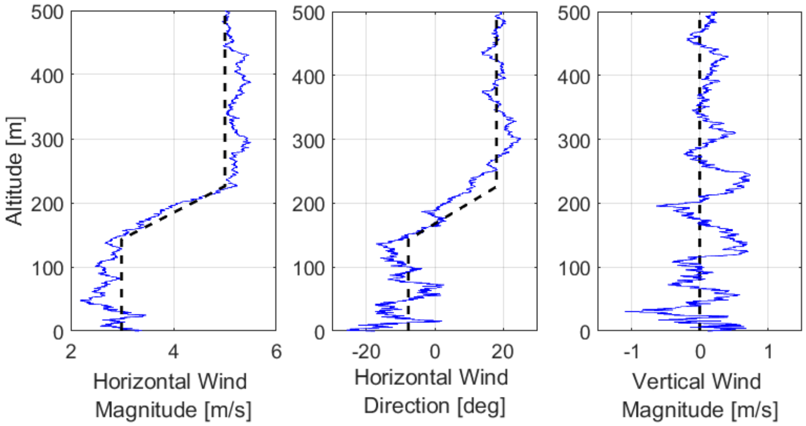


Figure 2.5: Simulated Horizontal wind shear profile.

Dryden turbulence (solid line). It can be observed that while thermals and sinks were not modeled in the underlying model, the turbulence model added segments of flight where there were up and down drafts around 0.5 m/s. Additionally, the impact of altitude dependent length scale can be seen by the increase in stochastic behavior near ground level.

CHAPTER III

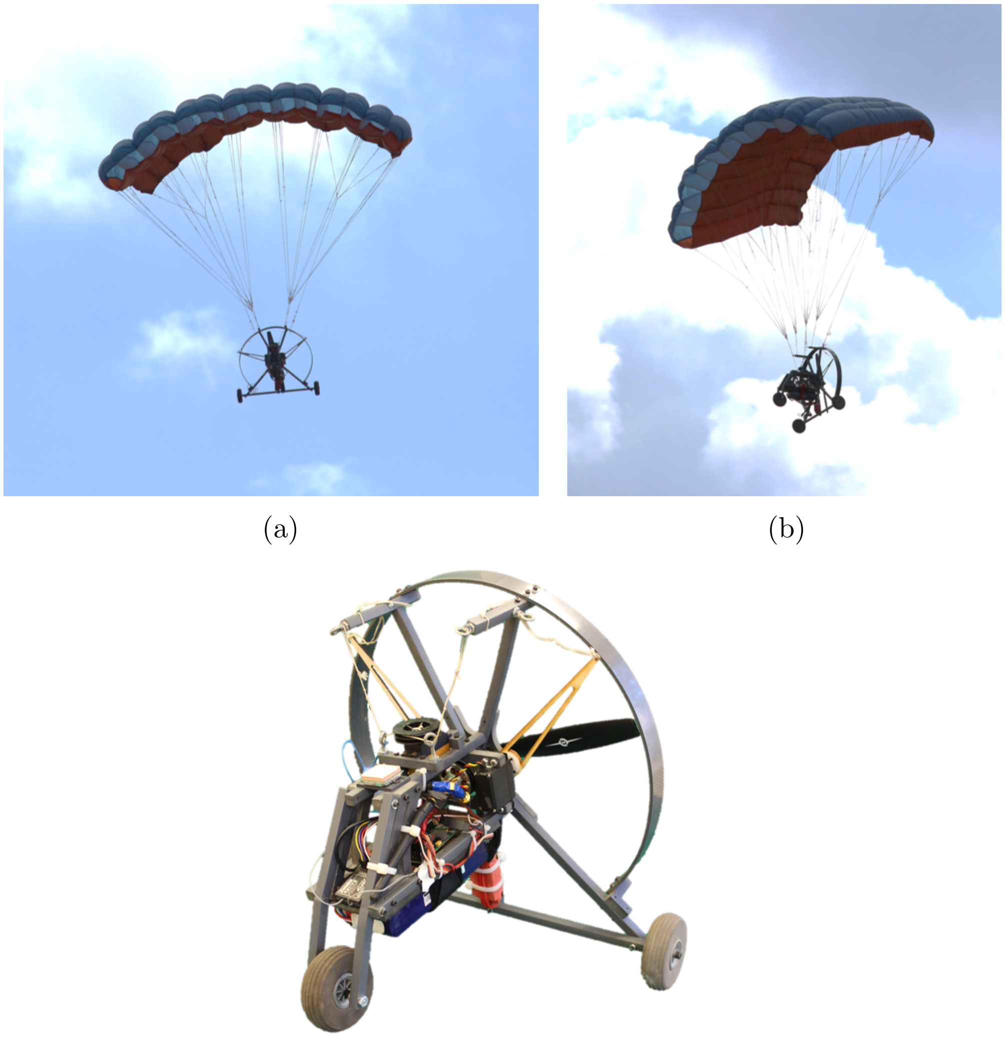
SMALL-SCALE EXPERIMENTAL PLATFORM

3.1 Vehicle Description

Simulated and experimental results are obtained using a small-scale parafoil and payload system shown in Figure 3.1. This small-scale, remote controlled vehicle was designed to provide an elegant alternative to full size airdrop system testing which requires guided payloads to be dropped from manned aircraft. The key benefit of this system is that it can be hand launched from ground level and flown under power until a desired release altitude is reached at which time the motor is turned off and the GNC algorithm is given control of the vehicle. This simulated drop method significantly aids the development, testing, and validation of novel GNC algorithms.

The chassis of the vehicle is equipped with an autopilot, servo motors, electric brushless motor, speed controller, and battery. Additionally, a small coin cell powered wireless MEMS IMU is lcoated in the canopy to provide heading rate feedback. A schematic of the primary computer electronics used for the small-scale flight vehicle is presented in Figure 3.2. The autopilot uses an PIC32MX family microcontroller with clock speed of 80MHz to handle wireless communications, sensor measurements, and GNC calculations at 4Hz. Flight data is stored in EEPROM memory and transmitted to a base station computer in real time via a 2.4 GHz XBee wireless link.

The vehicle is actuated via two high torque servo motors which enable lateral control of the vehicle through trailing edge deflection of the canopy. Power is provided from a 6000 mAh lithium polymer battery, sufficient to power the system over two complete flights. A rear mounted motor and propeller are included to enable ground launches and the simulated drop method. Last, a high torque servo winch is installed



(a)

(b)

(c)

Figure 3.1: Small-scale parafoil and payload system.

to control canopy incidence angle. This not only enables longitudinal control of the aircraft, but is necessary to trim the flight behavior during powered flight.

3.2 System Identification Procedure

This section outlines a procedure used to match the flight dynamic model to the characteristics exhibited by the experimental platform. A graphic overview of the method is presented in Figure 3.3. System identification is based on the extraction of data from steady state maneuvers, notably segments of constant differential brake

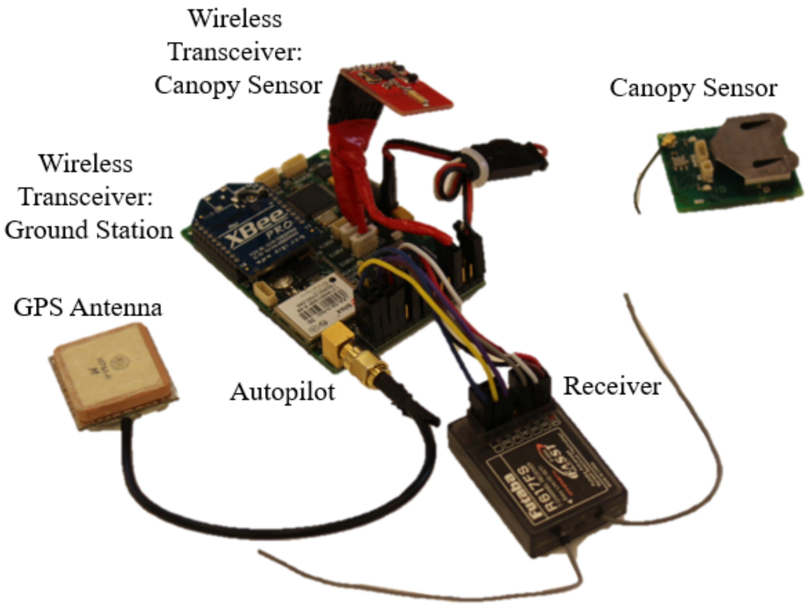


Figure 3.2: Flight electronics used in the small-scale precision airdrop system.

inputs resulting in constant turn rate. These segments enable the atmospheric winds to be characterized and decoupled from the airspeed of the vehicle by solving the underdefined vector diagram illustrated in Figure 3.4 based on measured GPS signals. Measured ground velocity during these segments have a sinusoidal nature as the aerial vehicle velocity vector combines both constructively and destructively with the atmospheric wind velocity depending on the vehicle heading. In each segment, airspeed and atmospheric winds are assumed constant to enable identification. Additionally, the side slip angle β is typically small for parafoil and payload systems which results in the assumption that heading angle is approximately equal to the azimuthal angle ($\psi \approx \chi_0$). The atmospheric wind components, V_{Wx} and V_{Wy}, can be identified by solving for Equation (3.1) based on the work done by Ward et al. [4].

$$\begin{bmatrix} \dot{x}_1^m - \mu_{\dot{x}} & \dot{y}_1^m - \mu_{\dot{y}} \\ \vdots & \vdots \\ \dot{x}_N^m - \mu_{\dot{x}} & \dot{y}_N^m - \mu_{\dot{y}} \end{bmatrix} \begin{Bmatrix} V_{Wx} \\ V_{Wy} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} (V_1^m)^2 - \mu_{V^2} \\ \vdots \\ (V_N^m)^2 - \mu_{V^2} \end{bmatrix} \tag{3.1}$$

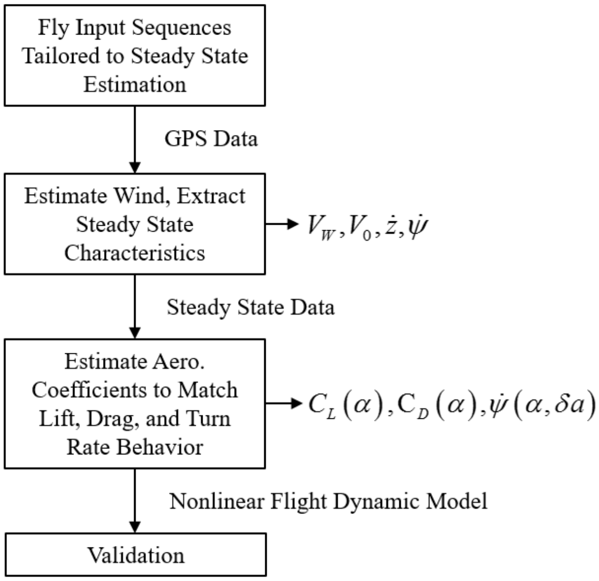

Fly Input Sequences Tailored to Steady State Estimation
GPS Data
Estimate Wind, Extract Steady State Characteristics
$V_W, V_0, \dot{z}, \dot{\psi}$
Steady State Data
Estimate Aero. Coefficients to Match Lift, Drag, and Turn Rate Behavior
$C_L(\alpha), C_D(\alpha), \dot{\psi}(\alpha, \delta a)$
Nonlinear Flight Dynamic Model
Validation

Figure 3.3: Strategy to characterize aerodynamic parameters from steady-state flight data [4].

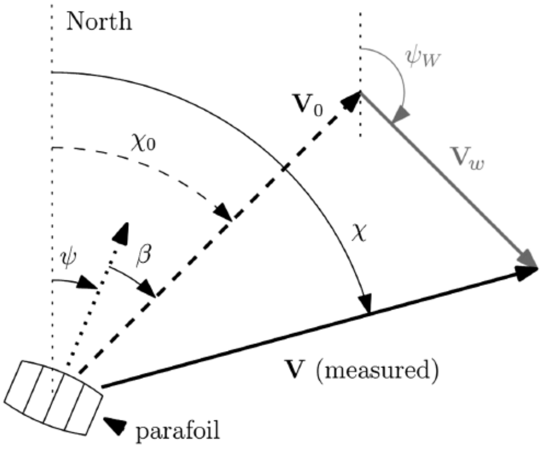


Figure 3.4: Vector diagram illustrating the combined vehicle and atmospheric velocities [4].

Here, $(\dot{x}_i^m, \dot{y}_i^m)$ represent the measured horizontal velocity components, $(V_i^m)^2$ is the squared velocity magnitude, and μ_α is the average value of parameter α in the set $i = 1 \cdots N$. Finally, the airspeed of the parafoil and payload system, V_0, can be estimated using the measured GPS data and horizontal wind estimates.

$$V_{0,i} = \sqrt{(\dot{x}_i^m - V_{Wx})^2 + (\dot{y}_i^m - V_{Wy})^2}, \quad V_0 = \mu_{V_0} \tag{3.2}$$

One major difficulty of the system identification process is the lack of observability of the vertical atmospheric winds from the desecent rate of the vehicle. As a result, descent rate of the vehicle is averaged over an entire flight (for a constant longitudinal) to reject the influences of atmospheric thermals and sinks. This requires the assumption that the vehicles descent rate does not change due to lateral turn rate commands which is reasonable as a turn rate limit of 15 deg/s does not significantly change the roll angle of the vehicle.

By varying both asymmetric brake and canopy incidence angle, the entire flight regime of the aircraft can be characterized to estimate the lift and drag performance in addition to the control sensitivity. Identified steady state data (turn rate, descent rate, and airspeed) is used to balance the lateral and longitudinal force diagrams presented in Figure 3.5. Total vehicle lift and drag, denoted L and D respectively, can be computed using Equations (3.3 - 3.6).

$$\gamma = tan^{-1}\left(\frac{\dot{z}}{V_0}\right) \tag{3.3}$$

$$L' = W cos(\gamma) \tag{3.4}$$

$$L = \sqrt{(L')^2 + (mV_0\dot{\psi})^2} \tag{3.5}$$

$$D = W sin(\gamma) \tag{3.6}$$

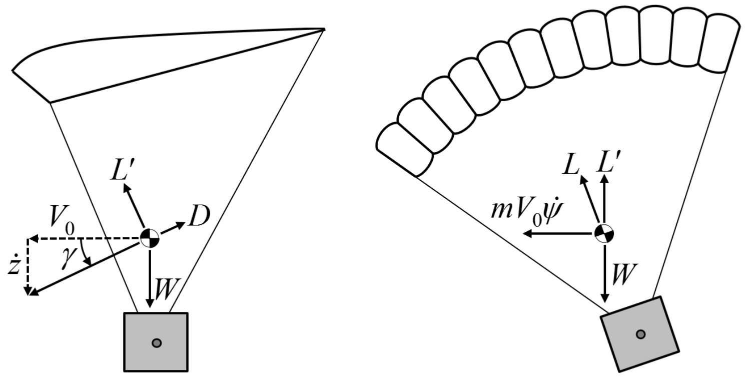


Figure 3.5: Decomposition of lift and drag forces for a parafoil in a steady turn [5].

3.3 Characterized Model Parameters

A series of experimental flights were conducted to characterize the behavior of the small-scale parafoil and payload system. For each flight, the canopy incidence angle was held constant and the various asymmetric brake commands were held to estimate steady state turn rate. Canopy incidence angle was varied from flight to flight to explore the entire operating regime of the vehicle. Flights were conducted in the early morning when atmospheric winds (particularly vertical winds) were mildest. The aforementioned system identification procedure was used to fit experimental data to the simulation model parameters detailed in Chapter II. A list of model parameters is presented in Tables 3.1 - 3.2.

Lateral and longitudinal control sensitivities were also characterized. This is necessary to build the simulation model and to provide the control algorithm with the mappings to effectively track a desired command. The lateral control mapping is shown in Figure 3.6(a) where a left turn bias is exaggerated as the canopy incidence angle rotates to the nose up position (associated with $\delta\Gamma = 1$). This nonlinearity is a result of slight inconsistencies in the rigging geometry of the canopy. The longitudinal response of the vehicle shown in Figure 3.7(a) is assumed to only vary as a result of changing incidence angle. The influence of asymmetric brake commands on airspeed is minimized by enforcing a constant symmetric brake level by splitting a desired δa

Table 3.1: Mass and Geometry parameters of a small scale airdrop system.

Parameter	Value	Units
Total Mass, m	2.7	kg
Canopy Span, b	1.8	m
Canopy Chord, $\bar{c}$	0.75	m
Nominal Incidence Angle, Γ_{nom}	-25	deg
Incidence Angle Range, $\Delta\Gamma$	13	deg
Payload Drag Area, S_p	0.01	m^2
Inertia, I_{xx}	1.93	kg-m^2
Inertia, I_{yy}	0.57	kg-m^2
Inertia, I_{zz}	0.37	kg-m^2
Inertia, I_{xz}	0.104	kg-m^2
Apparent Mass, A	0.05	kg
Apparent Mass, B	0.35	kg
Apparent Mass, C	1.85	kg
Apparent Inertia, P	0.07	kg-m^2
Apparent Inertia, Q	0.06	kg-m^2
Apparent Inertia, R	0.046	kg-m^2

Table 3.2: Aerodynamic parameters of a small scale airdrop system.

Coefficient	Value	Coefficient	Value
$C_{L,0}$	0.0	$C_{l,r}$	0.0
$C_{L,\alpha}$	3.9	$C_{l,\delta a}$	0.0
$C_{L,\alpha 3}$	20.0	$C_{l,p}$	-0.1
$C_{L,\delta b}$	0.0	$C_{m,q}$	-2.5
$C_{L,\alpha\delta b}$	0.39	$C_{m,\delta b}$	0.0
$C_{D,0}$	0.153	$C_{n,\beta}$	0.02
$C_{D,\alpha 2}$	2.4	$C_{n,p}$	0.0
$C_{D,\delta b}$	0.043	$C_{n,r}$	-0.11
$C_{D,\alpha 2\delta b}$	2.062	$C_{n,\delta a}$	0.005
$C_{Y,\beta}$	-1.0	$C_{n,\delta a2}$	0.0
$C_{D,p}$	0.5		

between the left and right actuators. Longitudinal performance indicates that glide slope varies from approximately 2 to 3 across the range of incidence angles.

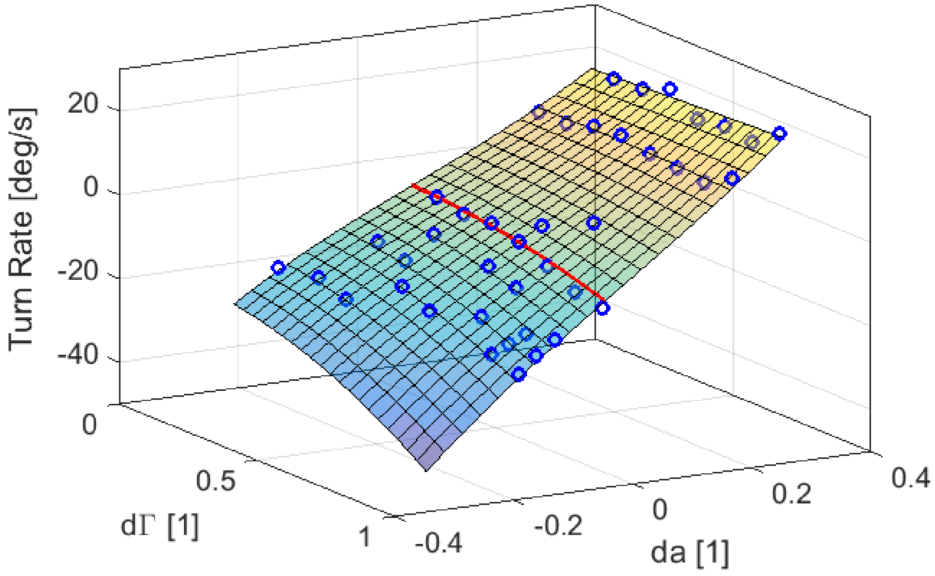


Figure 3.6: Lateral performance of the small-scale airdrop vehicle. Dots represent experimental measurements and the surface is a 2nd order fit of the data.

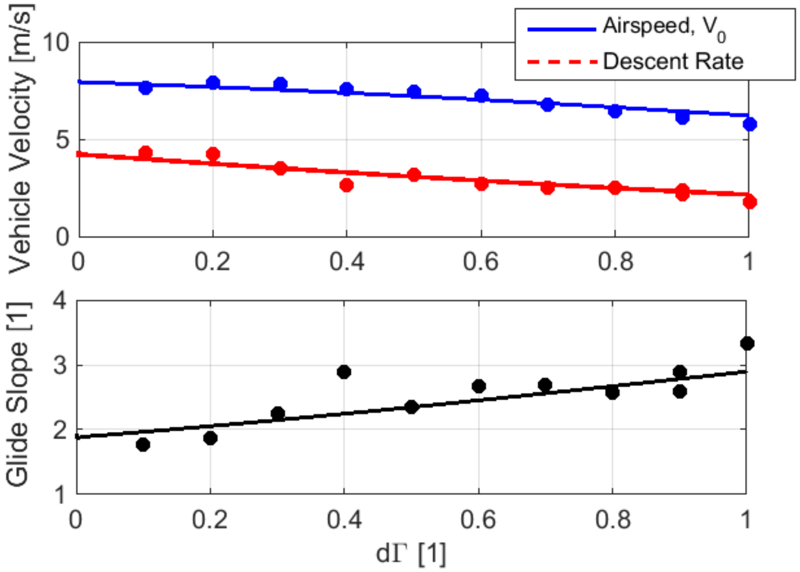


Figure 3.7: Longitudinal performance of a small-scale airdrop system. Dots denote experimental measurements and lines are simulation results.

CHAPTER IV

OVERVIEW OF A CONVENTIONAL GPS-BASED AUTONOMOUS ALGORITHM

This chapter details the fundamental aspects of a guidance, navigation, and control algorithm used for precision airdrop systems when GPS feedback is available. This chapter provides an introduction to guided airdrop strategies and algorithms which enable autonomous control. Many of these ideas are leveraged or expanded upon in the remaining chapters.

At a basic level, the guidance algorithm conducts path planning based on the current state estimates. The navigation algorithm uses GPS measurements to estimate the vehicle position, velocity, heading direction and rate in addition to atmospheric winds. The control algorithm acts upon current state parameters and desired parameters provided by guidance to steer the vehicle along a desired path. More detail on each of these methods are presented in the following sections.

4.1 Guidance

The guidance algorithm uses vehicle state and atmospheric estimates from the navigation algorithm to compute a set of desired paths to accurately reach the IP. To aid path planning, a wind based reference frame (WF) is established which is associated with the drift expected from the influence of the horizontal components of the atmospheric winds. Utilizing the wind based reference frame is advantageous as it decouples the absolute system movement into no wind flight conditions and the drift caused by atmospheric winds. This moving reference frame attempts to capture the impact of the winds from ground level to the vehicle altitude to characterize the

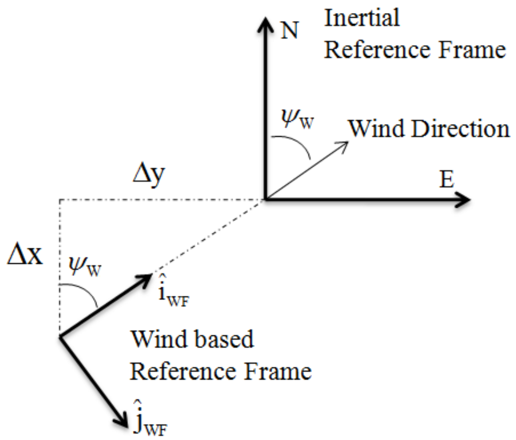


Figure 4.1: Visual representation of the wind based reference frame. It is offset from the inertial origin by an amount $(\Delta x, \Delta y)$ and rotated to align with the wind direction.

zero effort miss of a non-gliding ballistic parachute. With perfect wind knowledge, a ballistic system located at the origin of the wind based reference frame will descend and drift with the atmospheric winds and land precisely at the IP. Additionally, the frame is rotated from the N-E-D inertial frame to align the $\hat{i}_{WF}$ axis with the wind direction expected at ground level. A diagram outlining relation between the inertial and wind based reference frames are presented in Figure 4.1.

This method was initially proposed by Goodrick et al. [10] and more recently by Jann [34] who calculated the expected wind drift, $(\Delta x, \Delta y)$, based upon the integral of the wind profile and descent rate from ground level to the payload altitude, h.

$$\begin{aligned} \Delta x &= \int_0^h \frac{V_{Wx}(z)}{\dot{z}(z)} dz \\ \Delta y &= \int_0^h \frac{V_{Wy}(z)}{\dot{z}(z)} dz \end{aligned} \tag{4.1}$$

Here, $V_{Wx}(z), V_{Wy}(z)$, and $\dot{z}(z)$ are the horizontal wind velocity components in the inertial frame and the vehicle descent rate, respectively. The formula in Equation (4.1) can be simplified by assuming constant descent rate for the system which is normally an excellent assumption for airdrop systems [69, 70]. Using an estimate of the wind profile, a set of altitude - averaged wind components $(\bar{V}_{Wx}, \bar{V}_{Wy})$ are defined which represent the bulk average of the wind profile from the current altitude to the ground.

$$\begin{aligned}\bar{V}_{Wx} &= \frac{1}{h}\int_0^h V_{Wx}(z)dz \\ \bar{V}_{Wy} &= \frac{1}{h}\int_0^h V_{Wy}(z)dz\end{aligned} \tag{4.2}$$

The position of the parafoil and payload system in the wind based reference frame is calculated based on the inertial position, expected wind drift, and remaining time aloft, T_{rem}.

$$T_{rem} = \frac{h}{\dot{z}} \tag{4.3}$$

$$\begin{Bmatrix} x_{WF} \\ y_{WF} \end{Bmatrix} = \begin{bmatrix} cos(\psi_W) & sin(\psi_W) \\ -sin(\psi_W) & cos(\psi_W) \end{bmatrix} \begin{Bmatrix} x_I + T_{rem}\bar{V}_{Wx} \\ y_I + T_{rem}\bar{V}_{Wy} \end{Bmatrix} \tag{4.4}$$

It is important to note that vertical atmospheric winds are not used to shift the wind based reference frame as thermals and local sinks have a high amount of spatial variability and often do not persist throughout the entire flight. The orientation of the wind based reference frame is rotated using the four-quadrant arctangent of the current estimated horizontal atmospheric wind components.

$$\psi_W = atan2(V_{Wy}, V_{Wx}) \tag{4.5}$$

Path planning conducted by the guidance algorithm occurs within the wind based reference frame. The following sections detail the logic used which is broken into four phases: initialization, loiter, approach, and flare.

4.1.1 Initialization

During this phase, initial estimates of the horizontal wind components and vehicle airspeed are computed. This provides an initial condition for the navigation algorithm which will continue to estimate state quantities after the initialization period. This method is an online implementation of the system identification procedure presented in Section 3.2. Equation (3.1) is solved in real time using a standard Kalman filter.

The turn length is chosen to be approximately equal to a full turn of 2π such that the estimation error is bounded by the GPS velocity measurement error [4]. At the end of this open loop flight segment, the vehicle enters the loiter phase.

4.1.2 Loiter

After the initialization phase, the system can define the wind based reference frame which is constantly updated as the wind changes as a function of space and time. During this phase, the parafoil and payload system flies through a set of figure eight turns in order to maintain position relative to the target. This is accomplished by defining waypoints that are at the tips of 'T' shaped reference line that extends downwind of the IP in the wind based reference frame. The T-approach was proposed by Jann [34] and is shown in Figure 4.2. The vehicle uses Dubins path planning to reach the desired waypoints from its current location and heading. Dubins paths minimize the flight distance with three maneuver elements: initially turning at a constant rate in the direction of the next waypoint, flying straight to approach the waypoint, and when near the waypoint, turning to match desired heading defined by the waypoint [34, 69, 71]. When commanding figure eight turns, the system always turns upwind when reaching waypoints, preventing large drifts downwind.

Figure 4.2 illustrates the path planning during transition from initialization to figure eight turns. The key is to study the geometry of the situation by analyzing Dubins path options with circles of set radius (one at the starting location and one at the end) and the straight line that is tangent to both circles. The algorithm analyzes four paths the system could take, turning left or right from the current location and approaching the final location turning either left or right (Figure 4.2 only shows 2 of these options). The shortest total distance is the path chosen by the guidance algorithm, denoted by the solid line in Figure 4.2, one of the three suboptimal paths is also shown as the dashed line.

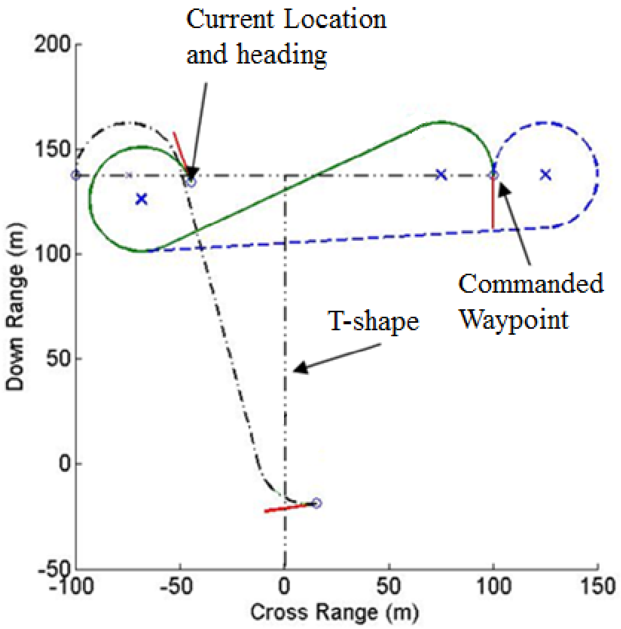


Figure 4.2: Visualization of Dubins path planning transitioning from the left waypoint to the right.

During the entire loitering period, the altitude required to reach the target from the current location is computed. The instantaneous distance to the target is defined by the arc of the circle required to turn from the current heading to point at the target and the straight line between the end of this turning circle and the target.

$$L = |d\psi|R + \|\bar{x}_1 - \bar{x}_T\| \tag{4.6}$$

Here, $d\psi, R, \bar{x}_1$, and $\bar{x}_T$ are the change in heading, turning radius, the coordinates at the end of the turn, and the coordinates of the landing target. Flight distance is converted to the required height based on the estimated descent rate and airspeed using Equation (4.7).

$$h_{REQ} = L\frac{\dot{z}}{V_0} \tag{4.7}$$

When the current altitude equals h_{REQ}, the system switches into the approach phase.

4.1.3 Approach

This section deviates from the figure eight holding pattern and attempts to fly upwind towards the target along the stem of the 'T'. An offset target is introduced that is downwind of the desired impact point which ensures that the end of the trajectory is a straight line segment up the stem of the 'T' with the vehicle pointed into the wind. For a vehicle with only lateral control authority as presented here (longitudinal control is only recently becoming feasible for large scale systems [5]), some adjustments to this trajectory can be made if the system is going to reach the offset target with too much or too little altitude. If the system has too much altitude, the system commands a short doublet maneuver (a left-right turn combination) until the excess altitude is lost and $h_{REQ} = h$. If the system has too little altitude, the offset target is abandoned early, and the system flies directly to the landing point.

4.1.4 Flare

The goal of the landing maneuver is to minimize the kinetic energy of the system just before impact. Below a preset altitude threshold the vehicle is commanded to fly upwind, even if it means turning away from the target. This stresses the importance of accurately lining up downwind of the target during the approach phase. Prior to impact, full symmetric brakes are applied to flare the canopy into pre-stall conditions which decreases the forward airspeed and minimizes the potential for the payload to roll after impact.

4.2 Navigation

For guided parafoil and payload systems, the important states to estimate are the atmospheric winds and vehicle position, velocity, heading angle and rate. After the open loop initialization procedure generates an initial estimate of these parameters, the navigation algorithm continues to propagate them forward in time. We assume that GPS is available to provide 3-dimensional measurements of position and velocity.

The horizontal and vertical positions and velocities are directly measured which allow filtered estimates to be generated through a standard Kalman Filter. For simplicity, presentation of the standard Kalman Filter is presented in terms of the generalized coordinate q which represents the North (x), East (y), and vertical (z) coordinates. Equation (4.8) defines the state propagation to estimate the current *a priori* (-) estimate given the previous *a posteriori* (+) value which is corrupted by the process noise, w.

$$\begin{Bmatrix} q_{k+1}^{-} \\ \dot{q}_{k+1}^{-} \end{Bmatrix} = A \begin{Bmatrix} q_{k}^{+} \\ \dot{q}_{k}^{+} \end{Bmatrix} + \begin{Bmatrix} w_q \\ w_{\dot{q}} \end{Bmatrix}, \qquad A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

The error covariance propagation, Kalman gain, and error covariance update are defined sequentially as:

$$P_{k+1}^{-} = AP_{k}^{+}A^{T} + Q \tag{4.9}$$

$$G_k = P_{k+1}^{-}\left(P_{k+1}^{-} + R\right)^{-1} \tag{4.10}$$

$$P_{k+1}^{+} = \left(I - G_k\right)P_{k+1}^{-} \tag{4.11}$$

where Q and R represent the process and measurement noise covariance, respectively. Since the state matrix A does not change with state or time, Equations (4.9)-(4.11) can be executed offline to identify the stead-state, constant Kalman gain. Q and R are defined by Equation (4.12) with process and measurement noise variances listed in Table 4.1.

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & Q_{\dot{q}} \end{bmatrix}, \qquad R = \begin{bmatrix} \sigma_q^2 & 0 \\ 0 & \sigma_{\dot{q}}^2 \end{bmatrix} \tag{4.12}$$

Using the measured quantities from GPS (denoted by superscript m), updated state estimates are provided using Equation (4.13).

$$\begin{Bmatrix} q_{k+1}^{+} \\ \dot{q}_{k+1}^{+} \end{Bmatrix} = \begin{Bmatrix} q_{k+1}^{-} \\ \dot{q}_{k+1}^{-} \end{Bmatrix} + G \begin{Bmatrix} q_{k+1}^{m} - q_{k+1}^{-} \\ \dot{q}_{k+1}^{m} - \dot{q}_{k+1}^{-} \end{Bmatrix} \tag{4.13}$$

Table 4.1: Process and measurement noise variance values.

	$Q_{\dot{q}}$ (m/s)^2	σ_q (m)	$\sigma_{\dot{q}}$ (m/s)
North and East Pos. and Vel.	2.0	2.0	0.2
Vertical Pos. and Vel.	1.0	3.0	0.5

An extended Kalman Filter is used to estimate the horizontal wind components and vehicle heading angle and heading rate. This filter varies from the stationary Kalman filter as the gain must be calculated at every time step. The state transition matrix is given in Equation (4.14) and assumes slowly varying vehicle heading rate and atmospheric winds. The nonlinear measurement model is presented in Equation (4.16) which relates measurable ground velocity to state parameters.

$$\bar{x}_{k+1}^{-} = A\bar{x}_k^{+} + w_k \tag{4.14}$$

$$\bar{x}_k = \begin{Bmatrix} V_{Wx,k} \\ V_{Wy,k} \\ \psi_k \\ \dot{\psi}_k \end{Bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad w_k = \begin{Bmatrix} w_{V_{Wx}} \\ w_{V_{Wy}} \\ 0 \\ w_{\dot{\psi}} \end{Bmatrix} \tag{4.15}$$

$$\bar{y}_k = g(\bar{x}_k^{-}) + n_k \tag{4.16}$$

$$\bar{y}_k = \begin{Bmatrix} \dot{x}_k \\ \dot{y}_k \end{Bmatrix}, \quad g(\bar{x}_k) = \begin{Bmatrix} V_{Wx,k} + V_0 cos(\psi_k) \\ V_{Wy,k} + V_0 sin(\psi_k) \end{Bmatrix}, \quad n_k = \begin{Bmatrix} n_{\dot{x}} \\ n_{\dot{y}} \end{Bmatrix} \tag{4.17}$$

The extended Kalman filter is a linear estimator and requires linearized models of the observer dynamics. As such, $g(\bar{x}_k)$ is linearized about the current state.

$$C_k = \left.\frac{\partial g(\bar{x}_k)}{\partial \bar{x}_k}\right|_{\bar{x}_k} = \begin{bmatrix} 1 & 0 & -V_0 sin(\psi_k) & 0 \\ 0 & 1 & V_0 cos(\psi_k) & 0 \end{bmatrix} \tag{4.18}$$

The error covariance propagation, Kalman gain, and error covariance update for

Table 4.2: Process and measurement noise variance values.

Parameter	Value	Units
Q_{V_W}	0.25	$(\text{m/s})^2$
$Q_{\dot{\psi}}$	0.03	$(\text{rad/s})^2$
σ_V	0.25	(m/s)

the extended Kalman filter is given below.

$$P_{k+1}^- = AP_k^+A^T + Q \tag{4.19}$$

$$G_k = P_{k+1}^- C_k^T \left(C_k P_{k+1}^- C_k^T + R\right)^{-1} \tag{4.20}$$

$$P_{k+1}^+ = \left(I - G_k C_k\right) P_{k+1}^- \tag{4.21}$$

The updated state estimates can be calculated using Equation (4.22) after computing the $k+1$ Kalman gain.

$$\bar{x}_{k+1}^+ = \bar{x}_{k+1}^- + G_k \left(\bar{y}_k^m - g(\bar{x}_{k+1}^-)\right) \tag{4.22}$$

Finally, the process and measurement noise covariance matrices are presented in Equation (4.23) with parameter values listed in Table 4.2.

$$Q = \begin{bmatrix} Q_{V_W} & 0 & 0 & 0 \\ 0 & Q_{V_W} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{\dot{\psi}} \end{bmatrix}, \qquad R = \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_V^2 \end{bmatrix} \tag{4.23}$$

4.3 Control

Airdrop systems are typically controlled using left and right trailing edge brake deflection. This increases drag on a specific side of the canopy, inducing lateral turning control through a skid turn. The steering controller presented here uses a nonlinear

PI controller to track a commanded heading angle. The proportional component is nonlinear to reduce the control effort when the heading error is small. A commanded turn rate ($\dot{\psi}_c$) is generated based on the heading error between the current desired heading (ψ_{des}) supplied by the guidance algorithm and the estimated heading filtered by the navigation algorithm, ψ_k.

$$\Delta\psi_k = \frac{\psi_{des} - \psi_k}{\Delta\psi_{max}} \tag{4.24}$$

$$\dot{\psi}_c = \begin{cases} \dot{\psi}_{max} & \text{for } \Delta\psi_k \geq 1 \\ -\dot{\psi}_{max} & \text{for } \Delta\psi_k \leq -1 \\ \dot{\psi}_{max}\Delta\psi_k\sqrt{|\Delta\psi_k|} & \text{else} \end{cases} \tag{4.25}$$

From the desired turn rate, asymmetric brake deflection, δa_c, is determined based on an *a priori* control mapping. This expression is supplemented by a bias term identified by the integral controller to allow the vehicle to fly straight under no proportional control.

$$\delta a_c = f(\dot{\psi}_c) + \delta a_{bias} \tag{4.26}$$

$$\delta a_{bias} = K_I \sum_{i=1}^{k} \left(\dot{\psi}_c(i) - \dot{\psi}_k(i)\right) \tag{4.27}$$

It should be noted that δa_c is a parameter that varies in the span $[-1, 1]$ signifying the maximum turning effort in both left and right directions. This asymmetric brake parameter is then converted into the individual brake deflection of the left and right trailing edge, δl_c and δr_c, respectively.

$$\delta l_c = \delta b_c - \frac{1}{2}\delta a_c \tag{4.28}$$

$$\delta r_c = \delta b_c + \frac{1}{2}\delta a_c \tag{4.29}$$

Here, symmetric brake level, δb_c, is included in order to command a flare. With a nominal, non-zero asymmetric brake, each actuator only needs to act over a portion of the needed brake differential, leading to faster actuator response. This method reduces the impact of asymmetric brake on the nominal airspeed of the airdrop system during turning, making path planning computationally more efficient. Only under flare in the final stage of the guidance algorithm, does the symmetric brake level change in order to slow the system to near stall prior to landing. Additionally note that based on these definitions, δl_c and δr_c are mapped from 0 to 1 which is the minimum to maximum range of the brake actuators.

4.4 Example Autonomous Flight

An example simulation based flight trajectory is shown in Figure 4.3 to highlight all 4 stages of the guidance algorithm. A 3 m/s constant wind field blows due North and the vehicle is initialized upwind (South) of the IP which is standard operating procedure. On the right, the vehicle trajectory is plotted in the wind based reference frame where the T-shape loiter pattern and approach is clearly visible. On the left, the trajectory is plotted in the inertial frame to exhibit how the loiter pattern is stretched out to account for the expected wind drift of the vehicle. The vehicle accurately aligns the approach phase into the wind and only slightly overshoots the IP which could be mitigated through the implementation of longitudinal control.

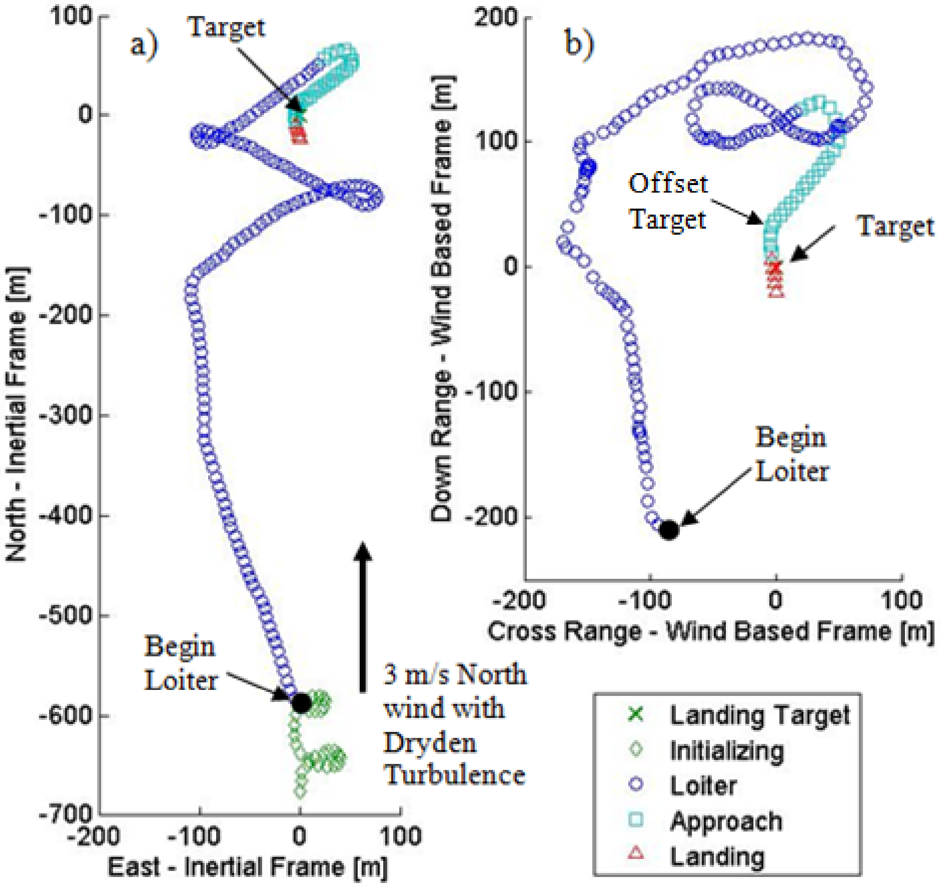


Figure 4.3: Example trajectory of an autonomous airdrop system plotted in the (a) inertial and (b) wind relative reference frames.

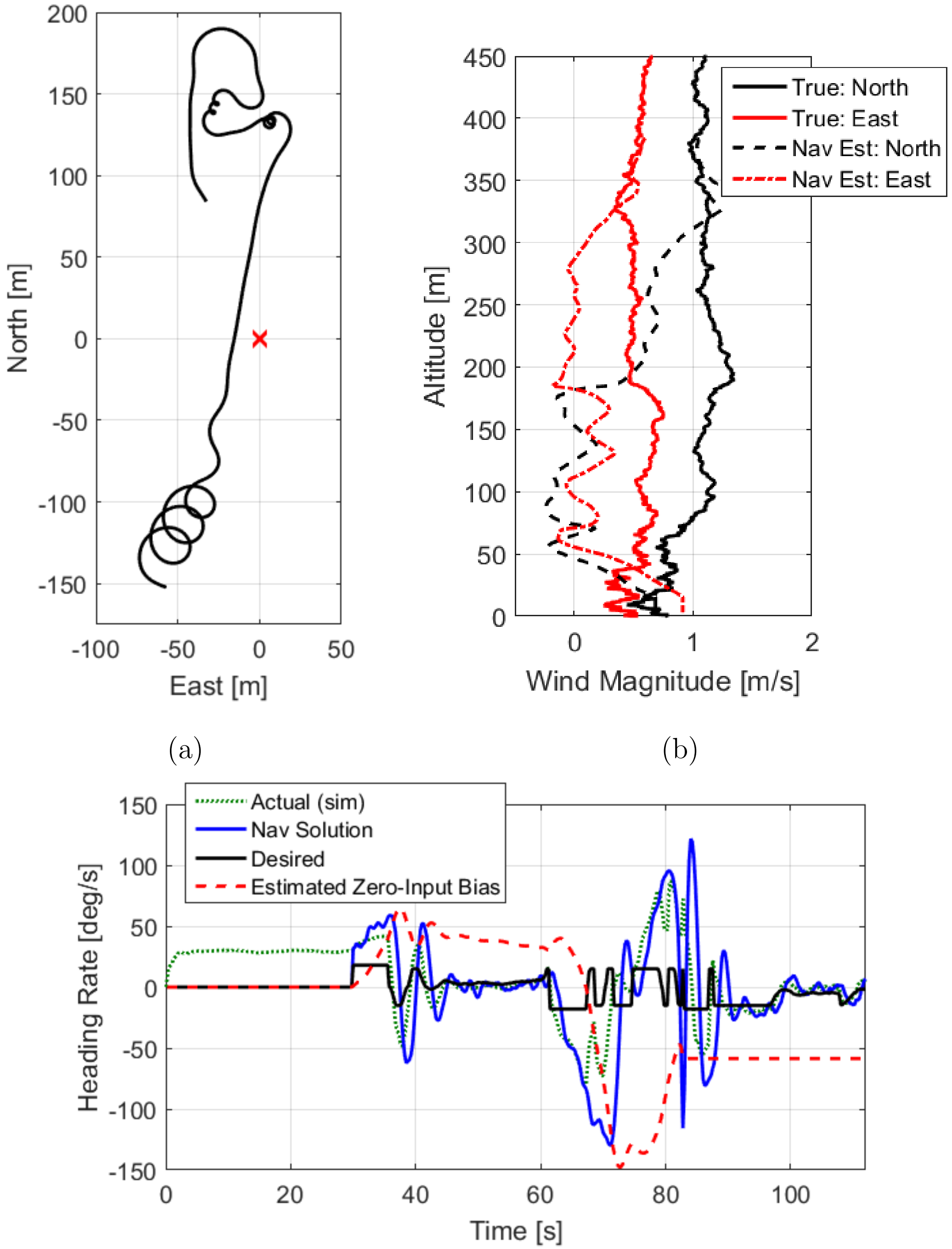
CHAPTER V

A METHOD FOR IN-FLIGHT IDENTIFICATION OF HIGHLY UNCERTAIN SYSTEMS

Most advances in the field of precision guided airdrop systems focus on improved landing accuracy and robustness to atmospheric winds. However, there is a growing need to address the high levels of variability that is observed in the performance of these soft and flexible aerodynamic vehicles. The Conventional GNC algorithm presented in Section 4 has a basic form of adaption as it can successfully characterize deviations in the vehicle airspeed and turn rate bias that occur each flight. Airspeed estimation ensures a convergent navigation solution and the identified turn rate bias improves the heading controller. However, this adaptive behavior is insufficient with high levels of uncertainty in vehicle flight dynamics.

An example flight simulation is presented in Figure 5.1 that highlights the susceptibility of conventional GNCs for precision airdrop systems to variation in the model dynamics. In this flight, the vehicle turn rate sensitivity is doubled causes poor heading and heading rate tracking behavior and lands approximately 90m from the target. The initialization algorithm over estimated airspeed by 10% due to the high turn rate during the initialization period and twice the vehicle entered a steep spiral which cannot always be recovered from in practice. Finally, the integral bias term δa_{bias} in Figure 5.1(c) adapts aggressively but cannot bring the turn rate dynamics under control. Overall, the inability to handle this type of damage or many others stresses the need for more adaptive control.

Precision airdrop systems present a great opportunity for system identification because precise tracking is not required at altitude. The loiter period by definition



(a)

(b)

(c)

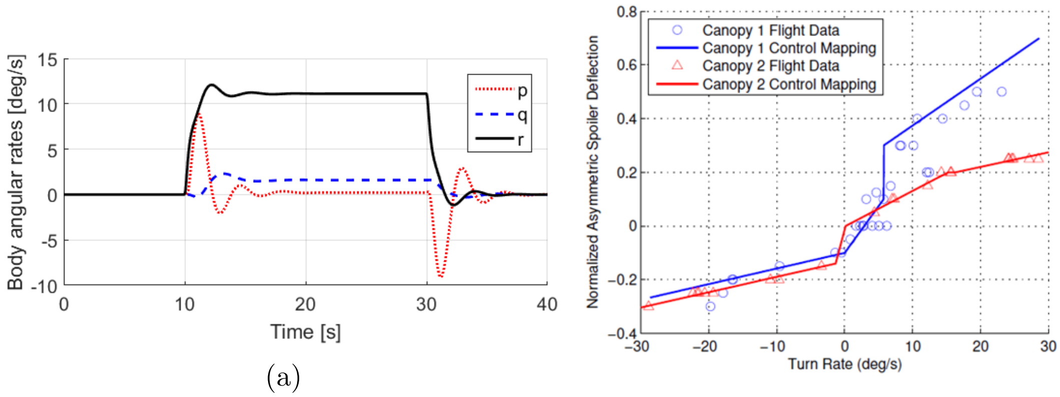
Figure 5.1: Simulated results of the Conventional GNC applied to a vehicle with above nominal turn rate sensitivity.

is a time where they vehicle waits for enough altitude to be lost in order to fly towards and land at the IP. This phase can be re-purposed to explore and characterize the vehicle control sensitivity and dynamic response. When the system transitions to approach, the identified characteristics of the vehicle can be exploited to ensure accurate tracking and precision landing.

Generally, there are two classifications for adaptive systems, direct and indirect. Direct adaptive systems internally adapt parameters to drive the output error to zero through feedback. At the root of these algorithms is a requirement that the sign of the control authority is known for convergent behavior. As this is not guaranteed based on the problem statement, the focus is placed on indirect adaptive systems which take a data driven approach by analyzing input and output data. This data is used to characterize a model which is then used to control the system.

Flight time, and hence identification time, is capped based on the release altitude. To accomplish in-flight system identification, a model is need that has sufficient complexity to capture the dynamic behavior but has a simple enough form to ensure convergence over a short period of flight time. Model selection is aided by knowledge of the typical flight behavior of parafoil systems. The vehicle response to a given input is relatively linear as shown in Figure 5.2a whereas the response level to varying inputs can show significantly nonlinear behavior as in 5.2b.

One model that well represents this is the Hammerstein model. The Hammerstein model is a specific type of nonlinear model which splits the plant into two elements: a linear, time-invariant dynamic block and a static (or memoryless) nonlinear block. Figure 5.3 shows an illustration of the model where inputs $\bar{u}(t)$ can be scaled and/or mixed by the Static Nonlinear Mapping (SNLM) to generate an intermediate, nonlinear control parameter $\bar{v}(t)$. The intermediate inputs excite the Linear, Time-Invariant Dynamics (LTID) to produce the system response $\bar{y}(t)$. This model captures the general characteristics of a precision airdrop system in a simple and concise manner,



(a)

(b)

Figure 5.2: (a) Simulated parafoil dynamic response to a step asymmetric brake input commanded at t = 40 sec; (b) Identified control mappings from flight testing of two 20 ft^2 canopies [5].

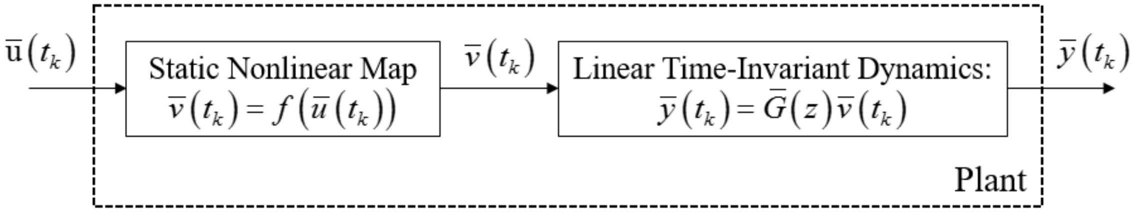

$\bar{u}(t_k)$
Static Nonlinear Map
$\bar{v}(t_k) = f(\bar{u}(t_k))$
$\bar{v}(t_k)$
Linear Time-Invariant Dynamics:
$\bar{y}(t_k) = \bar{G}(z)\bar{v}(t_k)$
$\bar{y}(t_k)$
Plant

Figure 5.3: Graphic representation of a Hammerstein Model.

greatly simplifying the system identification typically need (see Section 3.2). If the model identification time can be reduced sufficiently then the dynamic behavior and control sensitivity of the vehicle can be characterized in-flight for a fully adaptive control algorithm.

Many different representations of the Hammerstein model have been explored in the literature. The LTID block has been represented by autoregressive (ARX & ARMAX), output error (OE), and finite impulse response (FIR) models [72–76]. The order of the dynamic model is often selected *a priori* based on known characteristics of the system to which it represents. Similarly, the form of the SNLM depends heavily on the application and has seen both parametric and nonparametric methods to approximate it. Nonparametric models offer more generality over parametric ones, but can require significantly more computation.

With the parafoil system in mind, this work selects the model dynamics to be represented by an n^{th} order OE model given in Equation (5.1) and the SNLM is represented nonparametrically. This general form of the SNLM ensures robustness to various control mappings and coupling effects that may occur in a degraded condition.

$$G(z) = \frac{\alpha_1 z^{-1} + \ldots + \alpha_n z^{-n}}{1 + \beta_1 z^{-1} + \ldots + \beta_n z^{-n}} \tag{5.1}$$

As a final note, one key benefit of the Hammerstein model is the flexibility of how it can be expanded to higher number of inputs and outputs. In a simple and concise manner, the Hammerstein model accounts for the direct and cross channel correlations between input and output signals. There are also a clear benefits to expand past a lateral only (SISO) controller. Glide slope control has been implemented in several recent works [22, 29–31] and can significantly boost accuracy by reducing under and over shooting the target.

To enable in-flight glide slope estimation and control, accurate measurements of the airspeed and descent rate of the vehicle are required. While this isn't particularly challenging for many fixed wing aircraft, precision airdrop systems are limited in weight allowance and require rugged design to survive potentially hard impacts at landing. As a result, pitot tubes and many other common airspeed measurement devices are ill-suited to this application. Equipped with only GPS, barometric altimeter, and heading rate feedback from a MEMS Inertial Measurement Unit (IMU), the navigation algorithm cannot observe time-varying airspeed, vehicle descent rate, or vertical atmospheric winds. Of particular importance is the descent rate of the vehicle. With only inertial measurements, the effects of flying into and out of thermal columns in the wind can dominate the control authority of the parafoil leading to failed identification. Improvements in sensing can unlock these currently unobservable velocity components of the longitudinal channel.

For the benefit of future expansion, the model identification problem is presented for MIMO systems but focuses the application of the method towards the SISO problem of pure lateral control of a precision airdrop system.

5.1 Hammerstein Model Identification

There are many algorithms for Hammerstein model identification, specifically for unique model representations. The classic methodology by Narendra and Gallman [10] independently optimized the LTID and SNLM elements by iteratively assuming one is constant or known while optimizing the other. The method proved convergence on several trail functions but Stocia [11] further studied the method by noting the conditions when this method provided divergent results. Since the original identification method, several new iterative methods have been proposed [77, 78] which improve the stability of the iterative approach. In addition, researchers have used stochastic methods [79–81], overparameterization methods [75, 82–85], and separable least squares methods [74, 86] to estimate the Hammerstein model. Stochastic methods leverage the properties of white noise to separate the linear and nonlinear elements but suffers if the noise is colored. Overparameterization uses an explicit representation of the SNLM to generate closed form solutions of the input-output relationship in terms of both the LTID and SNLM parameters to allow dual estimation. The separable least squares solution decouples the LTID identification from the SNLM by exciting the system with a quasi-randomly oscillating square wave. This only linearly excites the SNLM and has persistent excitation for a linear system ensuring quality parameter estimation.

While many of these are viable identification procedures, the separable least squares method is ideally suited to the estimation of airdrop systems and will be used here. The LTID can be realized initially through an open loop process similar to the current Initialization procedure (Section 4.1.1). Then, the remainder of the flight is available to improve the SNLM estimate.

5.1.1 Linear Time-Invariant Dynamics

The key to separating the two elements of the Hammerstein model is through the use of a pseudo-random binary sequence (PRBS) or a square wave of changing frequency. A PRBS signal switches between two values $\pm c$ for some $c \neq 0$ in a deterministic, though uncorrelated fashion. As a result of only being excited by two input levels, the SNLM acts as a static gain (two points define a line) and the constant gain can be solved simultaneously with the LTID system. In fact, the value of the static gain is irrelevant as Hammerstein models are known to have non-unique solutions due to the multiplication of the two elements. For a 2x2 MIMO system, this is represented in Equation (5.2) where γ_1 and γ_2 represent non-zero scaling factors to be selected by the control designer.

$$\begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = \begin{bmatrix} \frac{G_{11}(z)}{\gamma_1} & \frac{G_{12}(z)}{\gamma_2} \\ \frac{G_{21}(z)}{\gamma_1} & \frac{G_{22}(z)}{\gamma_2} \end{bmatrix} \begin{Bmatrix} \gamma_1 f_1(u_1, u_2) \\ \gamma_2 f_2(u_1, u_2) \end{Bmatrix} \tag{5.2}$$

To characterize the dynamics, each input channel is individually excited by a PRBS signal. During stage 1 (PRBS on the 1st channel), the linear function $PRBS(c)$ switches between $\pm c$ causing the control input and intermediate nonlinear control to be characterized as follows.

$$u_1(k) = PRBS(c), \quad u_2(k) = 0 \tag{5.3}$$

$$v_1(k) = f_1(PRBS(c), 0) = PRBS(f_1(c, 0)) \tag{5.4}$$

$$v_2(k) = f_2(PRBS(c), 0) = PRBS(f_2(c, 0)) \tag{5.5}$$

As only two values pass through the SNLM, the response can be approximated by linear functions.

$$v_1(k) = b_{11} + \eta_{11} u_1(k) \tag{5.6}$$

$$v_2(k) = b_{21} + \eta_{21} u_1(k) \tag{5.7}$$

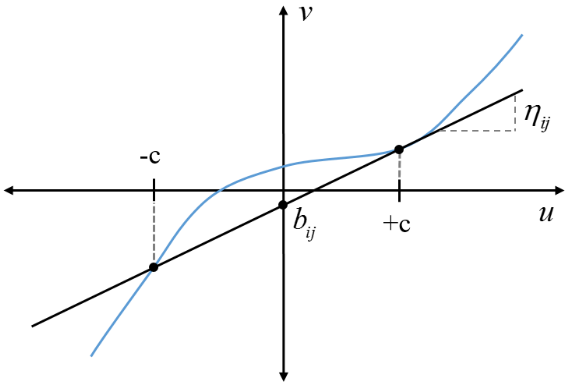


Figure 5.4: Linear approximation of an example SNLM under PRBS input.

Here, b_{ij} and η_{ij} represent the offset and slope of the line that approximates the relationship between the i^{th} output and j^{th} input. A 1-dimensional illustration of this approximation is shown in Figure 5.4 but exists between every input $u(k)$ and intermediate control $v(k)$. Additionally, due to the non-uniqueness discussed in Equation (5.2), the scaling parameters γ have a similar role as the direct channel sensitivity $(\eta_{ij}|i=j)$. As a result, the direct channel sensitivity is assumed equal to 1 and the γ parameter is chosen after realization of $\bar{G}(z)$ and is discussed later in this section.

$$b_{21} = p_1 b_{11} \tag{5.8}$$

$$\eta_{21} = q_1 \eta_{11} \tag{5.9}$$

Here p_i and q_i represent the ratio of cross channel to direct channel steady state bias and control sensitivity of the linear SNLM approximation, respectively. With this relationship, the i^{th} model output can be estimated as:

$$\begin{aligned} y_i(k) &= G_{i1}(z)v_1(k) + G_{i2}(z)v_2(k) && (5.10)\\ &= G_{i1}(z)(b_{11} + u_1(k)) + G_{j2}(z)(p_1 b_{11} + q_1 u_1(k)) && (5.11)\\ &= (G_{i1}(z) + p_1 G_{j2}(z))b_{11} + (G_{i1}(z) + q_1 G_{j2}(z))u_1(k) && (5.12)\\ &= S_{i1}(z)b_{11} + H_{i1}(z)u_1(k) && (5.13) \end{aligned}$$

Using the PRBS input, an input-output relationship is derived which enables a new dynamic model, $H_{i1}(z)$, to be identified. This sub-realization of the LTID model is

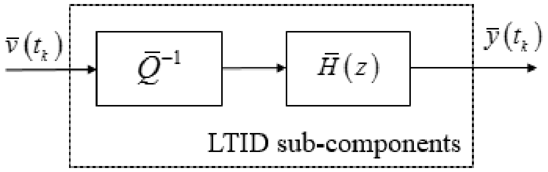


Figure 5.5: Decomposed realization of the LTID block under PRBS input.

represented in Figure 5.5. Note that $H(z)$ takes the same form as $G(z)$ given in Equation (5.1). Considering Equation (5.13) for each input and output, the relationship between $\bar{H}(z)$ and $\bar{G}(z)$ is given as follows.

$$\begin{bmatrix} H_{11}(z) & H_{12}(z) \\ H_{21}(z) & H_{22}(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \begin{bmatrix} 1 & q_2 \\ q_1 & 1 \end{bmatrix} \tag{5.14}$$

$$\bar{H}(z) = \bar{G}(z)\bar{Q}; \quad \bar{G}(z) = \bar{H}(z)\bar{Q}^{-1} \tag{5.15}$$

This method not only allows estimation of the LTID before the SNLM, but splits the problem into a series of SISO identification problems which can be solved in parallel for each output from a given input. By sequentially exciting each input channel with the PRBS sequence, each element of $H(z)$ can be identified.

To solve Equation (5.13), we make use of the fact that both $S(z)$ and $H(z)$ have the same denominator as they are linear combinations of the same elements elements of $\bar{G}(z)$ matrix of transfer functions. Additionally, since b_{11} is a constant the influence of $S(z)$ only adds a static gain. Multiplying Equation (5.13) by the denominator of $H(z)$ generates an equation governing the system output in terms of the unknown model parameters.

$$y_j(k) = -\sum_{s=1}^{n} \beta_s^{i1} y_j(k-s) + \sum_{s=1}^{n} \alpha_s^{i1} u_1(k-s) + b_{11} \sum_{s=1}^{n} \alpha_s^{i1} \tag{5.16}$$

$$y_j(k) = -\sum_{s=1}^{n} \beta_s^{i1} y_j(k-s) + \sum_{s=1}^{n} \alpha_s^{i1} u_1(k-s) + B_{11} \tag{5.17}$$

This can be rewritten in vector form using the data and parameter vectors, $\phi_{i1}(k)$ and θ_{i1}, respectively.

$$y_j(k) = \phi_{i1}^T(k)\theta_{i1} \tag{5.18}$$

$$\phi_{i1}(k) = \{u_1(k), \ldots, u_1(k-n), -y_i(k-1), \ldots, -y_i(k-n), 1\}^T \tag{5.19}$$

$$\theta_{i1} = \{\alpha_1^{i1}, \ldots, \alpha_n^{i1}, \beta_1^{i1}, \ldots, \beta_n^{i1}, B_{i1}\}^T \tag{5.20}$$

The parameter vector is estimated by optimizing the cost of the quadratic error criterion over the span of N measurements.

$$J(\theta) = \sum_{k=1}^{N} [y_i(k) - \hat{y}_i(k)]^2 \tag{5.21}$$

$$= \sum_{k=1}^{N} [y_i(k) - \phi_{i1}^T(k)\hat{\theta}_{i1}]^2 \tag{5.22}$$

The resulting linear least squares estimation problem can be solved recursively using an extended Kalman filter (5.23) - (5.25).

$$G(k) = P(k-1)\phi_{i1}(k)[\phi_{i1}(1)^T P(k-1)\phi_{i1}(k-1) + R(k)]^{-1} \tag{5.23}$$

$$\hat{\theta}_{i1}(k) = \hat{\theta}_{i1}(k-1) + G(k)[y_i(k) - \phi_{i1}^T(k)\theta_{i1}(k-1)] \tag{5.24}$$

$$P(k) = [I - G(k)\phi_{i1}(k)^T]P(k-1) \tag{5.25}$$

Here $G(k)$ is the Kalman gain, $R(k)$ is the process noise, and $P(k)$ is the covariance matrix. Initial conditions for $\hat{\theta}_{i1}(0)$ are based on a nominal system model.

The LTID model identification process can be performed across all outputs simultaneously to calculate each $\hat{\theta}_{i1}$ vector. The PRBS command is then applied to the remaining inputs sequentially, completing the estimation of the sub-dynamic model $\bar{H}(z)$. Based on Equation (5.14), the parameters q_1 and q_2 are selected to reconstruct the transfer function matrix $\bar{G}(z)$, provided $q_1 q_2 \neq 1$. Jeng and Huang [86] proposed two options for the selection of q_i with the goal of improving the controllability of the resulting system. The first method decouples the steady state response of model by selecting q_i such that the gain of the diagonal elements of $\bar{G}(z)$ are zero. The second method drives the system towards diagonal dominance which simplifies the design of a multiloop decentralized controller [87].

This work applies the method of steady-state decoupling in order to drive the majority of the constant, steady-state information into the SNLM instead of embedded in the dynamic equations. This implies that q_i are selected in order to drive $|G_{ji}| = 0$ for $i \neq j$. By defining $|\bar{H}(z)|$ as a matrix of the gain of each individual transfer function (and similarly for $|\bar{G}(z)|$), Equation (5.14) is solved where each matrix is replaced with the magnitude equivalent.

$$|G_{12}(z)| = |H_{12}(z)| + q_1|H_{22}(z)| \tag{5.26}$$

$$|G_{21}(z)| = |H_{21}(z)| + q_2|H_{11}(z)| \tag{5.27}$$

$$q_1 = -\frac{|H_{12}(z)|}{|H_{22}(z)|}; \quad q_2 = -\frac{|H_{21}(z)|}{|H_{11}(z)|} \tag{5.28}$$

This unique solution of the LTID sub-system removes the cross channel gain in the LTID but accounts for their effects in the SNLM based on the definition of q_i as the relationship governing the ratio between cross and direct channel linear control sensitivities.

The initial realization $\bar{G}(z)$ is used to address the non-uniqueness between the dynamic and static nonlinear blocks. The parameters γ_1 and γ_2 given in Equation (5.2) are selected to drive the gain of the diagonal elements of $\bar{G}(z)$ to unity, again to push steady state data to the SNLM.

$$\gamma_1 = |G_{11}(z)|; \quad \gamma_2 = |G_{22}(z)| \tag{5.29}$$

The resulting linear dynamic model captures both the direct and cross channel effects while having a system gain equal to the identity matrix. A linear estimate of the SNLM is given as:

$$\begin{aligned} v_1(k) &= v_{1,0} + \gamma_1 u_1(k) + \gamma_2 q_2 u_2(k) \\ v_2(k) &= v_{2,0} + \gamma_1 q_1 u_1(k) + \gamma_2 u_2(k) \end{aligned} \tag{5.30}$$

where $v_{i,0}$ are the zero input biases calculated from the offset terms, B_{11} and B_{22}, defined in Equations (5.16 - 5.17).

$$\begin{aligned} v_{1,0} &= \frac{\gamma_1 B_{11}}{\sum_{i=0}^{n} \alpha_i^{11}} \\ v_{2,0} &= \frac{\gamma_2 B_{22}}{\sum_{i=0}^{n} \alpha_i^{22}} \end{aligned} \tag{5.31}$$

Here, α_i^{11} represents the numerator coefficients of the LTID sub-system transfer function $H_{11}(z)$.

Finally, note that as a result of $\bar{G}(z)$ having gain equal to the identity matrix, the nonlinear control inputs are equal to the desired stead state output of the system, $\bar{y}_d(k)$. This implies that the SNLM is not just a nonlinear scaling and mixing function, but an explicit representation of the control mapping of the vehicle and Equation (5.30) provides a first order estimate.

5.1.2 Static Nonlinear Map

The goal of this section is to improve upon the initial linear estimate of the static nonlinear map calculated in Equation (5.30) by using real-time input and output data. Since this approach does not assume a structure to the SNLM based on *a priori* knowledge, the estimation method at its core is non-parametric. To solve this, several articles that use the PRBS identification method highlight that once the LTID are known, the intermediate control signal $\bar{v}(k)$ can be estimated by applying the inverse dynamics to the output measurements [74, 76]. Assuming $\bar{G}(z)$ is invertible, the intermediate control is simply $\bar{v}(k) = \bar{G}(z)^{-1}\bar{y}(k)$. This enables a series of points $(u(k), v(k))$ to be directly compared and the form estimated in either a parametric or non-parametric fashion.

There are several problems associated with generating $\bar{v}(k)$ from $\bar{y}(k)$. First, the realization of a non-minimum phase system for $\bar{G}(z)$ significantly complicates the computation. One solution is to convert the model into a minimum phase system using an all-pass filters [88]. This approximates the original transfer function with one where the non-minimum phase zeros are replaced with their conjugate pair inside the

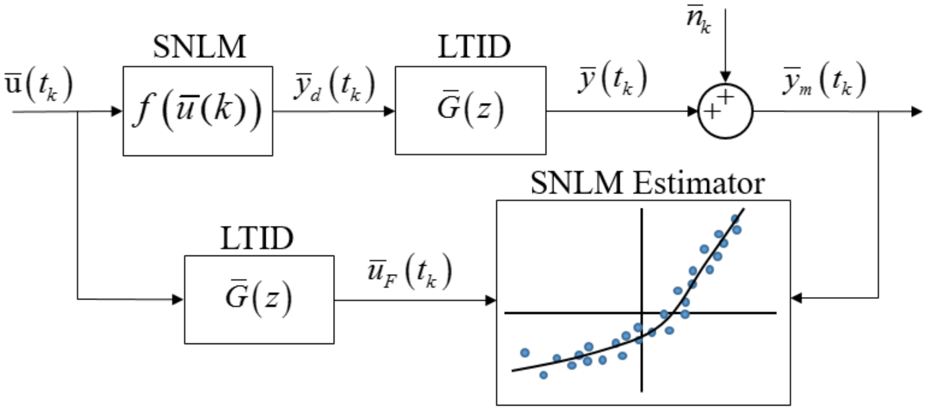


Figure 5.6: Representation of method used to generate a real-time comparison of the input-output relations.

unit circle. The new all-pass system has identical magnitude behavior but introduces a phase difference which can add significant error to a series of input-output data used to realize the mapping. Second, inverted models can have lead-compensator characteristics which amplify high frequency components in the signal, notably measurement noise. While these methods are viable depending on the system being characterized, the need for a fast, in-fight, and robust method limits their application.

A significantly simpler option is available by re-framing the problem. First, note that as a result of the unity gain of the LTID block, any signal passed through the dynamics will only experience a phase delay but not steady state scaling. Furthermore, a constant value passed through $\bar{G}(z)$ is unchanged. As a result, the entire static nonlinear function can be passed through the dynamics without scaling or phase delays. Instead of removing the phase dynamics from the measured output as would be done through dynamic inversion, this method adds phase to the input command to estimate the delayed input command, $\bar{u}_F(k)$. Figure 5.6 depicts how the phase delayed input signal can be compared to the measured system response to improve the estimate of the SNLM.

One of the primary benefits of this method is that it allows for the control sensitivity mapping to be characterized in the parameter domain instead of temporally.

All steady state and dynamic data is passed into the SNLM estimator whereas conventional parameter estimation of parafoil systems is purely based on time-averaged, steady state data. At this point it should be noted that to properly characterize the nonlinear behavior of the SNLM, sufficient parameter exploration must be conducted. Given a stream of phase matched input and output data entering the SNLM estimator, any parametric or non-parametric method can be used to model the control sensitivity.

As far as the general identification problem is concerned, this method for the characterization of the SNLM concludes the model identification process. Model fitting of this data is highly specialized based on the number of inputs/outputs and how much of the parameter space needs to be explored to achieve the goals of the GNC algorithm. Further detail on modeling the lateral only data stream sent to the SNLM estimator used in this work is presented in the following Section.

5.2 Hyperadaptive Guidance, Navigation, and Control

This section details extensive expansions to guided airdrop flight software to embed the Hammerstein model and identification process at the core of the GNC. Expansions include a pre-initialization input check, an expanded initialization for airspeed and LTID estimation, control considerations using a variable control mapping, and improved path planning.

For the lateral-only GNC developed here, the mathematical formulation of the previous section can be simplified to the SISO case. The control input is taken as the asymmetric deflection of the trailing edge brake ($u_1 = \delta a$) and the output is the measured heading rate ($y_1 = \dot{\psi}$). The assumed form of the dynamic transfer function which governs the turn rate response is chosen to be first order. Analysis into higher order models characteristically lead to over fit data or systems dominated by the first order pole. Additionally it was noted that adding a short delay of one time update to the first order model reduced RMS error.

$$G(z) = \frac{\alpha_2 z^{-2}}{1 - \beta_1 z^{-1}} \tag{5.32}$$

The following sections detail the novel hyperadaptive GNC designed to handle a large range of flight behavior. At its core are many of the underlying elements of the Conventional GNC presented in Chapter 4, including an unchanged Navigation algorithm. For brevity, the following section focus on the key updates and relies heavily on the concepts presented previously.

5.2.1 Guidance

Within the guidance algorithm, two major updates were implemented to integrate the hyperadaptive architecture with the strategies employed for precision landing. First, the initialization phase was expanded to include a pre-initialization phase to ensure sufficient turn rate response to conduct the initialization procedure. The initialization phase itself was expanded to simultaneously estimate the constant vehicle airspeed, horizontal components of the atmospheric winds, and the linear turn rate dynamics. Second, path planning was updated to account for the influence of slow turn rate dynamics on the ability to track conventional Dubins paths.

5.2.1.1 Updated Initialization

The pre-initialization phase is essential to find an input command that sufficiently excites the turn rate response for airspeed estimation but does not over actuate the vehicle causing a spiraling descent. The principle of this approach is a sweep of the control input until a viable turn rate is identified. The initial turn rate command is chosen to be conservative in case of highly sensitive turn rate response. During this phase, if the measured turn rate is outside of the desired bounds ($|\dot{\psi}(k)| < \dot{\psi}_L$, $|\dot{\psi}(k)| > \dot{\psi}_U$) then the input command is varied linearly to attempt to find the optimal turn rate, $\dot{\psi}_{max}$ which is the desired constant turn rate used by the path planning algorithm (15 deg/s).

$$da(k) = \gamma \left| \frac{\dot{\psi}_{max,gd}}{\dot{\psi}(k)} \right| da(k-1) \tag{5.33}$$

The value of da is changed monotonically until either the exit condition is reached or the maximum input command is reached. If the latter occurs, the input command is switched to a conservative input of the opposite sign and the process repeats. This relatively blind controller is computed at a low frequency to ensure the measured output has sufficient time to respond to a change in command.

The initialization phase begins when sufficient steady state turn rate is generated. Similar to the conventional GNC, the initialization begins with a constant control input period to estimate the vehicle airspeed and horizontal components of the atmospheric wind. It is important to note that this algorithm only depends on the heading change and not the sign of which is not guaranteed.

One of the primary benefits of the PRBS approach to the identification of the Hammerstein model is that it can be combined with the maneuver required to run the airspeed and wind estimation algorithm. This saves altitude by using the first constant control segment of the PRBS as the command for the airspeed estimator. After the vehicle airspeed and initial estimates of the atmospheric winds are generated, the navigation algorithm presented in Section 4.2 continues to propagate state estimates including the atmospheric winds. Additionally at this point, the guidance algorithm can define the vehicle's heading and position in the wind based reference frame (Equation (4.4)). To ensure the vehicle maintains a tight proximity to the target, the second segment of the PRBS command is used to align the heading towards the target as defined in the wind based reference frame. This only applies in cases where the second control input produces sufficient turn rate, ($|\dot{\psi}(k)| > \dot{\psi}_L$). The combination of these first two commands represents a long doublet input. The remaining inputs of the PRBS have shorter switching times to ensure sufficient exposure to the dynamic response. If full turn control authority is is available, the final input sequence results in a weaving maneuver generally aligned towards the target.

5.2.1.2 Updated Path Planning

After the LTID model has been estimated, the system exits open loop control and enters the loiter phase. This phase and the remaining phases of the guidance strategy have relatively minimal updates except for an improvement to Dubins path planning. As discussed in Section 4.1.2, Dubins paths are characterized by three maneuvers: a turn at constant turn rate toward the desired waypoint, a segment of straight flight and when close to the waypoint, a second turn at constant turn rate to match the desired heading. These ideal paths assume instantaneous changes in turn rate. When applied to vehicles with fast turn rate dynamics, the tracking error is primarily driven by the disturbance of atmospheric winds. However, as the turn rate dynamics slow, the assumption for instantaneous changes in heading rate becomes problematic and introduces error to the landing accuracy.

This work proposes a modification to the standard Dubins algorithm [71] to include the distance traveled while transitioning between straight and circular flight. This transition is governed by the settling time of the turn rate dynamics which were previously identified. Used both for tracking waypoints and estimating the altitude margin, these updates provide pathing solutions that the vehicle is capable of tracking. An illustration of the current flight distance to the target is presented in Figure 5.7. The solid line presents the modified Dubins path which adds segments $\overline{PA}$ and $\overline{EF}$ to transition the vehicle between the circular turn rate and the turn rate constraints. Points A and E each represent the new initial and final location used by the regular Dubins path as now all of the assumptions of the algorithm have been satisfied.

The calculation of the transition paths $\overline{PA}$ and $\overline{EF}$ is performed using the schematic insert in Figure 5.7. The initial offset $(\Delta x, \Delta y)$ is calculated using an approximation that the turn rate acceleration is equal to the ratio of heading rate change to the

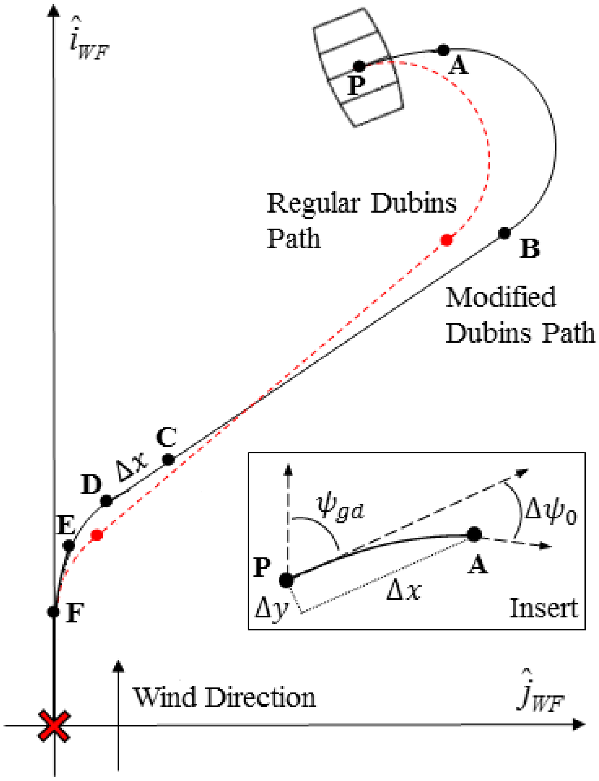


Figure 5.7: Comparison of regular and modified Dubins path planning algorithms.

dynamic settling time, T_s.

$$\ddot{\psi} = \frac{\Delta\dot{\psi}}{T_s} \tag{5.34}$$

Double integration of this function yields the change in heading expected over the maneuver, $\Delta\psi_0$. This is based on the direction of the 1st turn, characterized by s_1, and the current turn rate given by $\dot{\psi}_0$.

$$\Delta\psi_0 = \frac{1}{2}(s_1\dot{\psi}_{max} + \dot{\psi}_0)T_s \tag{5.35}$$

Finally the offset point A is defined using a small angle approximation for $\Delta\psi_0$.

$$\bar{x}_{A,WF} = \bar{x}_{P,WF} + \Delta x\hat{e}_t + \Delta y\hat{e}_n \tag{5.36}$$

$$\Delta x = V_0 T_s \tag{5.37}$$

$$\Delta y = V_0 T_s \Delta\psi \tag{5.38}$$

Here, $\hat{e}_t$ and $\hat{e}_n$ represent the tangent and normal vectors defined by the Frenet - Serret (FS) basis vectors [89]. Transformation from the WF to FS reference frame is accomplished by a single axis rotation about $\hat{z}_{WF}$ by the heading angle in the WF frame ψ_{gd}.

Calculations for the final offset $\overline{EF}$ are based on the final waypoint constraints but follow an identical strategy.

5.2.2 Control

Major updates replace the static turn rate controller which depends on *a priori* control mappings with an algorithm that focuses on the exploration and exploitation of the static nonlinear map (SNLM). Using the data driven approach presented in Section 5.1.2, the control input δa is filtered by the LTID model to align the phases of the input and output signals. The key benefit is that this approach allows the input and output signals to be compared in the parameter space of the SNLM instead of depending on time averaged, steady state data. As a result, the control authority can be estimated entirely in-flight and used to improve the tracking and landing accuracy of the vehicle.

5.2.2.1 Estimation of the Control Sensitivity Mapping

This work uses a dynamic table look up approach to characterize the SNLM, graphically represented by a piecewise linear fit of the data that is passed into the SNLM estimator. In a concise and computationally cheap manner, this method is flexible to adapt to many nonlinearities that may be the resultant of degraded conditions including deadbands and control reversal.

The linear piecewise function is comprised of N continuous line segments in to characterize the $(\delta a, \dot{\psi})$ parameter space using measurements of $(\delta a_F, \dot{\psi}_m)$. Each segment in the set $j = 1...N$ is defined by the two endpoints (a_j, b_j) & (a_{j+1}, b_{j+1}). The values of a_j are assumed to be fixed and it is the task of the estimation algorithm to find the optimal values of b_j. Each line segment is constrained by the continuity constraint in Equation (5.41) to yield the governing linear equation below.

$$\dot{\psi} = f_j(\delta a);\; a_j \leq \delta a < a_{j+1} \tag{5.39}$$

$$= \frac{b_j(a_{j+1} - \delta a) + b_{j+1}(x - a_j)}{a_{j+1} - a_j} \tag{5.40}$$

$$f_j(a_{j+1}) = f_{j+1}(a_{j+1}) \tag{5.41}$$

This series of equations is moved into parametric vector form to enable a recursive solution of b_j.

$$\dot{\psi}(k) = \phi^T(k)\theta \tag{5.42}$$

$$\phi(k) = \left\{0, ..., 0, \left.\frac{\partial f_j(\delta a)}{\partial b_j}\right|_{\delta a = \delta a_F}, \left.\frac{\partial f_j(\delta a)}{\partial b_{j+1}}\right|_{\delta a = \delta a_F}, 0, ..., 0\right\}^T \tag{5.43}$$

$$\theta = \{b_1, ..., b_{N+1}\}^T \tag{5.44}$$

The parameters are updated using an extended Kalman filter which is identical to the estimation problem set up and discussed in Equations (5.22), (5.23) - (5.25).

Selection of the proper values of a_i is very important to the success of this linear fit and the control algorithm. If the $N + 1$ endpoints are stretched across the parameter domain $\delta a \in [-1,\ 1]$, then systems with very high control sensitivity have poor resolution. In addition, segments with relatively few data points within its endpoints tend to have very high variance which can greatly increase the impact of measurement noise on the turn rate mapping. Knowledge of the vehicle and the paths generated by the GNC provide insight into proper selection. Due to Dubins path planning the primary regions of control are around the points $\dot{\psi} = \{0, \pm\dot{\psi}_{max}\}$. Similarly, the region around 75% of $\dot{\psi}_{max}$ typically has significantly fewer data points in comparison to other segments. The optimal region to place endpoints was characterized to be associated with the turn rate values $\dot{\psi}_{SNLM,0} = \{-1, -1/3, ..., 1/3, 1\}\dot{\psi}_{max}$.

When this algorithm is initialized at the start of the loiter phase, the values of b_j are set based on $\dot{\psi}_{SNLM,0}$ and the parameters a_j are estimated using the linear estimate

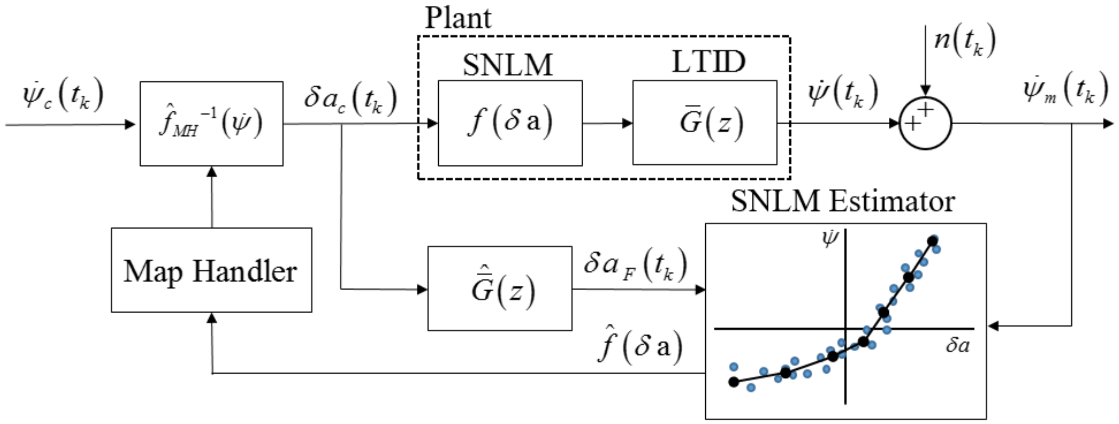


Figure 5.8: Block diagram representation of the control strategy used in the hyper-adaptive GNC.

of the SNLM in Equation (5.30). As the map evolves, it is important to periodically recalculate the a_j terms based on the current mapping such that it continues to represent the domain of $\dot{\psi}_{SNLM,0}$. Updates to a_j occur at a low frequency (at the conclusion of any Dubins constant turn rate segment) and the covariance matrix of the recursive least squares estimator is boosted to account for the shift in parameters.

5.2.2.2 *Control of Hammerstein Systems*

The common control strategy for Hammerstein systems is to cancel out the effects of the SNLM and control the linear dynamic system, $G(z)$ [90–92]. This strategy is presented in Figure 5.8 and implemented here to track the desired heading rate that is commanded from guidance algorithm. However, it is important to note that the SNLM estimate cannot blindly be inverted to generate a valid control mapping. It is assumed that $\dot{\psi} = f(\delta a)$ is a function which enabled parameterization of the mapping, but the inverse is not guaranteed due to potential nonlinear behavior.

Issues with invertability of the SNLM are resolved by using a map handler which approximates the SNLM with an invertible function. Any non-monotonic behavior is removed from the mapping by placing a deadband across the region. This prevents the control algorithm from commanding a δa value within this region and effectively

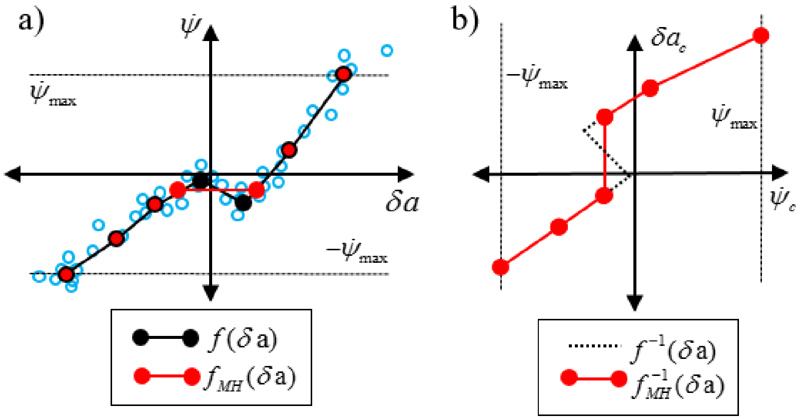
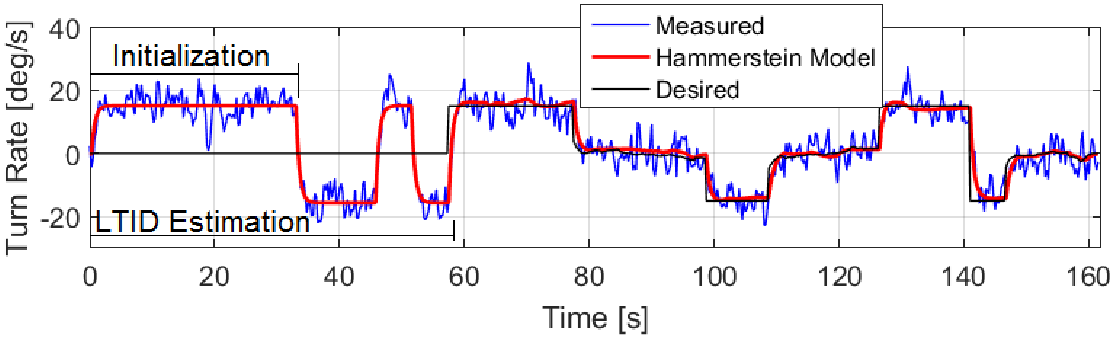


Figure 5.9: a) Graphical representation of the SNLM estimator; b) Control mapping characterized by the Map Handler.

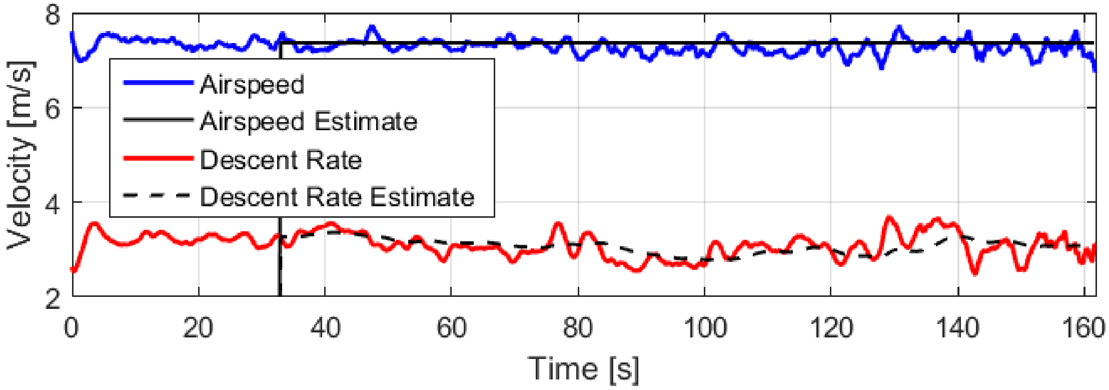
excludes the complicated control behavior from the operating regime of the flight controller. An example of how a non-monotonic mapping estimated in the SNLM is converted by the map handler and inverted into the control mapping, $\delta a_c = f_{MH}^{-1}(\dot{\psi}_c)$ is shown in Figure 5.9.

5.2.3 Example Autonomous Flight

The Hyperadaptive GNC is applied to the same degraded vehicle with higher than nominal control sensitivity as tested previously with the Conventional GNC in Figure 5.1. The added robustness (via the pre-initialization check) and adaptive nature of the controller enable the degraded system to effectively estimate the wind and land approximately 6m from the target as shown in Figures 5.10 and 5.11. The combined airspeed and LTID estimation period estimates airspeed within 3%, and due to the vehicle remaining within the turn rate bounds, this constant airspeed estimate well characterizes the entire flight. The LTID estimation of the model parameters in Figure 5.11(a) show that the turn sensitivity is initially characterized as bias until both left and right turns are commanded and the values jump to near their final value. Additionally, the SNLM accurately tracked the identified behavior as shown in Figure 5.11(d). The task was relatively trivial in this case as the linear approximation generated during estimation of the linear dynamics also well characterized the predominantly linear control mapping. The primary cost of this improved robustness to damage is an initialization phase which uses almost 200m of altitude, twice that of the Conventional GNC.

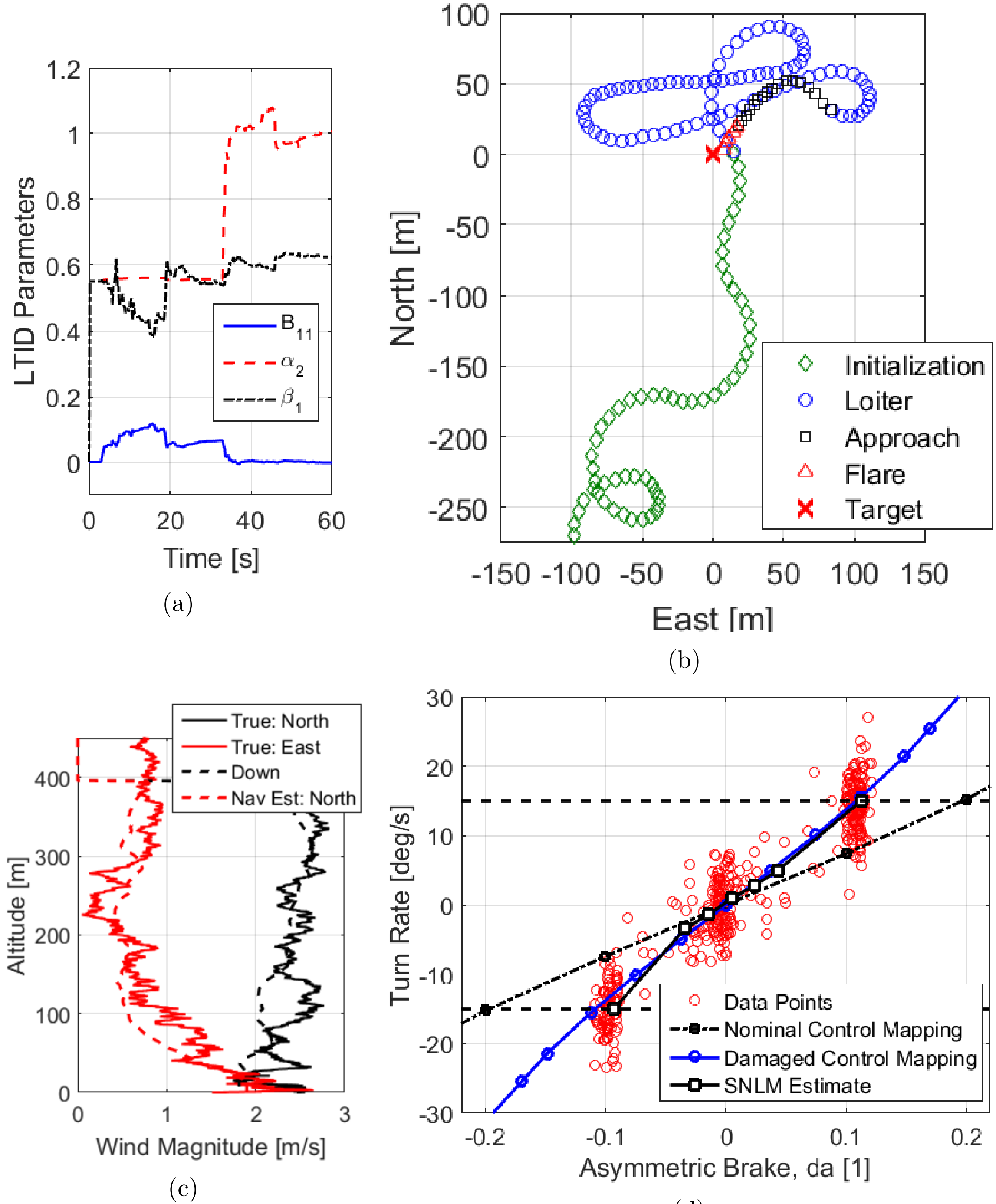


(a)



(b)

Figure 5.10: Simulated flight of the Hyperadaptive GNC steering a vehicle with increased control sensitivity.



(a)

(b)

(c)

(d)

Figure 5.11: Simulated flight of the Hyperadaptive GNC steering a vehicle with increased control sensitivity.

CHAPTER VI

A COMPARISON OF ADAPTIVE GNC ALGORITHMS

The goal of this chapter is to analyze the impact of damage on the landing accuracy of several precision airdrop guidance, navigation, and control algorithms. A series of degraded conditions are imposed upon the vehicle which deviate the performance away from the nominal conditions. Small deviations such as turn rate bias to extreme degradation such as control reversals and different left and right turn rate performance are induced on the vehicle. Extensive simulation results are presented for the hyper-adaptive GNC in addition to two algorithms used commonly in the field of guided airdrop systems which present the current state-of-the-art autonomous algorithms. Flight test results using the small-scale airdrop system are presented to validate the proposed algorithm and conclude the chapter.

Two comparator GNC algorithms are studied to directly compare the hyperadaptive GNC presented in Chapter 5 to currently fielded flight software. The first is the Conventional GNC presented in Section 4. While limited in its adaption, this algorithm estimates vehicle airspeed and turn rate bias along with the horizontal components of the atmospheric wind. The initialization estimation algorithm mitigates the impact of changing airspeed on the Navigation algorithm which can exhibit poor tracking or even divergent results when airspeed error is introduced. The second algorithm, termed the XFly GNC, upgrades the turn rate PI controller with a Model Reference Adaptive Controller (MRAC) to estimate both turn rate bias and linear control sensitivity. This method is expected to handle more types of damage than the conventional GNC but can only linearly approximate the turn rate response. More details on this method, including the derivation of the MRAC controller, are presented in Appendix A. An overview of the adaptive capabilities of each algorithm is presented in Table 6.1.

Table 6.1: Adaptive capabilities of three GNC algorithms.

Adaptive Capability	Conventional	XFly	Hyperadaptive
Airspeed	✓	✓	✓
Horizontal atmospheric wind	✓	✓	✓
Turn rate bias	✓	✓	✓
Turn rate sensitivity (linear)	-	✓	✓
Turn rate sensitivity (nonlinear)	-	-	✓
Control reversal	-	-	✓
Speed of Dynamic response	-	-	✓

To analyze the capabilities of the aforementioned adaptive GNC algorithms, a series of example trajectories in the presence of damage build off of the initial simulations presented thus far. After, an extensive trade study into the impact of different damage metrics are system performance is characterized. This chapter concludes with flight test results using the small-scale system.

6.1 Example Damage Cases

Details on the strategies and robustness of each algorithm is presented through simulation testing of four degraded flight conditions. The first linearly scales the vehicle turn rate sensitivity as depicted in Figure 6.1(a). This has been presented for the Conventional and Hyperadaptive GNC algorithms in Figures 5.1 and 5.10, respectively. The second example considers a change in control sensitivity that only effects one side of the turn rate mapping, shown in Figure 6.1(b). This asymmetric change in turn rate response broadly characterizes types of damage that only impact one side of the canopy or one actuator. The third damage case studies the impact of a control reversal on the adaption laws as it can have destabilizing results. Finally, the fourth

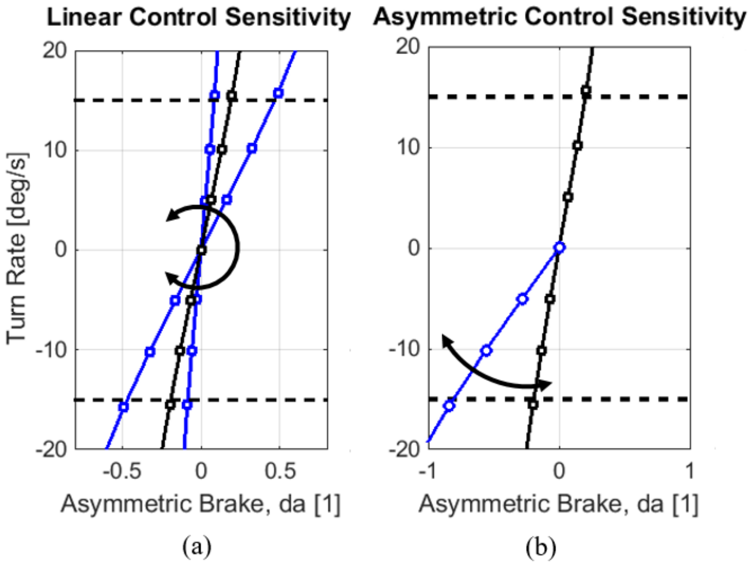


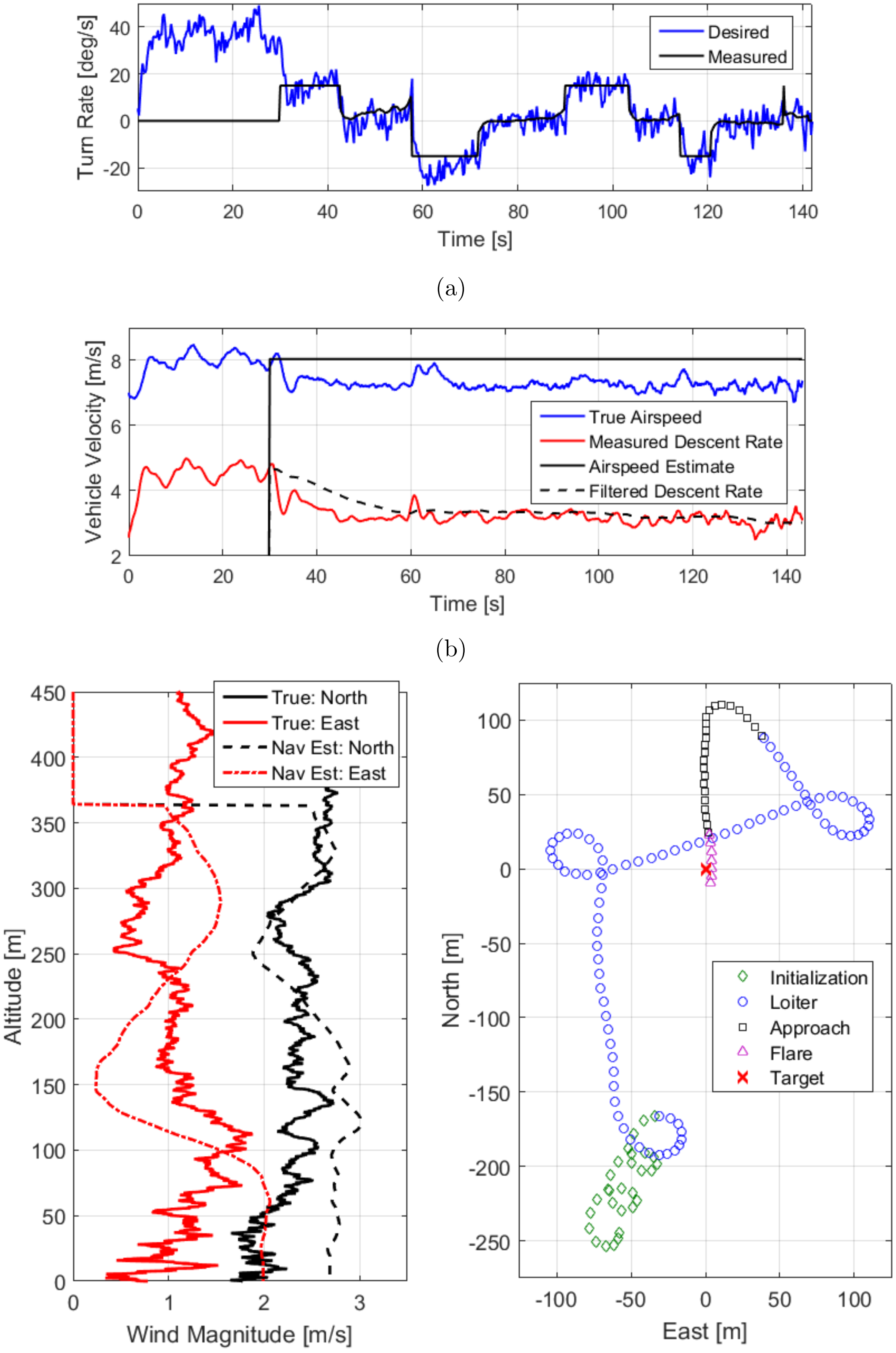
Figure 6.1: Depictions on how the control authority can be rotated (a) symmetrically and (b) asymmetrically to represent various types of degraded flight conditions.

damage case explores changes in the speed of the dynamic turn rate response which must be accommodated by adaptive path planning.

For brevity, focus is placed on the XFly and Hyperadaptive GNC algorithms as results for the Conventional GNC show consistent trends in potential for poor initialization and difficulty tracking the desired heading rate due to errors in the *a priori* control mapping (Figure 5.1).

6.1.1 Linear Control Sensitivity

This section presents the results of the XFly GNC algorithm applied to a vehicle with a change in linear control sensitivity to accompany the previous analysis for the Conventional and Hyperadaptive GNC algorithms in Figures 5.1 and 5.10, respectively. This class of damage can result from issues such as incorrect payload weight or a new canopy/actuator configuration. Simulation results for the XFly GNC algorithm is provided in Figures 6.2 and 6.3 for the same damage conditions. At the start of the flight, the open loop initialization phase commands the nominal input, generating a high turn rate of 30 deg/s, double the maximum desired turn rate of the vehicle. The



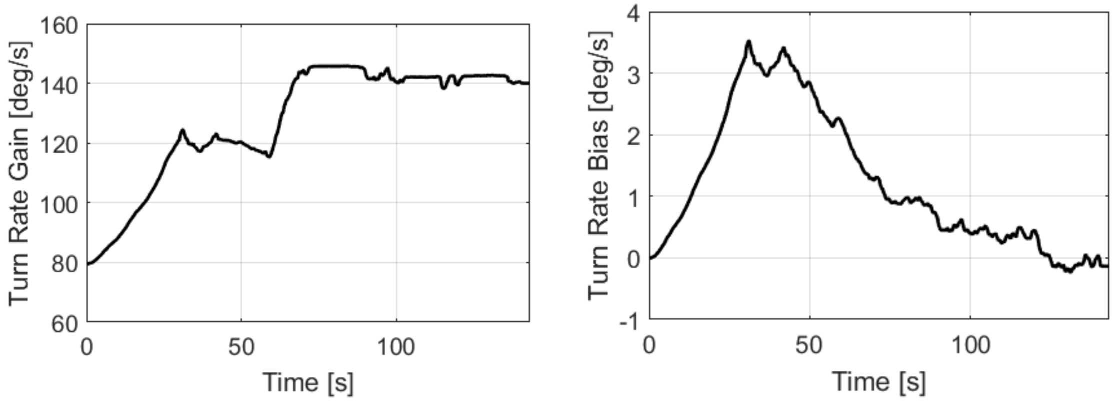
(a)

(b)

(c)

(d)

Figure 6.2: Simulated flight of the XFly GNC steering a vehicle with increased control sensitivity.



(a) (b)

Figure 6.3: Simulated flight of the XFly GNC steering a vehicle with increased control sensitivity.

high turn rate of the initialization period deviates from the desired flight regime due to a lack of a pre-initialization phase and results in a 10% error in airspeed estimation. Slow adaption of the MRAC gain is allowed during initialization but does not change the commanded signal. Once the vehicle turns in both directions (starting at $t \approx 65$ sec), the MRAC parameters converge to near their final value and the tracking error is minimized, enabling the vehicle to accurately track Dunbins paths and land only 16 meters from the target.

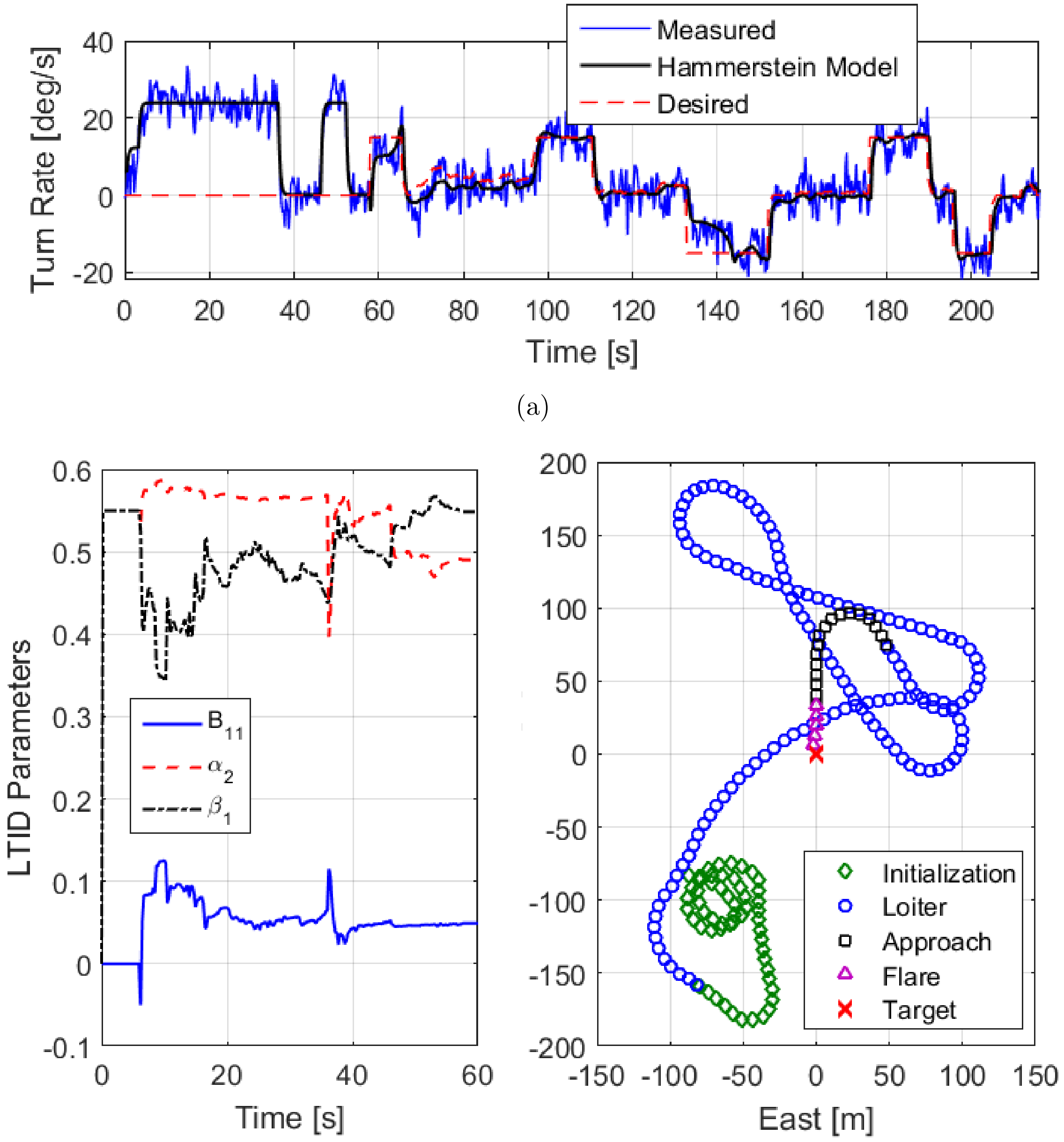
Overall, both the XFly and Hyperadaptive GNCs are able to adapt to this type of error quickly and through very different approaches. The Conventional GNC however, showed poor tracking behavior as the ability to vertically offset the control mapping using δa_{bias} cannot account for the slope change of the turn rate response.

6.1.2 Asymmetric Control Sensitivity

The example case presented here is for a scenario when the control sensitivity to turn left is greatly reduced. This damage case generally characterizes degraded conditions that only significantly impact one side of the canopy such as the example damage presented in Figure 1.4(b). This is the first method that extends beyond the adaptive capabilities of both comparator GNC algorithms as indicated in Table 6.1. Even though the right turn mapping is nominal for the positive turn direction, the Conventional GNC can exhibit tracking error in both directions due to unstable adaption of the turn rate bias δa_{bias}.

The simulation results of the hyperadaptive GNC applied to this damaged vehicle model are presented in Figures 6.4 and 6.5. The guided airdrop system is initially placed upwind of the target following standard industry practice and uses the entire flight to estimate the control mapping and dynamic response. Figure 6.4(a) shows the effect of pre-initialization update at $t = 3$s where the control input is increased above the initially conservative command. With the initial turn rate between the desired limits ($\dot{\psi}_L > |\dot{\psi}_{init}| > \dot{\psi}_U$), the airspeed and LTID estimation algorithms accurately characterize the steady state longitudinal behavior and the lateral turn rate dynamics. Note that as a result of the control softening on the left side, the vehicle does not actually command a substantial left turn. This does not effect the estimation of the turn rate dynamics (associated with β_1), but the linear approximation of the SNLM. Figure 6.5(a) shows that the linear approximation is not representative of the actual mapping and only aligns near the commanded values used in LTID identification (pts. A and B). This error does not impact the final landing accuracy because poor tracking behavior at altitude is acceptable assuming there is sufficient altitude margin. Accurate tracking of the Dubins path is only crucial during the final stages of flight when the entire flight has been used to characterize the SNLM estimate.

After initialization, the time period from approximately $t = 60 \rightarrow 130$ s is used to update the SNLM on the side of positive control authority. This shifts the estimate of the SNLM down from the linear estimate to align with the measured response. When the vehicle first commands a left turn at $t \approx 130$ s, significant tracking error populates the SNLM estimator with points in the region of $(\delta a, \dot{\psi}) = (-.4, -8)$ deg/s as shown in Figure 6.5(b). This causes the SNLM estimate to update the mapping

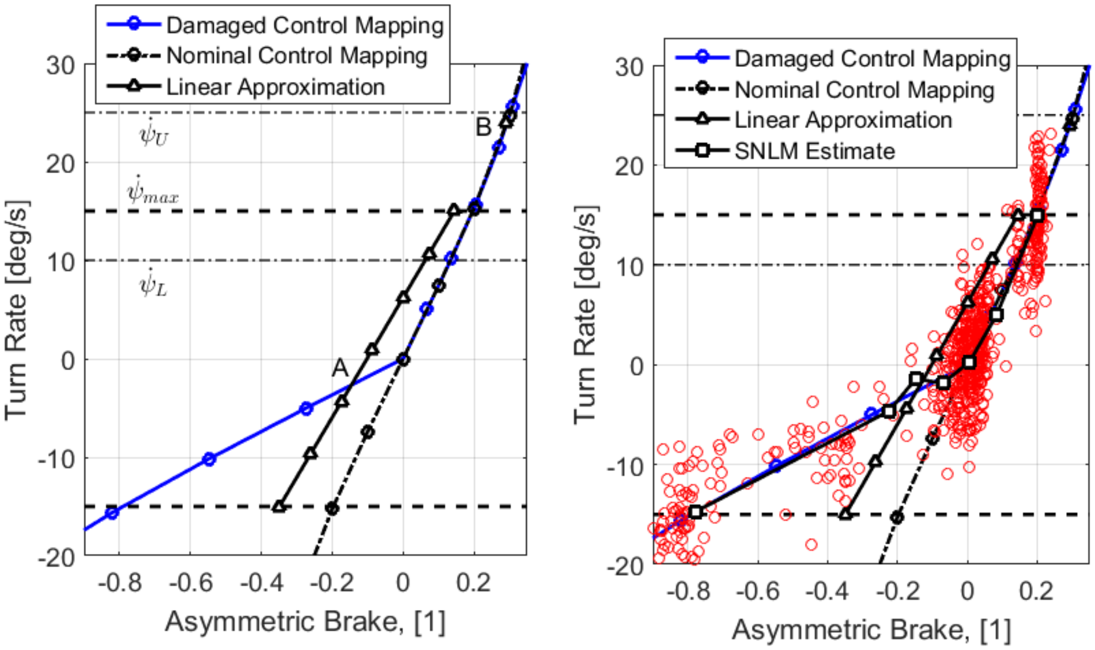


(a)

(b)

(c)

Figure 6.4: Simulated flight of the Hyperadaptive GNC steering a vehicle with asymmetric turn rate response.



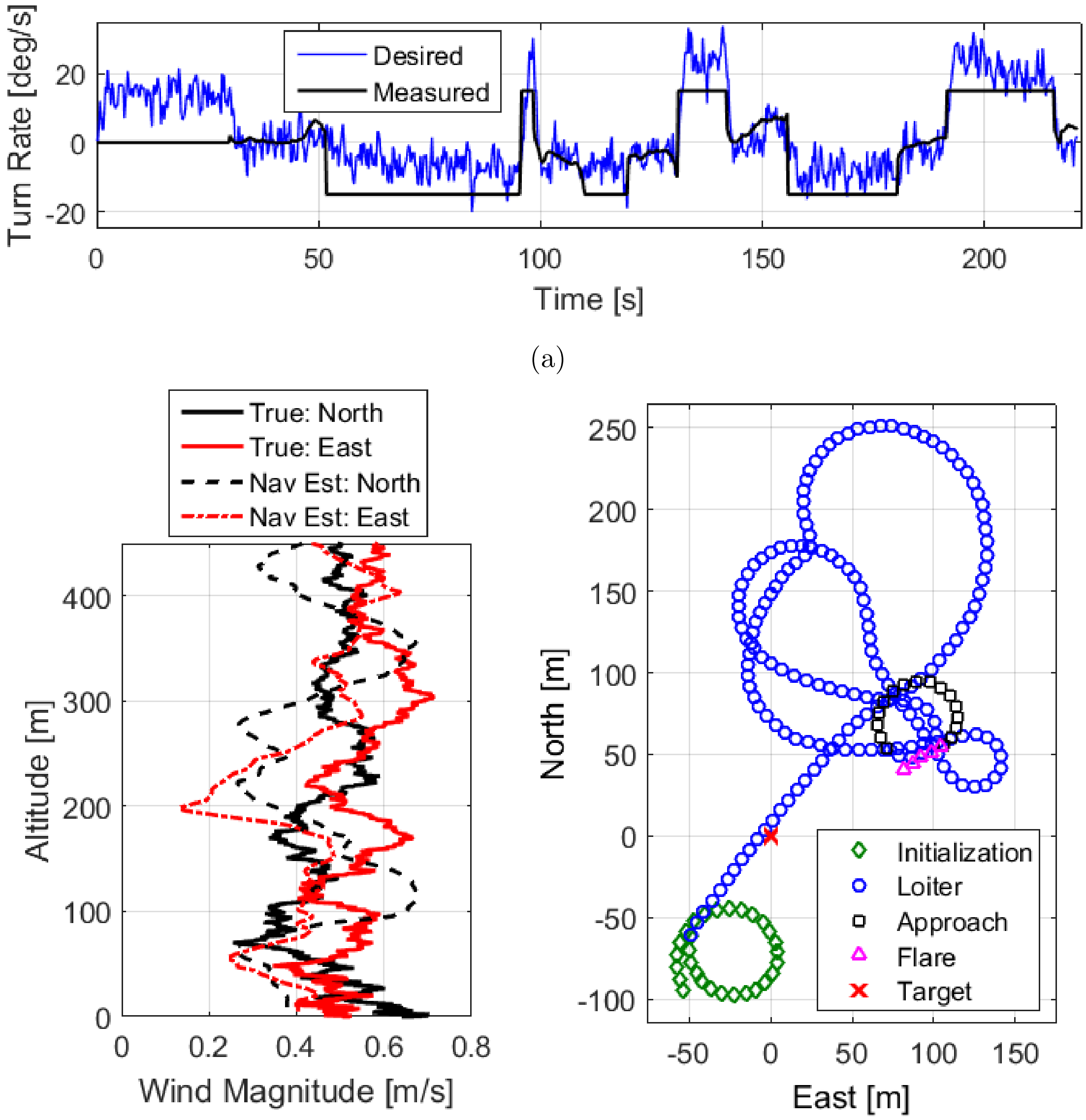
(a)

(b)

Figure 6.5: SNLM Estimate for a vehicle with asymmetric turn rate response after (a) estimation of the LTID parameters and (b) the conclusion of the flight.

towards the proper negative turn rate which is achieved in under 10 seconds. Independently updating each segment in the parameter space enables an accurate nonlinear approximation of the vehicle turn rate response that well approximates the applied nonlinear behavior.

In comparison, results of the XFly software is presented in Figures 6.6 and 6.7. After initialization, the vehicle only commands right turns and straight flight until $t \approx 95$s.At this point, the MRAC estimate accurately matches the nominal model which governs the positive turn rate response. The first left turn from approximately $t = 50 \rightarrow 95$s, exposes the nonlinearity. The adaption is forced to match the turn rate response with a single linear function causing the turn rate gain to decrease and turn rate bias increase. The final control mapping estimate shown in Figure 6.7(c) is a linear average of the damaged control mapping. This incorrect estimate shows poor tracking behavior as commanded left and right turns excite a -8 deg/s and +25 deg/s turn, respectively, which deviates from the assumption in Dubins path planning that all turns are conducted at $\dot{\psi}_{max} = 15$ deg/s. The resulting trajectory in Figure 6.6(d) shows the wide left turns and sharp right turns that cause the vehicle to land over 100m from the target.

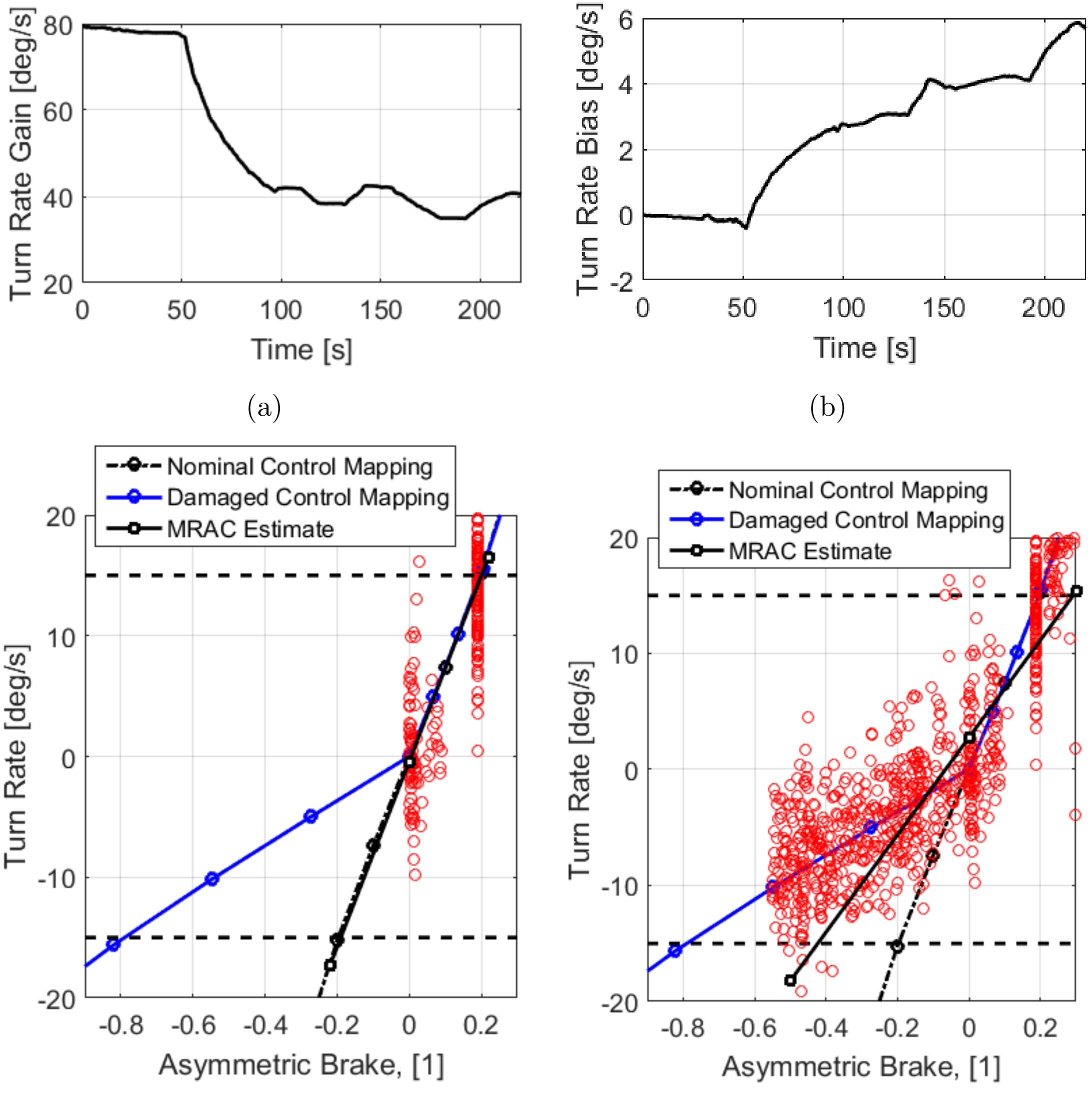


(a)

(b)

(c)

Figure 6.6: Simulated flight of the XFly GNC steering a vehicle with asymmetric control sensitivity.



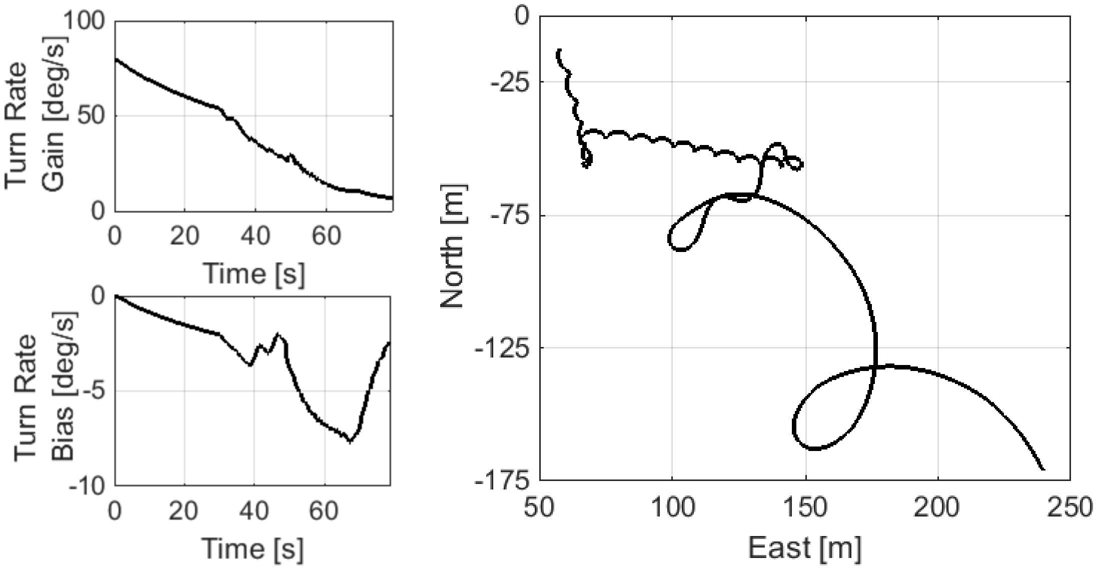
(a)

(b)

(c) $t \approx 50\text{s}$

(d) $t = end$

Figure 6.7: MRAC estimates generated by the XFly GNC steering a vehicle with increased control sensitivity.



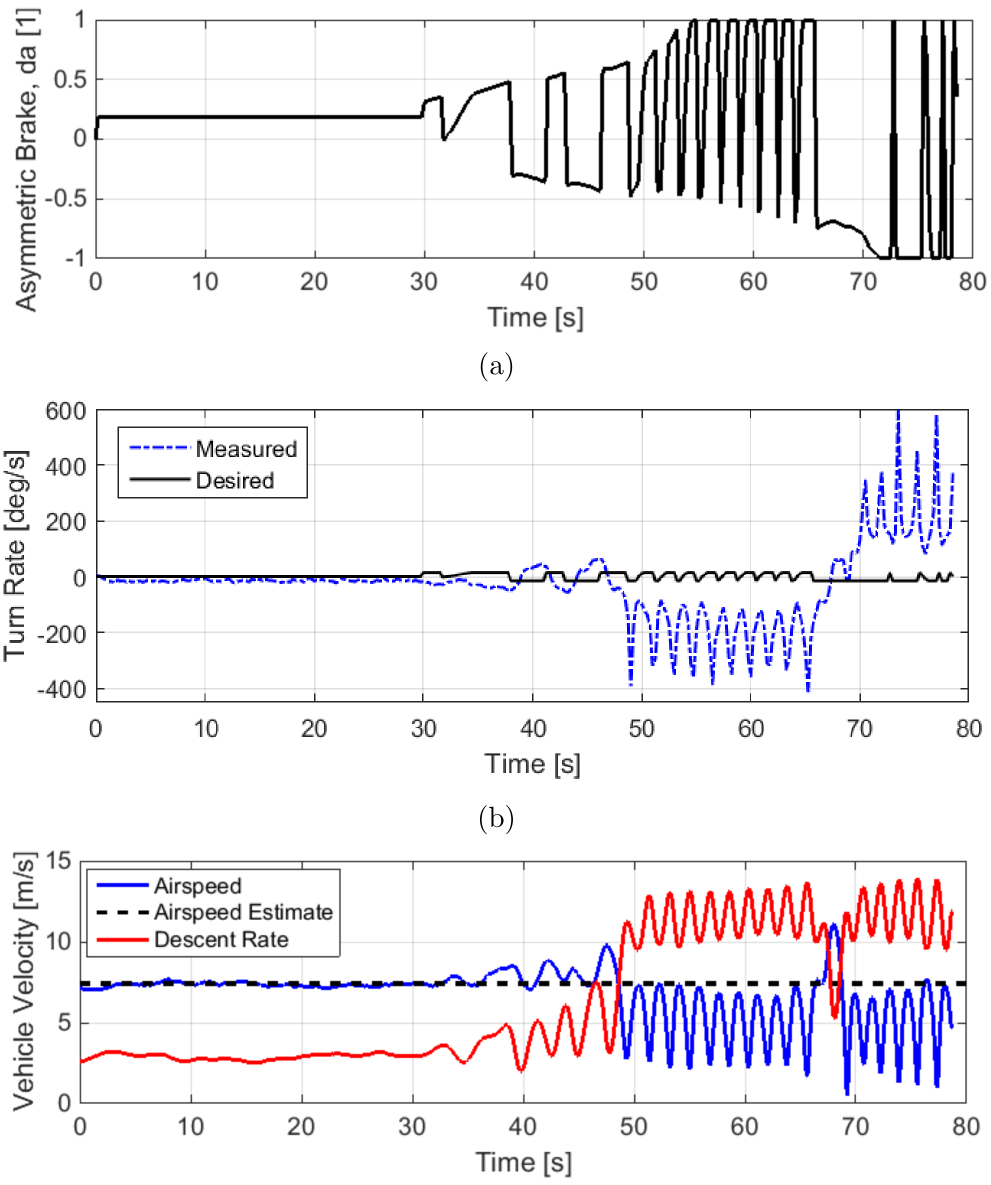
(a) (b)

Figure 6.8: Simulated trajectory of the XFly GNC acting on a vehicle with a control reversal.

6.1.3 Control Reversal

The issue of control reversal presents a large problem for many adaptive controllers. The MRAC in the XFly software adapts the control gains based on the sign of the control effort which is assumed to be positive. Figures 6.8 and 6.9 depict a simulation where the XFly GNC is used to control a vehicle with inverted control authority. The turn rate gain update Equation (A.9) is always negative as a result of the control reversal (excluding the effects of measurement noise). As a result, the estimate of the turn rate gain steadily decreases to zero as shown in Figures 6.8(a). This inversely causes the control sensitivity to grow unbounded and the actuators saturate. This induces a tight spiral that is effectively maintained until the system crashes in to the ground. This is characterized in Figure 6.9(c) when the descent rate jumps to approximately 12 m/s due to the lift vector of the canopy no longer being aligned to counter the effects of gravity.

One of the clear benefits to the hyperadaptive strategy is the independence from

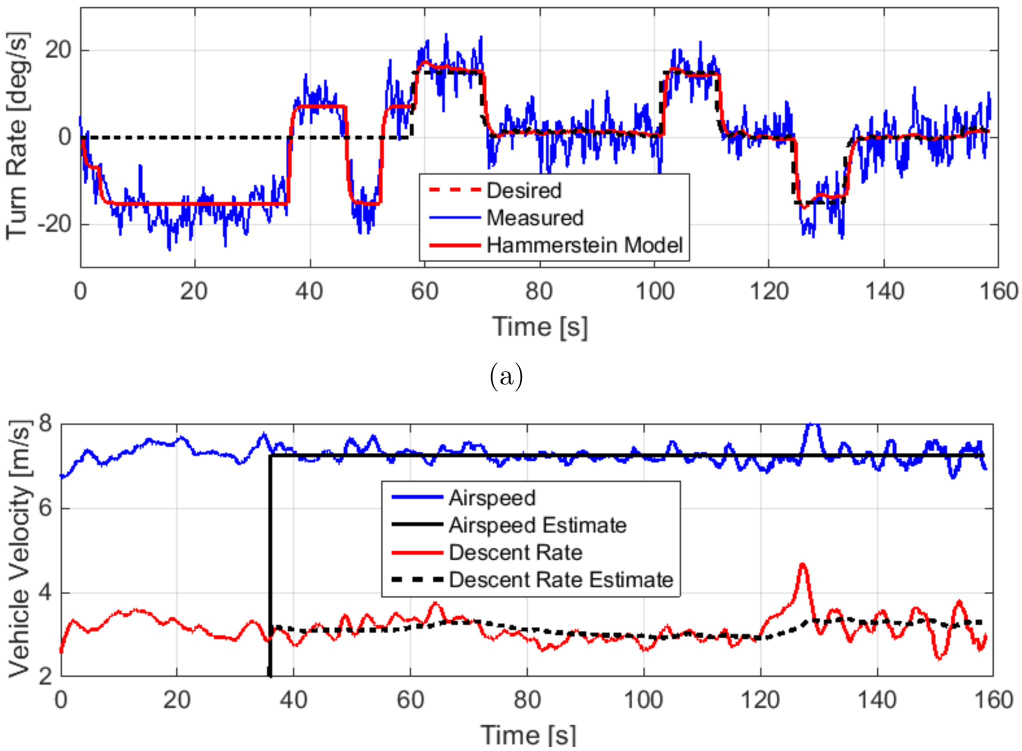


(a)

(b)

(c)

Figure 6.9: Simulated trajectory of the XFly GNC acting on a vehicle with a control reversal.

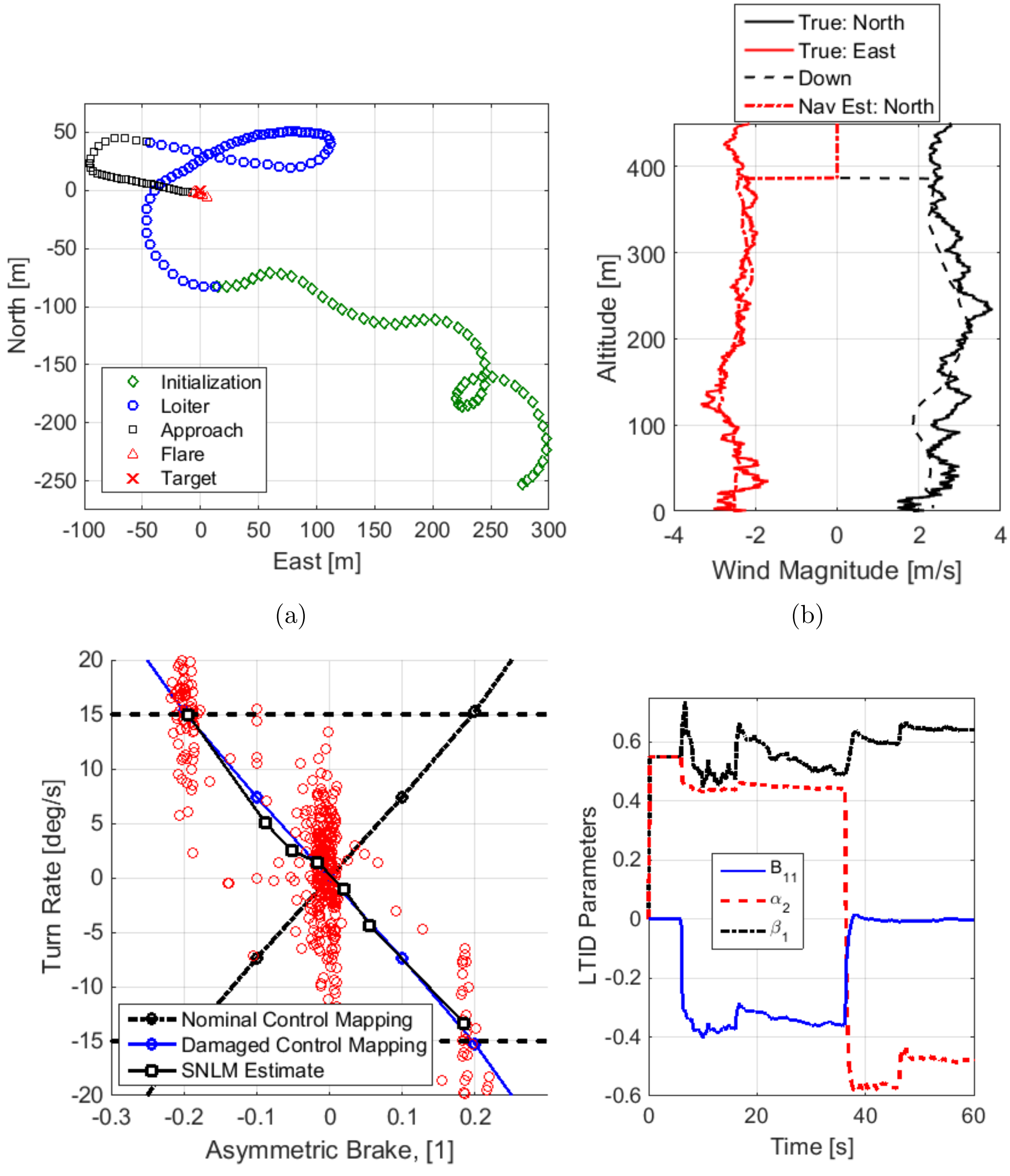


(a)

(b)

Figure 6.10: Simulated trajectory of the hyperadaptive GNC acting on a vehicle with a control reversal.

the sign of the control authority. This data driven approach is able to characterize the control reversal and land accurately at the target as shown in Figure 6.11 - 6.10. The LTID estimation characterizes the control reversal by estimating a negative turn rate gain (α_2) in Figure 6.11(d). The resulting linear approximation of the SNLM captures the control reversal and the SNLM estimator is used to further adapt to any potential nonlinear behavior. Additionally, the ability to accurately track the desired heading ensures the vehicle stays within the turn rate limits to accurately track airspeed, descent rate, and the horizontal components of the atmospheric winds.



(a) (b)

(c) (d)

Figure 6.11: Simulated flight of the hyperadaptive GNC steering a vehicle with inverted control sensitivity.

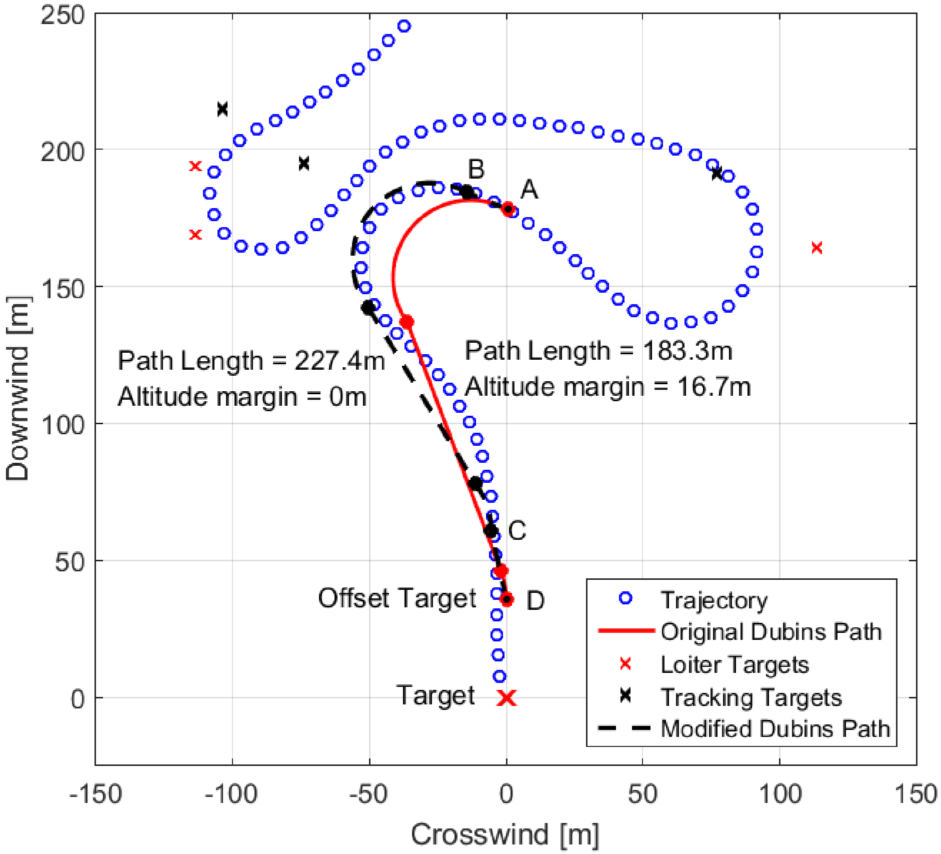


Figure 6.12: Simulation of the hyperadaptive GNC showing the modified Dubins paths used to calculate the flight distance to the offset target.

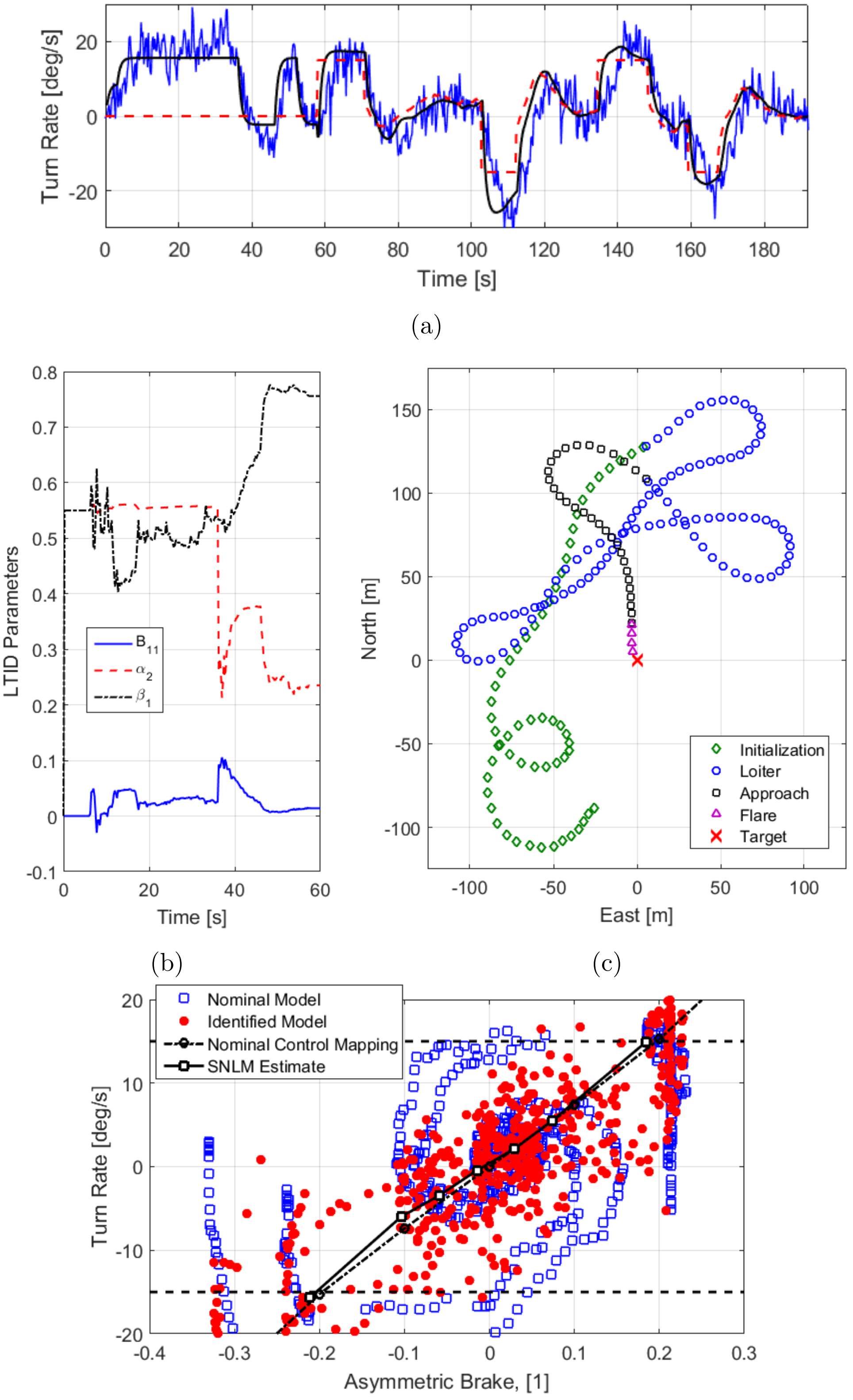
6.1.4 Slow Turn Rate Response

The final case focuses on estimation of the turn rate dynamics and the adaptive path planning algorithm in the hyperadaptive GNC. It is expected that the modified Dubins algorithm will provide more feasible paths for the vehicle to track. This is the most important when calculating the altitude margin to transition from loiter to approach. The altitude margin, h_{Margin} and altitude required to reach the offset target h_{Req}, which govern this transition directly depend on the path length, L. Any error in path length results in an altitude shortage or surplus when landing at the target. Figure 6.12 shows the original and modified Dubins paths used to estimate the flight distance from the current position at points A to the offset target (point D). Calculation of the altitude margin using Equation (6.1) and the modified Dubins path planning algorithm identified point A as the transition to approach. This used the estimated settling time ($T_s = 3.3$s) to identify a transition length of 18.2m between points AB and CD. The unmodified Dubins calculation was used to connect points B and C to estimate a total path length of 227.4m, only a 3% error from the actual path flown by the vehicle. The original Dubins algorithm is used to fly from point A to D which represents the non-adaptive calculation. The Dubins algorithm applied from points A to D estimate a path length of 183.3m, a 22% error from the actual path flown. This also results in a positive altitude margin of 16.7m implying that the system would not have transitioned to approach and would not have reached the target.

$$h_{Margin} = h - h_{Req} = h - L\frac{\dot{z}}{V_0} \tag{6.1}$$

Additional simulation results for this example trajectory are presented in Figure 6.13. Most notable is the SNLM estimator in Figure 6.13(d) which shows data points lotted using both the nominal system model and the identified LTID model. The phase difference between the nominal and true system cause arching streaks in the figure due to the phase lag of the output. By characterizing the LTID, the noise fed to the SNLM estimator is reduced, improving the turn rate estimate.

Overall, the hyperadaptive GNC is accurately able to capture many forms of degraded conditions by using a data driven approach. From the pre-initialization phase to the SNLM estimator and map handler, this algorithm is significantly more robust to the uncertainty frequently apparent in experimental flight testing of parafoil systems. The natural ability to adapt to control reversals and capture nonlinear effects of the control sensitivity give strong justification for this algorithm over the MRAC controller in the XFly GNC.



(a)

(b)

(c)

(d)

Figure 6.13: Simulated flight of the hyperadaptive GNC on a vehicle with slow turn rate response.

6.2 Parameter Trade Study

This section delves further into the relative capabilities of the three adaptive GNC algorithms by randomly varying a series of parameters to add high levels of model uncertainty. The goal is to stress test each algorithm to define a point at which damage makes an algorithm ineffective at achieving precision landing. Seven parameters are considered which cover a wide array of potential damage scenarios and influence both steady state and dynamic behavior of the precision airdrop system. Generally speaking, these forms of damage fall into two categories. The first are direct modifications to the simulation model to induce a change in flight behavior such as changing the canopy incidence angle to cause the system to fly with a different steady state airspeed and descent rate (Figure 3.7). The second are modifications to the turn rate sensitivity mappings which significantly effects the turn rate behavior of the vehicle.

A list of the damage parameters and their effect on the vehicle response is given in Table 6.2. Parameters 2 - 6 all manipulate the steady state turn rate response of the vehicle. Focus is placed here because the guidance structure of the autonomous algorithm also relies heavily on steady state response. Initial analysis of damage parameters 2-4 and 7 were used to present the example flight trajectories shown in the previous sections. Parameter 5 introduces a deadband in the control mapping of the vehicle. This nonlinear modification is presented in Figure 6.14 where the variables δa_{DB} and $\dot{\psi}_{DB}$ define the width and turn rate location of the deadband, respectively.

These linear and nonlinear modifications of the turn rate mapping are implemented by scaling the asymmetric brake command from the GNC to elicit the desired response from the simulation model. This is implemented in Equation (6.2) by calculating the desired damaged turn rate response $f_{Damage}(\delta a_c)$ and solving for the input to the simulation based on the known simulation control sensitivity, $f_{Sim}()$.

$$\delta a_{Sim} = f_{Sim}^{-1}(f_{Damage}(\delta a_c)) \tag{6.2}$$

This method provides the flexibility to study a large range of flight conditions which result from a variety of sources of actual damage. As a final note, this method was chosen for its general nature after confirming that it matched simulation results where damage was included by modifying the lift/drag characteristics of elemental strips of the canopy as done by Culpepper et al. [3].

Finally, the 7th parameter modifies the actuator dynamics to influence the speed of the turn rate dynamics. The typical fast response rate of the small-scale system is be slowed by over an order of magnitude to capture effects such as differences in wing loading, linear size of the system, or actuator degradation [26].

Table 6.2: List of parameters and their impact on the response of the dynamic model.

Parameter Varied	Induced Change in Model
1. Incidence Angle, Γ	Airspeed, V_0 & descent rate $\dot{z}$
2. Turn Bias, TR_0	$\dot{\psi}(\delta a = 0) \neq 0$
3. Sensitivity Scaling	Figure 6.1(a)
4. Asymmetric Sensitivity Scaling	Figure 6.1(b)
5. Deadband, δa_{DB} & $\dot{\psi}_{DB}$	Figure 6.14
6. Control Reversal	$\delta a_{Sim} = -\delta a_c$
7. Actuator Time Constant, τ_a	Speed of turn rate dynamics

6.2.1 Single-Parameter Variation

The effects of individual damage parameters on the ability of the adaptive GNC algorithms to steer the vehicle to the target is tested through a series of Monte Carlo trade studies. For each parameter, 250 simulated flights were run for each GNC to characterize the impact of a given uncertainty on the algorithm and the landing

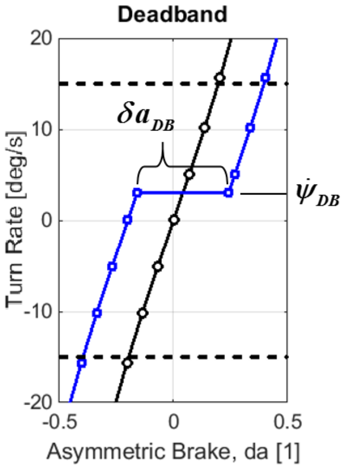


Figure 6.14: Nonlinear mapping that introduces a deadband of width δa_{DB} at a given turn rate $\dot{\psi}_{DB}$.

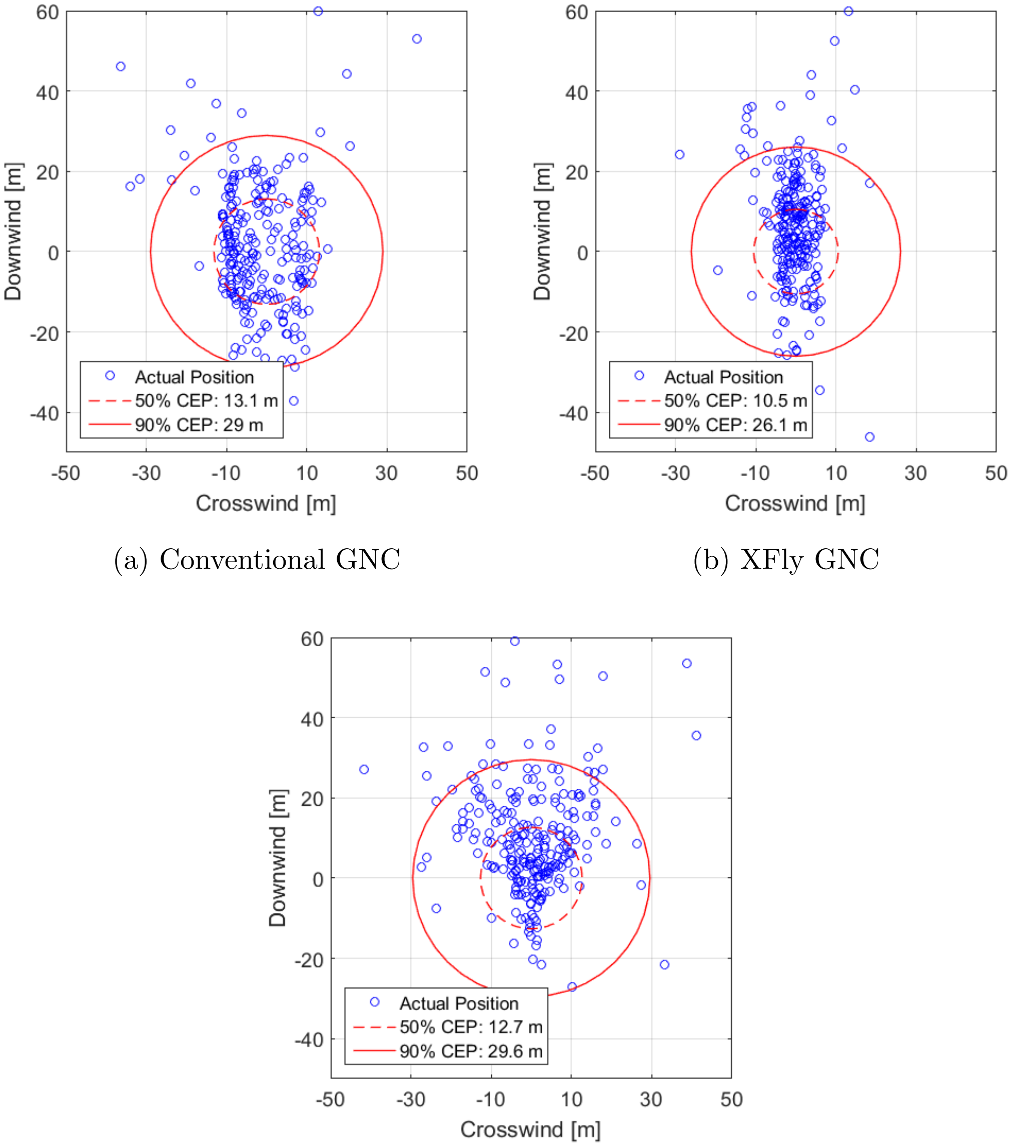
accuracy of the system. The simulations use an altitude independent constant wind field instead of the simple-shear wind field. This allows a better correlation between the damage parameters and landing accuracy of the vehicle since significant error can be introduced due to low altitude wind shears [93]. The Monte Carlo trade study randomly varies the wind magnitude, direction, and turbulence across each flight. The vehicle is positioned at an altitude of 500m and located upwind of the target following standard industry practice. The study also randomly varies the value of the damage parameter (except the control reversal which is a binary parameter) to study how the degree of damage impacts the landing accuracy.

The accuracy of precision airdrop systems is evaluated using the circular error probable (CEP). This metric is defined as the radius of a circle centered at the IP at which a given percentage of landings reside within. Primary accuracy is denoted by 50% CEP (equivalent to the median miss distance) and 90% CEP characterizes the extent of impacts with large miss distances. Landing statistics for each damage case is presented collectively at the end of this section in Table 6.3.

6.2.1.1 Nominal

The nominal case is presented first to define a baseline accuracy of each algorithm on a vehicle without damage. Results of the Monte Carlo simulation are plotted in Figure 6.15 where the dots represent the landing location of each flight. The algorithms all show similar results with 50% CEP values of 13.1m, 10.5m and 12.7m for the Conventional, XFly, and hyperadaptive GNC algorithms, respectively. It is important that all algorithms work similarly under nominal conditions as it is ideally the most common flight scenario. Note that these CEP values are relatively low for this vehicle as a result of the simplified mean wind field.

Figure 6.15 does show a slight difference in landing pattern of each algorithm. The Conventional and XFly dispersions show a close alignment to the downwind axis implying the algorithms consistently reach the offset target before flying into the wind and landing at the target. A slightly wider dispersion in the crosswind direction is noted for the hyperadaptive GNC due to two primary factors. First, measurement noise can cause slight errors in the SNLM estimate, particularly near zero where the map is allowed to be most flexible. This can introduce a small tracking error in the final stages of flight which causes the system to miss in the crosswind direction. Second, the hyperadaptive guidance algorithm is based on many parameters characterized in-flight instead of being predefined through extensive tuning (such as the location and altitude of the offset target). As a result, error in parameter estimates is propagated more heavily through the flight. However, the total impact of these issues do not represent a significant change in accuracy and while they may slightly degrade the nominal performance, they provide the robustness necessary to handle large amounts of model uncertainty.



(a) Conventional GNC

(b) XFly GNC

(c) hyperadaptive GNC

Figure 6.15: Landing dispersions of three adaptive GNC algorithms controlling the nominal vehicle model.

6.2.1.2 Incidence Angle

When incidence angle is used to change the steady state airspeed and descent rate of the vehicle, all algorithms can adapt to the full range of airspeeds commanded (approximately ±1 m/s). The algorithms had a 50% CEP of 13.2m, 9.7m, and 13.2 m for the Conventional, XFly, and hyperadaptive algorithms respectively. This indicates a negligible difference in comparison to the nominal vehicle indicating that all methods are able to reject the effects of uncertain airspeed on the landing accuracy. Note that this is achieved by the airspeed estimation algorithm including all methods. Without this estimation algorithm, errors in airspeed can destabilize the navigation algorithm.

Figure 6.16 shows each algorithm with a generally flat response to changes in the nominal airspeed as expected. More interesting is the error associated with the wind estimate and airspeed expected at the specified incidence angle. Figure 6.17 plots the airspeed error as $|V_0^{nv} - V_0(\Gamma)|$ which indicates that the hyperadaptive algorithm is the most prone to airspeed error which also has the greatest impact on landing accuracy. The higher variation in airspeed estimates is due to the initialization occurring at turn rates between 10 and 25 deg/s (defined by the pre-initialization algorithm) instead of the finely tuned 15 deg/s of the Conventional and XFly algorithms. This error has a greater impact on the final landing accuracy for the hyperadaptive algorithm due to guidance calculations using more estimated data instead of specifically tuned parameters.

6.2.1.3 Turn Rate Bias

The next damage scenarios induces constant turn rate biases in the flight behavior of the vehicle. As expected, effects of turn rate biases also present a relatively minimal challenge to all algorithms. Each method explicitly attempts to characterize the turn rate biases as they are one of the most common errors. The Conventional and

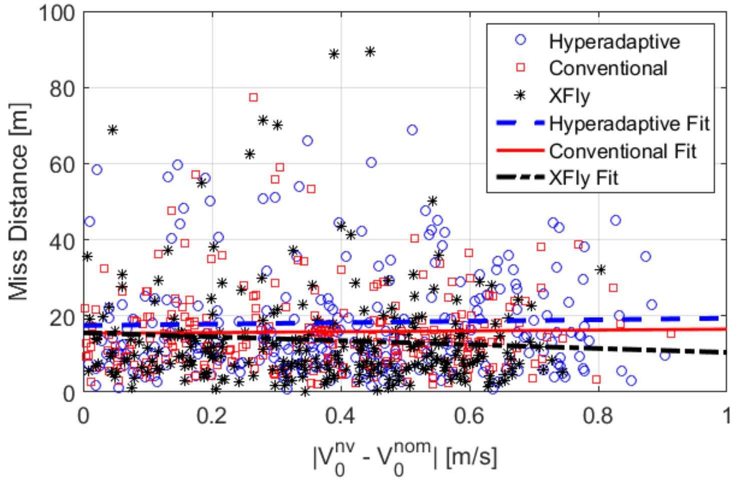
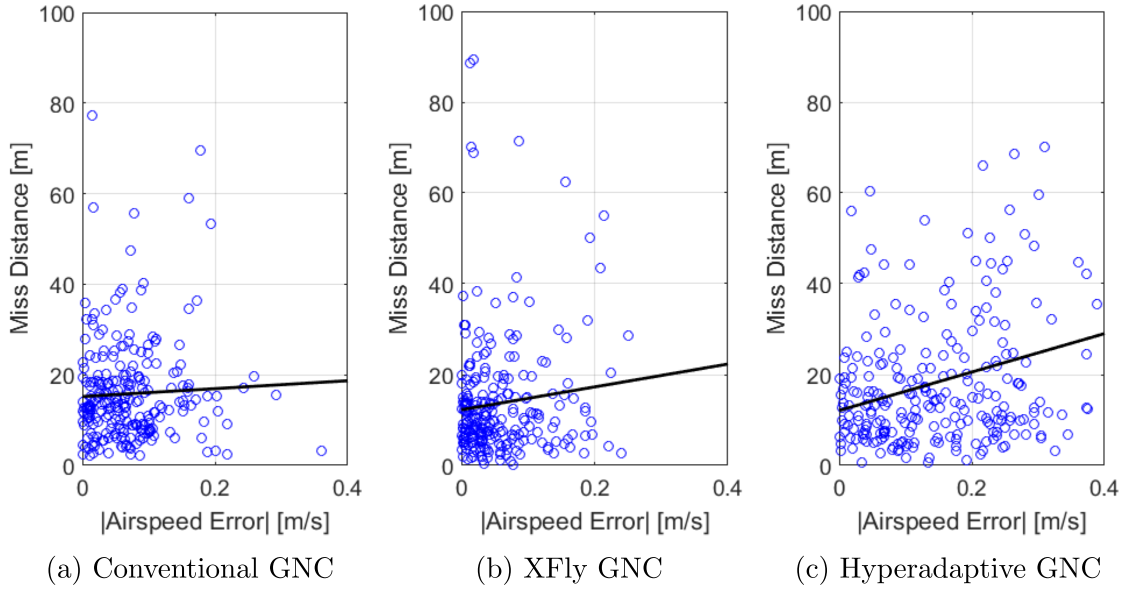


Figure 6.16: Landing accuracy dependence of three adaptive GNC algorithms on the vehicle airspeed uncertainty

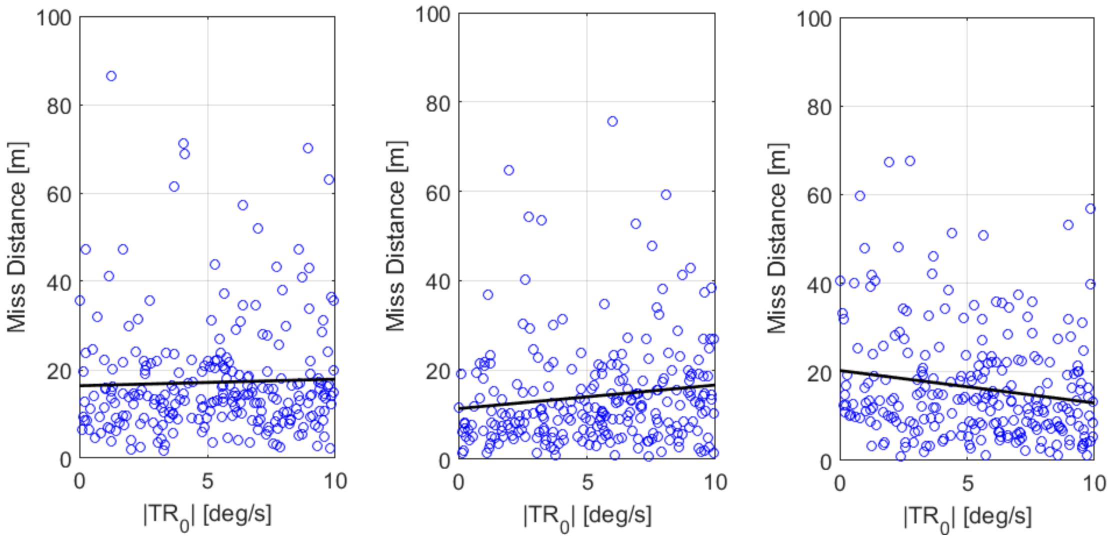


(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

Figure 6.17: Landing accuracy dependence of three adaptive GNC algorithms on the accuracy of airspeed estimate.



(a) Conventional GNC (b) XFly GNC (c) Hyperadaptive GNC

Figure 6.18: Monte Carlo results characterizing the relationship between imposed turn rate biases and the landing accuracy of each GNC algorithm.

XFly algorithms do this directly by characterizing δa_{bias} and the hyperadaptive GNC through the re-seeding of the SNLM $delta_a$ values estimate to ensure the map captures effects around $\dot{\psi} = \{0, \pm\dot{\psi}_{max}\}$. Landing results also indicated almost no increase in median miss distance over the nominal system with the 50% CEP results of 14.1m, 10.6m, and 13.2m for the Conventional, XFly, and hyperadaptive GNC algorithms.

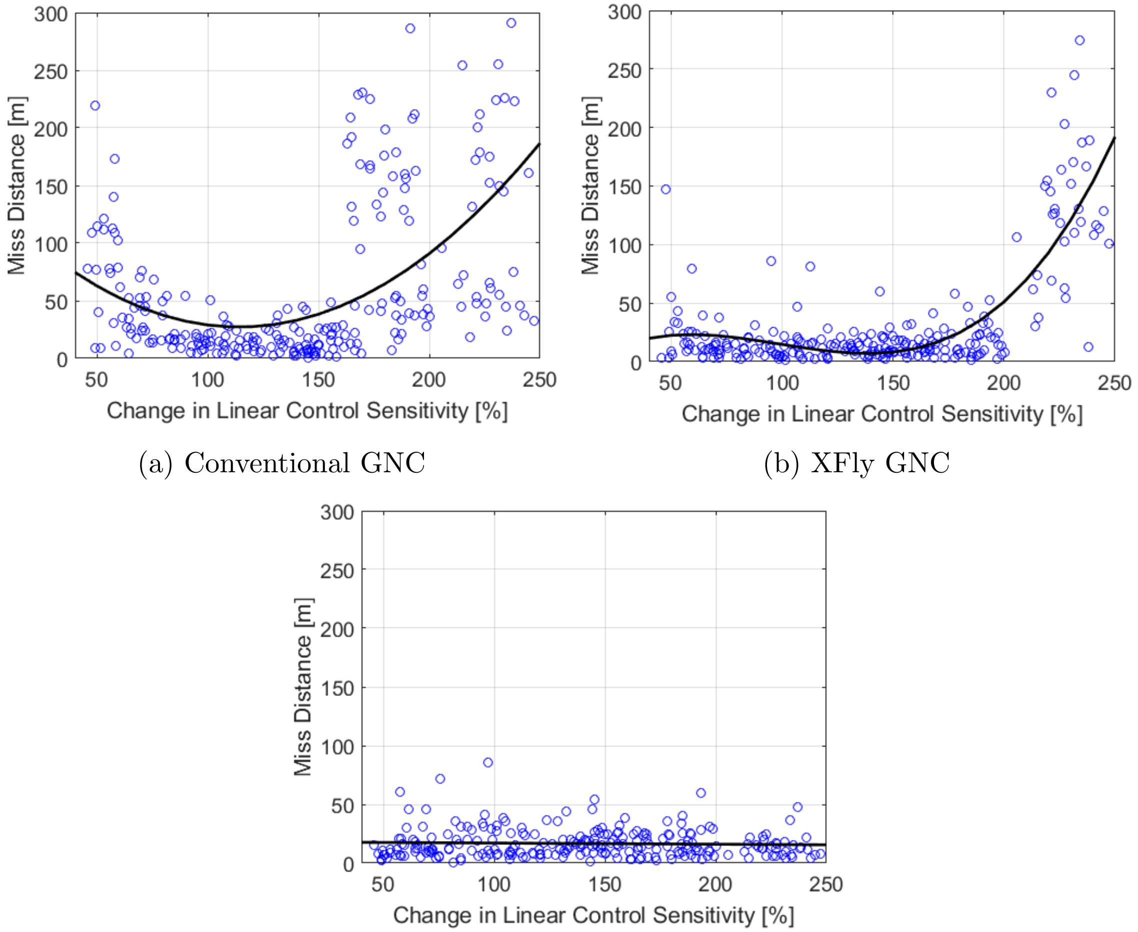
6.2.1.4 Control Sensitivity (linear)

This damage scenario presents the first case where the accuracy of the adaptive GNC algorithms can start to degrade. The limits to the linear scaling of the control sensitivity are confined between a 50% softening and a 250% stiffening of control authority. Figure 6.19 illustrates the trends in landing accuracy observed when the control sensitivity is scaled. First, the conventional GNC has a clear trend that landing error increases with increased deviation from the nominal mapping. However, we see that the band of acceptable uncertainty for this GNC extends from 75% to 150% of the nominal control sensitivity. This indicates that the Conventional algorithm is more susceptible to a decrease in control authority than an increase, as long as it isn't increased to the point where the initialization breaks down (starting at approximately 170%). The XFly algorithm is able to maintain a high level of accuracy across most of this parameter space but fails at high sensitivity due poor initialization. The pre-initialization phase is used to remove this influence from the hyperadaptive algorithm and it works effectively to have consistent landing accuracy, independent of the linear control sensitivity.

Aggregate landing statistics for the linear change in control sensitivity characterized 50% CEP values of 26.9m, 14.4m, and 13.3m for the Conventional, XFly, and hyperadaptive GNC, respectively. These values represent a 105%, 34%, and 5% decrease in landing accuracy. More dramatic is the increase in 90% CEP due primarily to high control sensitivity. 90% CEP increased from the nominal configuration 483%, 308%, and 5% for the three algorithms, respectively. The combination of the pre-initialization phase and adaptive control mapping provide the hyperadaptive algorithm with the robustness to be insensitive to the control effectiveness of the vehicle.

6.2.1.5 Asymmetric Control Sensitivity

The asymmetric change in control sensitivity introduces nonlinear behavior to the turn rate control sensitivity by linearly scaling only one side of the control mapping (chosen randomly). An initial analysis of this damage type was presented in Section 6.1.2 for each algorithm. This is expanded by randomly varying the amount the control sensitivity changed, varying from 25% to 100% of the nominal control sensitivity. Only softening of the control response is considered (implying more control input is needed for a given output) because the previous section characterized that the primary issue of increased sensitivity is failed initialization which has been sufficiently analyzed. At 25% turn rate sensitivity, a command 4 times greater than nominal is



(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

Figure 6.19: Monte Carlo results of three adaptive GNC algorithms controlling a vehicle with uncertain control sensitivity.

necessary to produce the desired turn while at 100%, the nominal, linear turn rate response is generated.

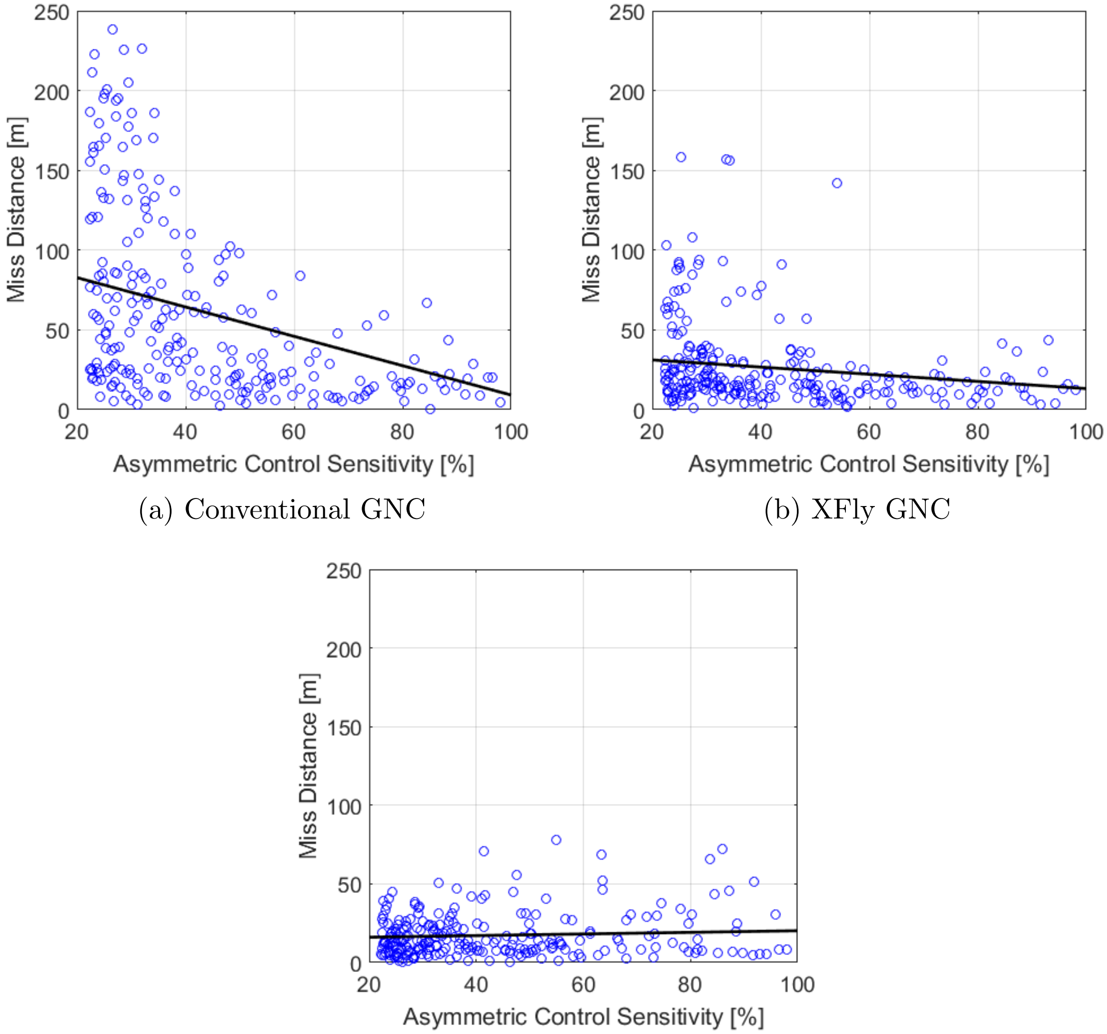
Figure 6.20 depicts the dependence of landing accuracy on the amount of nonlinearity introduced to the system. For the Conventional GNC, the decreased turn rate response significantly impaired the ability to track the desired heading. At large levels of output softening ($<$ 40% nominal) the controller can go unstable and induce spiral behavior. Large changes in the estimate of the control bias attempts to align the nominal map with the current measured response. This only attempts to match the current turn rate sensitivity which can exacerbate errors when the control input is changed, potentially exciting spiral behavior. The XFly software starts to show significant issues below 50% asymmetric control softening as a result of being unable to represent the nonlinear behavior. Finally the hyperadaptive algorithm is independent to this form of model uncertainty implying that the SNLM estimator sufficiently captures the nonlinear behavior to ensure landing accuracy.

Aggregate landing results from the trade study indicate that the Conventional, XFly, and Hyperadaptive GNC algorithms have 50% CEPs of 37.5m, 17.1m, and 12.8m, respectively. The number of large misses also increased for the first two algorithms indicating their lack of robustness to this form of model uncertainty.

6.2.1.6 Control Reversal

Control reversals present a significant challenge to almost all feedback systems. Without proper characterization, this induces unstable lateral performance and the vehicle often enters an uncontrollable spiral into the ground as shown in Figure 6.9 for the XFly GNC. This case is not uncommon in practice because human induced errors can cause the rigging lines for the trailing edge brakes to be connected to the opposite actuator.

The landing dispersion for each GNC applied to a vehicle with a control reversal



(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

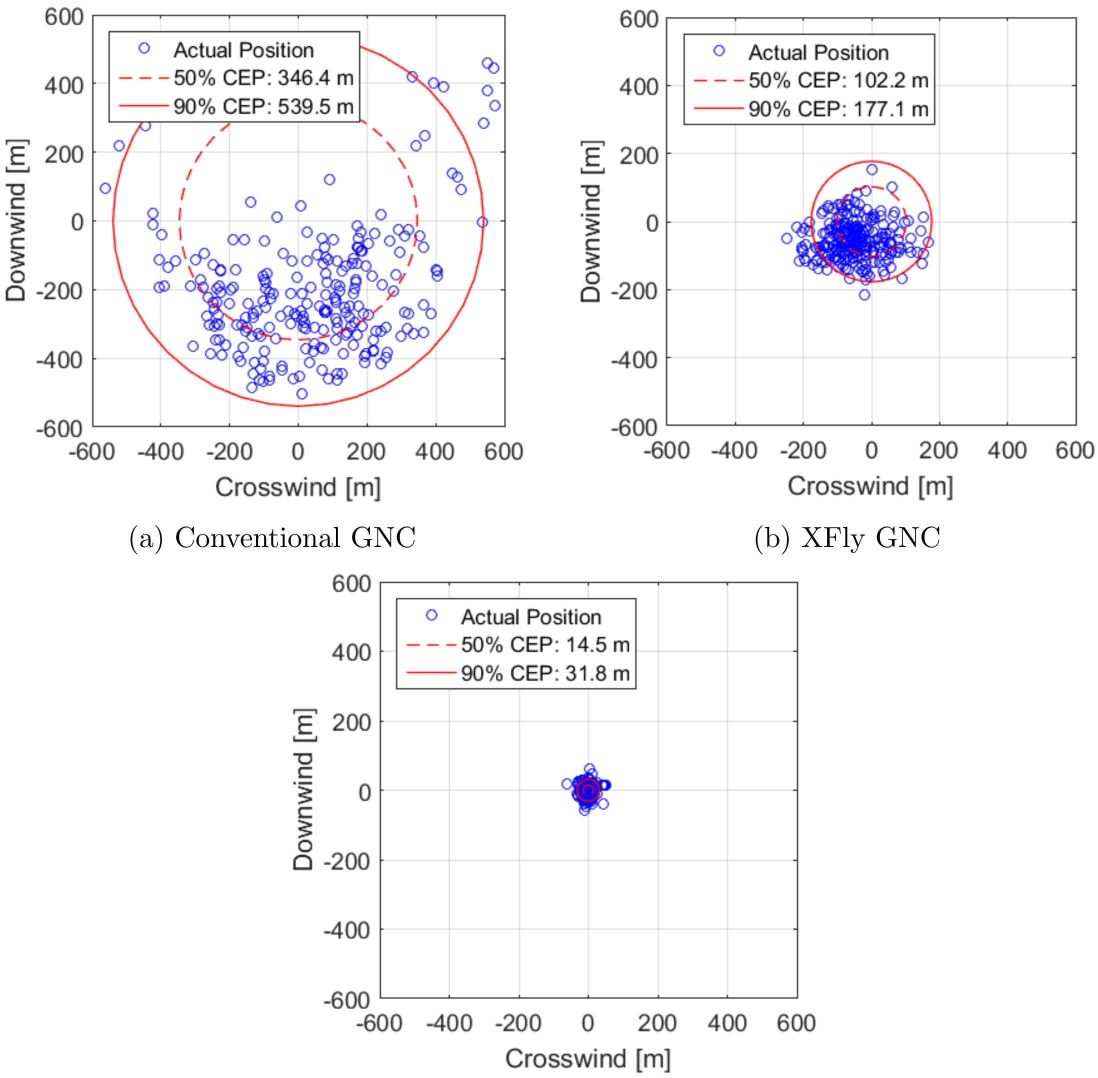
Figure 6.20: Monte Carlo results of three adaptive GNC algorithms controlling a vehicle with uncertain asymmetric left and right control sensitivity.

is presented in Figure 6.21. Over an order of magnitude increase in miss distance was noted for both the Conventional and XFly GNC algorithms with 50% CEP values of 346.4m and 102.2m, respectively. The control reversal also greatly effected large scale misses with 90% CEP values of 539.5m and 177.1m. The Hyperadaptive algorithm still maintains nominal accuracy with a 50% CEP of 14.5m and a 90% CEP of 31.8m. Additionally, the XFly results can be deceiving as the controller quickly destabilizes and enters into a spiral, drifting quickly down to the ground. Due to the constant mean wind field and an initialization upwind from the target, the unstable vehicle naturally tends towards the target. In more realistic conditions with shifting wind fields, the landing accuracy is expected to decrease further.

6.2.1.7 Deadband

The deadband nonlinearity is the final damage case that modifies the lateral control sensitivity. This is an important damage configuration to study as it removes the control effectiveness over a given range of asymmetric brake inputs. It also represents the mapping generated by the map handler if the SNLM estimator characterized non-monotonic behavior. This presents two primary obstacles. First, the greater the length of the deadband, the more nonlinear the mapping becomes. Second, there is a delay in response associated with crossing the deadband that makes it difficult to track a heading rate near $\dot{\psi}_{DB}$.

The simplicity of the Conventional GNC provides significant insight into the difficulties of controlling precision airdrop vehicles with deadband behavior. Figure 6.22(a) characterizes the effects of the deadband function on landing accuracy of the vehicle. The contour plot is based on the landings within the 90% CEP range to characterize the effect of the deadband. Flights with greater miss distance are represented in Figure 6.22(a) by black asterisk markers which are dominated by scenarios in which



(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

Figure 6.21: Monte Carlo landing dispersions characterizing the effects of a control reversal on three adaptive GNC algorithms

the deadband consumed the entire nominal control mapping of the conventional algorithm. Results indicate that the landing accuracy is most impacted by deadbands located near $\dot{\psi} = 0$, decreasing the effectiveness of the controller during approach which is the most vital time to be accurate. Increasing δa_{DB} also increased the miss distance due to more nonlinear effects. When the deadband is applied above 8 deg/s, the majority of the turn rate mapping is continuous (from the opposite maximum turn direction to the deadband) and the bias estimator can update δa_{bias} to shift the *a priori* turn rate mapping to align with the majority of the deadband nonlinear control mapping, improving landing accuracy.

The impact of deadbands on the XFly and Hyperadaptive GNC algorithms is more difficult to quantify as it doesn't sufficiently induce errors greater than the baseline error. Results of Monte Carlo analysis on all algorithms found that the Conventional, XFly, and Hyperadaptive GNC algorithms have 50% CEPs of 45.1m, 18.1m, 14.9m, respectively. All methods suffer from deadbands located near $\dot{\psi} = 0$, regardless of whether they are characterized by the adaptive algorithm or not. Analysis of the results identified that the primary improvement of the Hyperadaptive over the XFly GNC is improved crosswind miss distance as shown in Figure 6.24. The crosswind miss is most closely related to the effectiveness of the lateral heading controller which aligns towards the IP during approach. The ability of the Hyperadaptive GNC to most effectively match the damage turn rate mapping is shown in Figure 6.23. The improved characterization of the deadband enables the controller to more quickly move through the region, limiting the adverse effects on the heading controller.

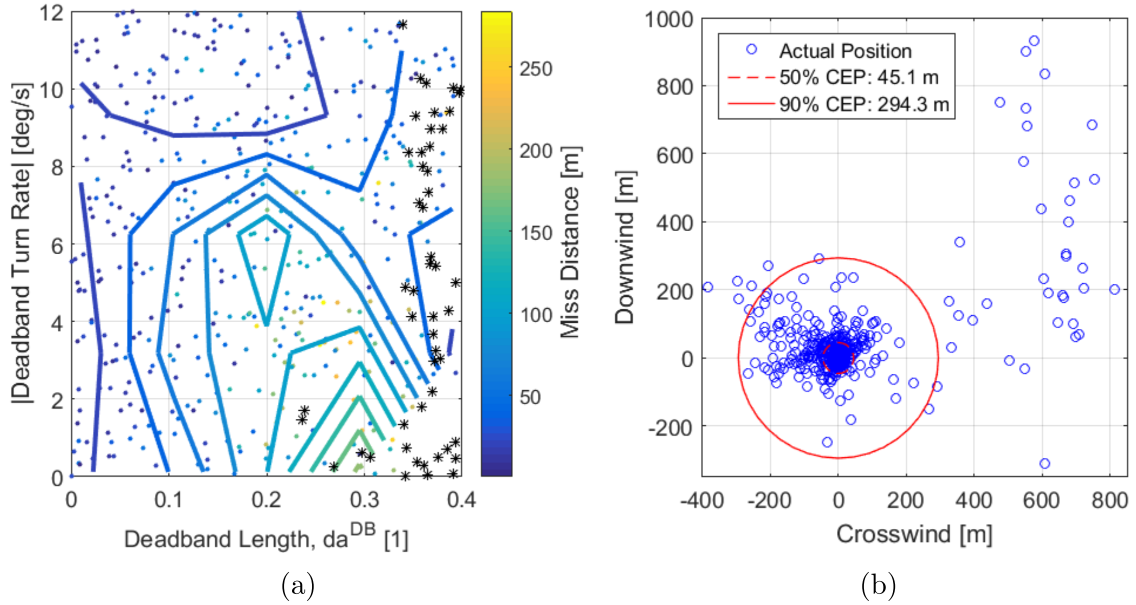
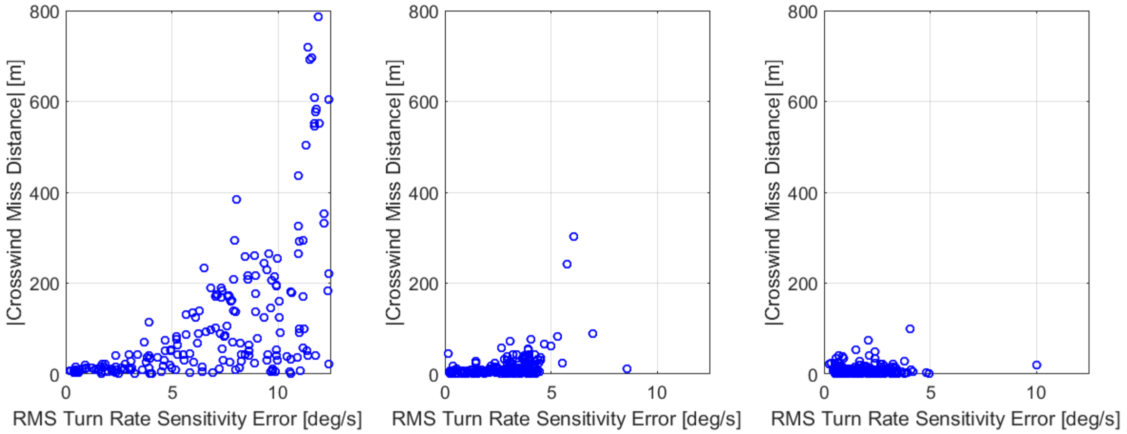
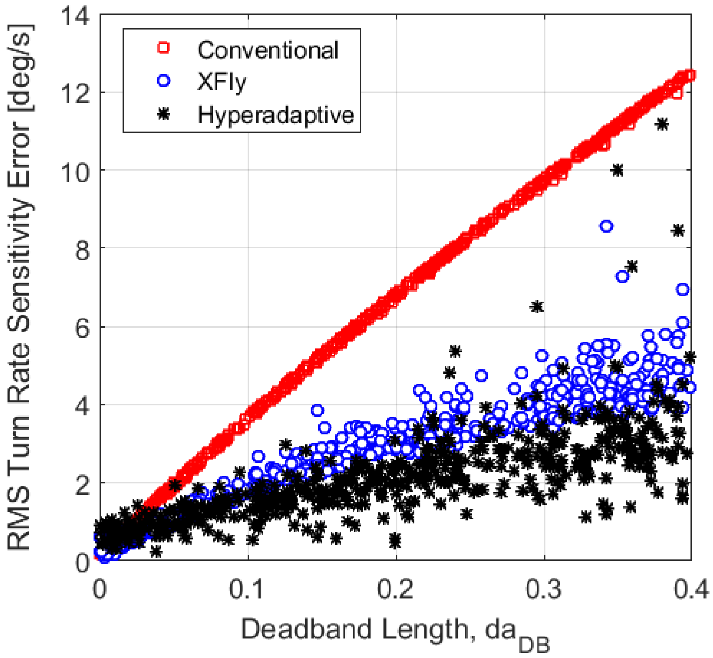


Figure 6.22: Monte Carlos results of the Conventional GNC applied to a precision airdrop system with deadband behavior.



(a) Conventional GNC (b) XFly GNC (c) Hyperadaptive GNC

Figure 6.23: Monte Carlos results of three adaptive algorithms analyzing the effect of deadband length on accuracy of SNLM estimate.



(a) Conventional GNC

Figure 6.24: Monte Carlos results of three adaptive algorithms controlling a vehicle with a deadband nonlinearity.

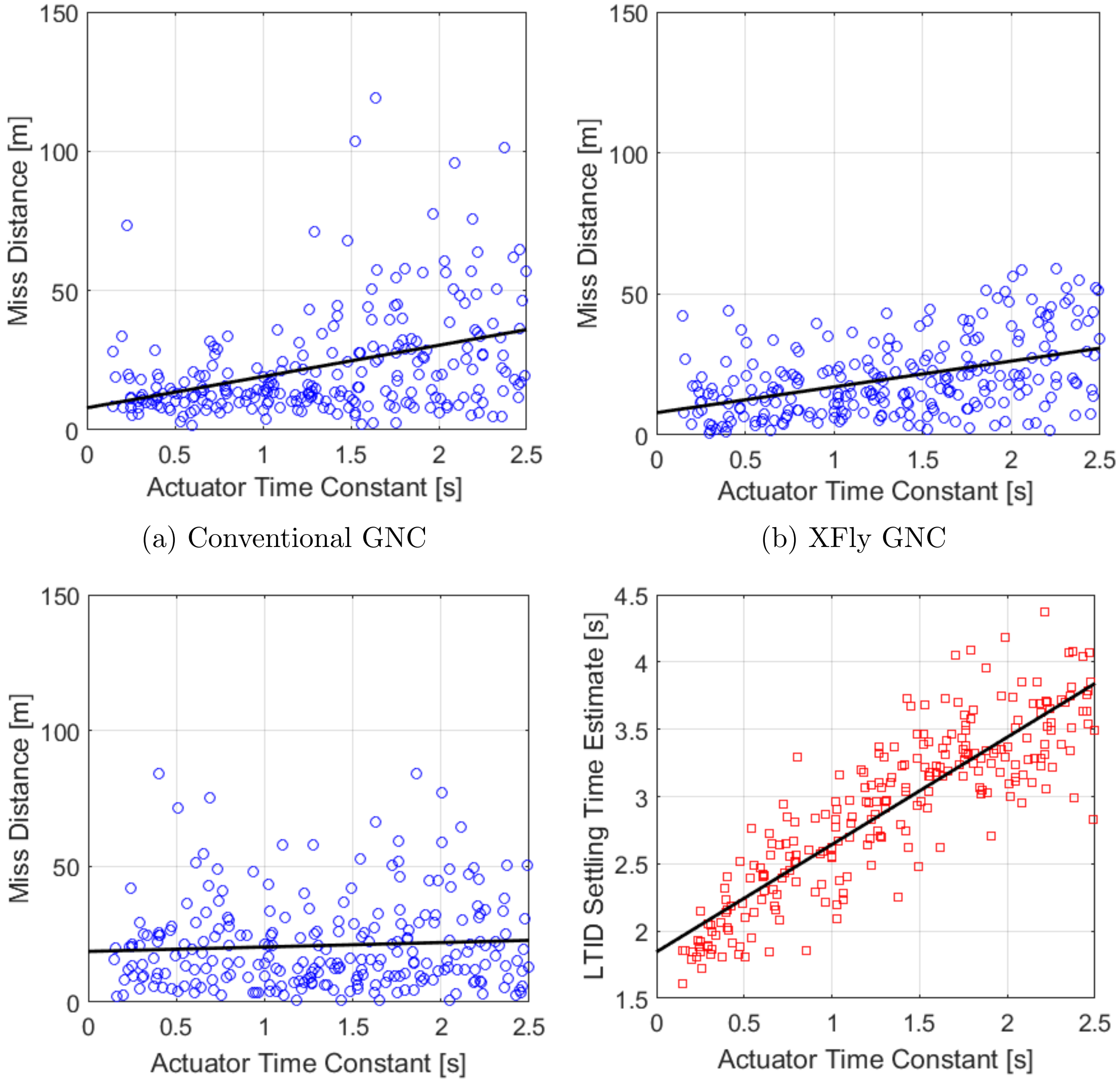
6.2.1.8 Slow Actuators

The final damage scenario slows down the servo actuators to replicate a change in dynamic behavior associated with a different wing loading or linear scaling of the vehicle being controlled [26]. Both the Conventional and XFly GNC algorithms are blind to changes in vehicle dynamic response. Note that some increase in error is expected for all algorithms due to the fact that the vehicle is less agile and can not react as quickly to disturbances such as atmospheric turbulence.

Landing results for each algorithm indicated a small but consistent decrease in accuracy. The 50% CEPs are 15.4m, 17.5m and 15.9m for the Conventional, XFly, and Hyperadaptive GNC algorithms, respectively. While the results are relatively similar for each algorithm, the benefits of estimating the dynamics online are clear in Figure 6.25. Both the Conventional and XFly GNC algorithms exhibit greater miss distance when the actuator time constant is increased. Landing error for slow vehicles lumped downwind of the IP indicating that the vehicle consistently lacked the altitude to reach the target signifying an error in how the altitude margin is calculated.

The hyperadaptive algorithm overcomes this issue by estimating the settling time of the turn rate dynamics and including their effects in the Dubins path planning algorithm (Section 5.2.1.2). When the system response slows, the assumption in the original Dubins algorithm of instantaneous transition between straight and circular flight breaks down. The updated Dubins algorithm uses the results of the estimated LTID settling time, shown in Figure 6.25(d) to estimate the additional flight distance traveled while transitioning between straight and circular flight. This provides feasible paths for the identified vehicle and ensures an accurate calculation of the altitude margin for precise landing.

The compiled statistical results of each algorithm and form of damage is presented in Table 6.3. The percent chance in landing accuracy is presented to quantify the impact of damage on the GNC algorithm in comparison to the nominal case. Results indicated that all methods were able to track changes in vehicle airspeed and constant turn biases. However, as the damage becomes more complex, the landing accuracy of the Conventional and XFly GNC algorithms steadily decrease. The worst results for these two methods is for the control reversal when over an order of magnitude increase in the miss distance was noted due to the unstable lateral tracking controller. The Hyperadaptive algorithm only shows moderate increase across the parameters tested with worse results coming from changing the actuator speed.



(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

(d) Hyperadaptive GNC

Figure 6.25: Monte Carlo analysis of three adaptive GNC algorithms to uncertainties in the speed of the turn rate response.

Table 6.3: Aggregate landing statistics from the single parameter Monte Carlo trade study.

	Conventional GNC				XFly GNC				Hyperadaptive GNC			
	50% CEP [m]		90% CEP [m]		50% CEP [m]		90% CEP [m]		50% CEP [m]		90% CEP [m]	
Damage Type	Value	% Diff	Value	% Diff	Value	% Diff	Value	% Diff	Value	% Diff	Value	% Diff
Nominal	13.1m	-	29.5m	-	10.5m	-	26.1m	-	12.7m	-	29.6m	-
Incidence Angle	13.2m	1%	28.3m	-4%	9.7m	-8%	27.9m	7%	13.2m	4%	41.4m	40%
Turn Rate Bias	14.1m	8%	31.9m	8%	10.6m	1%	26.9m	3%	13.2m	4%	34.2m	16%
Linear Sensitivity Scaling	26.9m	105%	172.1m	483%	14.1m	34%	106.6m	308%	13.3m	5%	31.2m	5%
Asymmetric Sensitivity Scaling	37.5m	186%	164.6m	458%	17.1m	63%	63.5m	143%	12.8m	1%	35.8m	21%
Control Reversal	346.4m	2544%	539.5m	1729%	102.2m	873%	177.1m	579%	14.5m	14%	31.8m	7%
Deadband	45.1m	244%	294.3m	898%	18.1m	72%	53.3m	104%	14.9m	17%	41.3m	40%
Actuator Delay	15.4m	18%	48.5m	64%	17.5m	67%	40.8m	56%	15.9m	25%	44.6m	51%

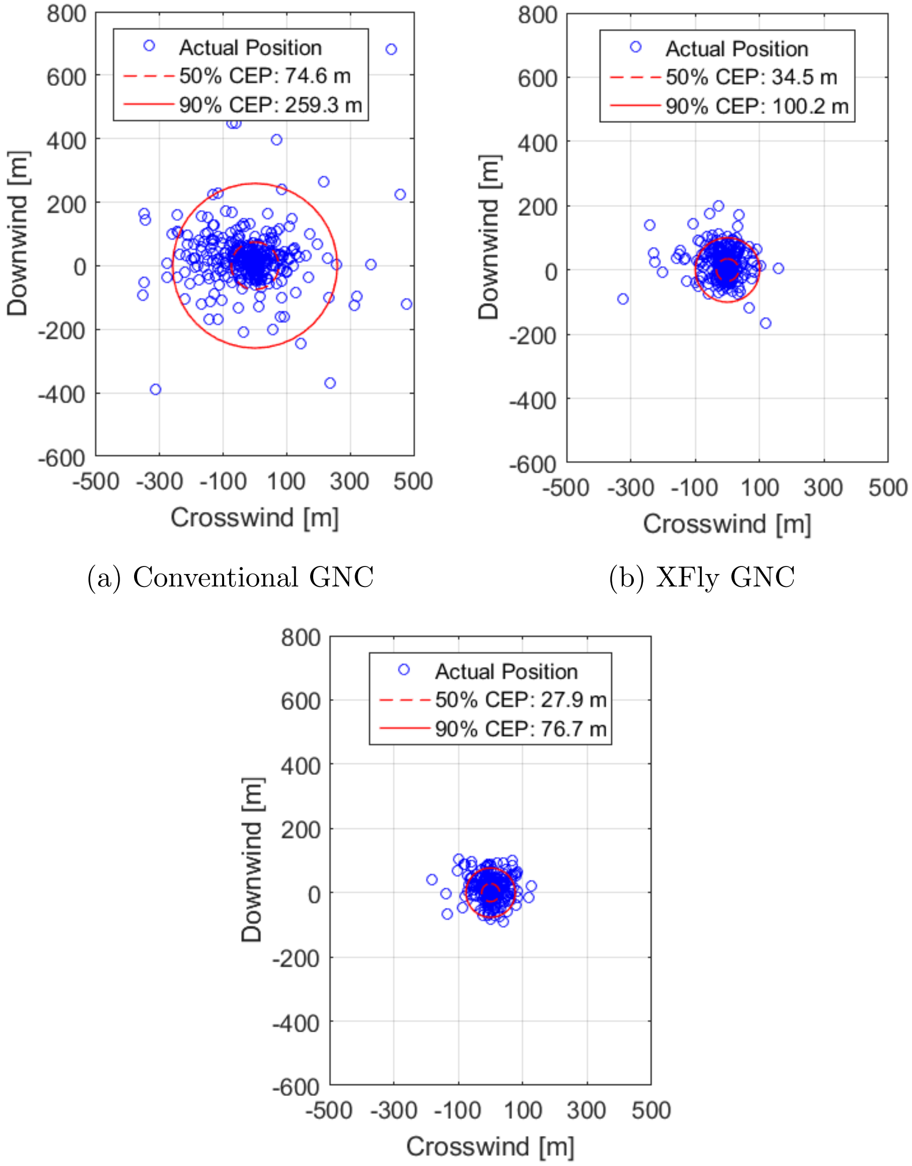
6.2.2 Multi-Parameter Variation

To conclude the simulation based trade studies, the algorithms are stress tested by allowing all damage parameters to be varied simultaneously and including realistic winds. The binary control reversal parameter was left out of the study to provide a more even comparison of the algorithms. When the control reversal is engaged for the Conventional and XFly cases, the landing accuracy becomes less dependent on the (unstable) GNC and more on the wind characteristics and initial location. As a result, the landing accuracy of the Conventional and XFly GNC algorithms are slightly more conservative, particularly the 90% CEP.

Landing dispersions for the Conventional, XFly, and Hyperadaptive algorithms controlling a highly uncertain precision payload system are presented in Figure 6.26. Results indicated the algorithms have 50% CEPs of 74.6m, 34.5m, and 27.9m, respectively which shows a three fold improvement in the Hyperadaptive GNC over the Conventional GNC. Table 6.4 presents the landing statistics for both the nominal and highly uncertain vehicle configurations in the presence of realistic wind fields. The conventional GNC experienced a 288% rise in 50% CEP and over 400% increase in 90% CEP when uncertainty was introduced to the vehicle. The turn rate sensitivity adaption of the XFly GNC reduced the effect of uncertainty, but still exhibited a 100% decrease in landing accuracy for both landing metrics. Finally, the Hyperadaptive GNC was the most robust to the model uncertainty with an increase of 41% and 33% for the 50% and 90% CEPs.

6.3 Flight Testing Results

The final validation for the Hyperadaptive GNC is through experimental flight testing. The small-scale system described in Chapter 3 is used as a testbed to implement studied forms of degraded flight in a real-world environment. In addition to the nominal vehicle configuration, software modifications external to the GNC were used to



(a) Conventional GNC

(b) XFly GNC

(c) Hyperadaptive GNC

Figure 6.26: Multi-parameter Monte Carlo landing dispersions of three adaptive GNC algorithms tested on a vehicle with highly uncertain dynamics.

Table 6.4: Landing statistics for the multi-parameter Monte Carlo simulation

GNC	50% CEP	Change vs. Nominal	90% CEP	Change vs. Nominal
Conventional (Nominal)	19.2m	-	50.9m	-
XFly (Nominal)	17.4m	-	51.9m	-
Hyperadaptive (Nominal)	19.8m	-	57.7m	-
Conventional (Damaged)	74.6m	↑ 288%	163.59m	↑ 410%
XFly (Damaged)	34.5m	↑ 98%	100.2m	↑ 93%
Hyperadaptive (Damaged)	27.9m	↑ 41%	76.7m	↑ 33%

implement a control reversal, asymmetric control sensitivity, and decreased actuator speed cases tested in the parameter study. These tests change both the control sensitivity mapping and the turn rate dynamics to accurately gage the algorithm's ability to handle a large range of model uncertainty. The final test case was implemented on a second parafoil and payload system, shown in Figure 6.27. This second vehicle had a different payload mass, canopy size and material rigidity, and shortened servo actuation arms to reduce the control sensitivity. This system was controlled by the Hyperadaptive algorithm tuned for the regular system and provides a great test for the Hyperadaptive algorithm.

Flight testing was conducted south of Atlanta, GA in a large wooded clearing across all times of the day to test the algorithms in typically calm morning winds and more turbulent afternoon conditions. The results of 51 flights across all case studies are presented in Figure 6.28(a) with individual landing metrics listed in Table 6.5. Aggregate results of the Hyperadaptive GNC indicate a 50% CEP of 35.9m and 90% CEP of 69.0m. This aligns well with the multi-parameter simulation testing which had a 50% and 90% CEP of 27.9m and 76.7m, respectively. As a reference, the Conventional GNC controlling only the nominal system had a 50% and 90% CEP of 27.3m and 48.6m, respectively. Degraded conditions were not applied to the

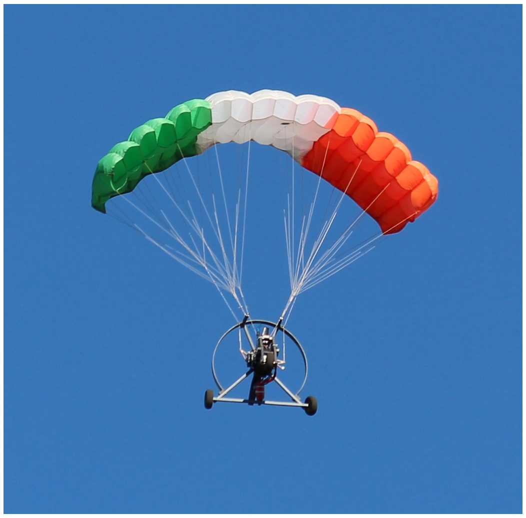


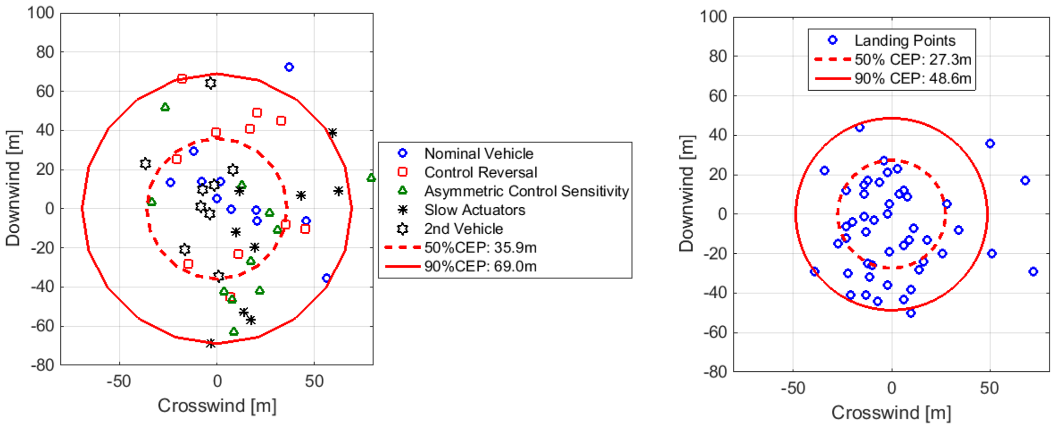
Figure 6.27: Second parafoil and payload system used to flight test the hyperadaptive GNC. Termed the Irish canopy due to the color selection.

Table 6.5: Experimental flight test results for the Hyperadaptive algorithm.

Vehicle Configuration	Number of Flights	50% CEP	90% CEP
Nominal	11	21.2m	66.3m
Control Reversal	11	44.4m	55.8m
Asymmetric Control Sensitivity	9	42.4m	63.4m
Slow Actuators	11	59.5m	90m
2nd Vehicle	9	21.5m	43.0m
Combined	51	35.9m	69m

Conventional controller as recovery of the aircraft could become difficult. As a result, the two metrics cannot be directly compared but the decrease in landing accuracy of only 32% is minimal in comparison to the expected results of the Conventional GNC in similar scenarios.

Further detail on the Hyperadaptive airspeed and LTID estimation results are presented in Figure 6.29. Airspeed estimates show relative consistency for the first three cases which are all tested on the primary vehicle. The case for slow actuators (also applied to the primary vehicle) noted a slight decrease in the average airspeed

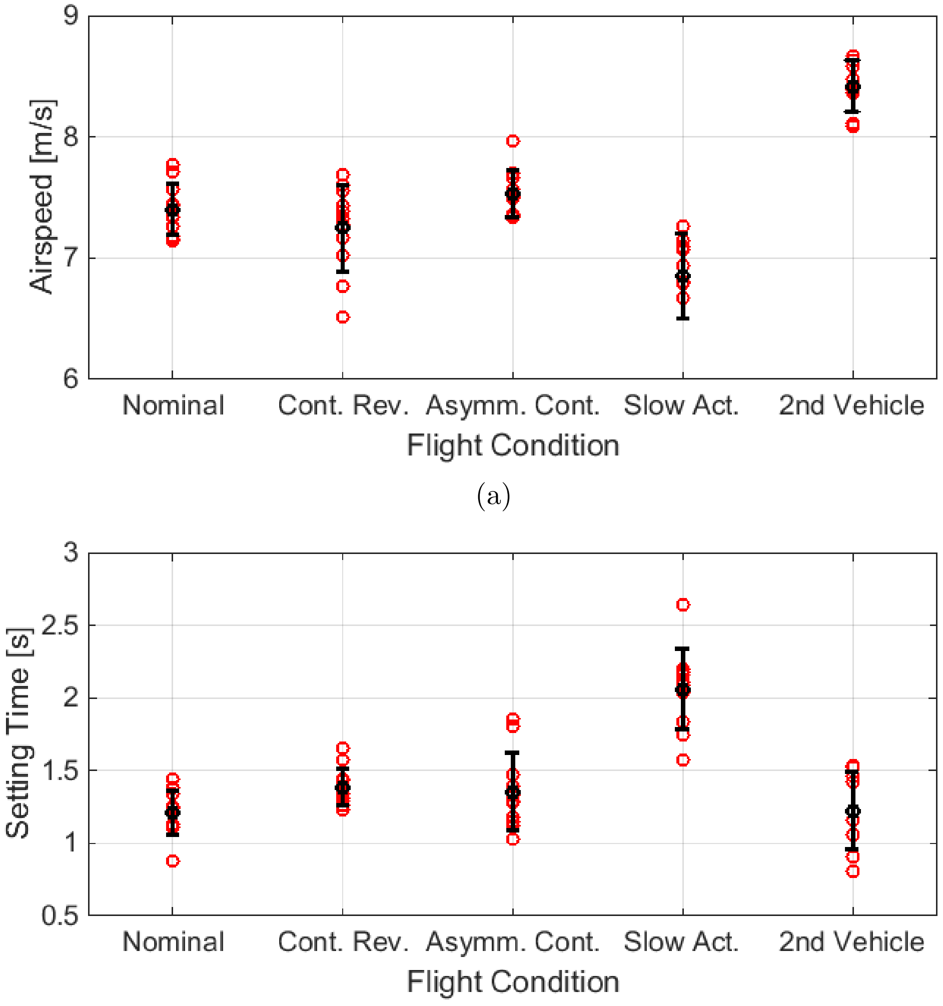


(a) (b)

Figure 6.28: Experimental Landing results of the (a) Conventional GNC and (b) Hyperadaptive GNC algorithms.

estimated. The 2^{nd} vehicle also exhibited a significantly higher airspeed but did not cause significant impact on the landing accuracy. Similarly, the influence of slowing the actuator response was captured by an increased estimate of settling time that was used to improve path planning.

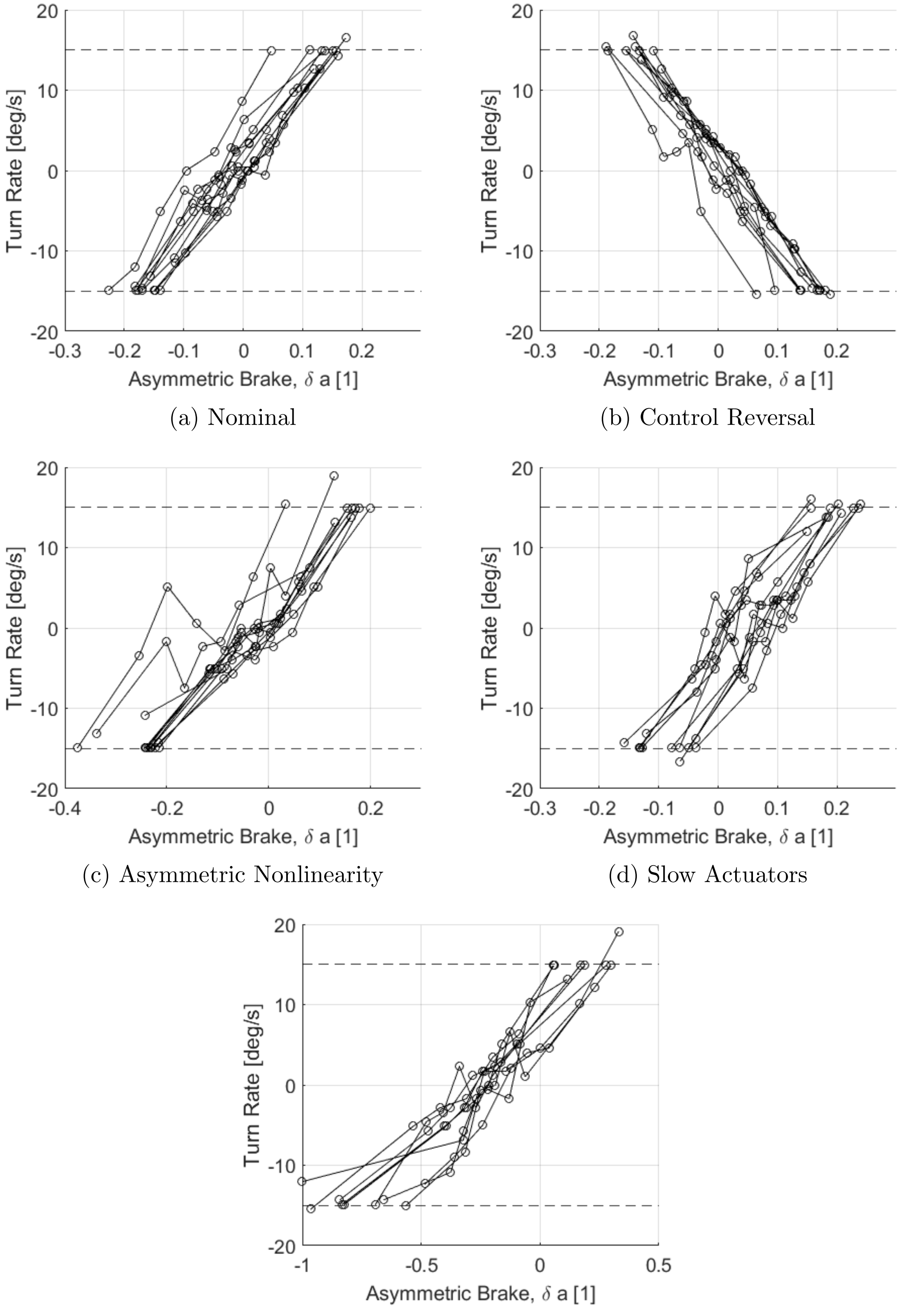
Finally, the set of SNLM estimates for each flight condition is presented in Figure 6.30. The cases highlight the variability noticed in flight testing. First, true atmospheric winds and non-white noise can inject more variability into the estimate. Second, there are more deviations than just the underlying turn rate sensitivity mapping. Experimental flights such as the top estimate of the Nominal case (Figure 6.30(a)) was run after rigging lines of the canopy were damaged during flight test operations and redone. A slight error re-rigging the canopy caused an almost 10 deg/s positive turn rate bias. This was noted from the flight log and the rigging lines were modified to re-trim towards nominal flight. These mappings were able to sufficiently capture the steady state control mappings of the vehicle to provide accurate landings in the presence of high vehicle uncertainty.



(a)

(b)

Figure 6.29: Experimental results of the airspeed and vehicle dynamic model estimators.



(a) Nominal

(b) Control Reversal

(c) Asymmetric Nonlinearity

(d) Slow Actuators

(e) 2nd Vehicle - Irish Canopy

Figure 6.30: Experimental results of the SNLM estimator.

CHAPTER VII

GUIDANCE, NAVIGATION, AND CONTROL OF GPS DENIED AIRDROP SYSTEMS

The accuracy of autonomously guided airdrop systems is highly dependent on the quality and richness of available feedback signals. Since first being made available to the public in 1983, GPS has been the focus of autonomously guided vehicles. After significant development using GPS, current guided airdrop systems cannot function successfully without it.

This work aims to revitalize the application of beacon technology for use in aerial cargo delivery by conducting a study into the viability of both range to target and direction to target as a source of feedback. Additionally, an onboard barometer provides altitude above ground. A schematic of a dropzone configured for beacon guided airdrop systems is presented in Figure 7.1. The beacon is assumed to be the origin of the inertial coordinate system and the desired impact point. Beacon range, R_B, is the spherical distance of the vehicle from the beacon and relative heading to beacon, ψ_B, is the angle between the vehicle's heading and horizontal position vector, $\bar{r}_B$, and z is the altitude above ground level.

The available signals are combined into two sets for GNC development and analysis: method 1 with $\{\psi_B, z\}$ feedback and method 2 using $\{R_B, z\}$ feedback. Unlike GPS, these data sets do not fully define the vehicle position in the Inertial space and limits the ability to estimate the atmospheric winds. Fundamental concepts to the GPS based algorithms such as the wind fixed reference frame are intractable presenting the need for a new approach. This is accomplished though novel beacon-based GNC algorithms which strategically mitigate the effects of limited feedback

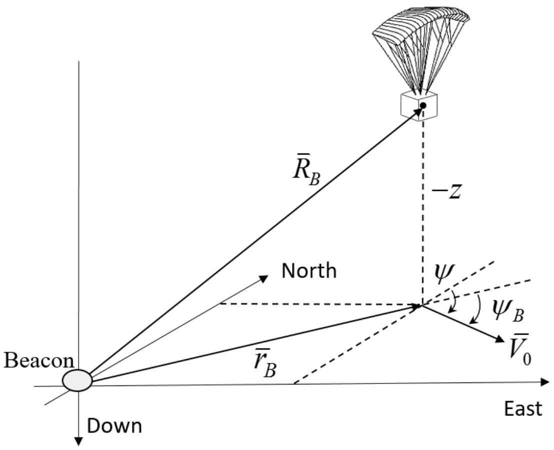


Figure 7.1: Spherical polar representation of the drop zone for beacon feedback control.

and maintain precision landing capabilities in GPS denied scenarios.

7.1 Beacon-based Guidance, Navigation, and Control

Both beacon-based methods use a lateral control algorithm to steer the vehicle using deflection of the trailing edge brakes. Similar to the conventional GNC algorithm presented in Chapter 4, the commanded turn rate is used to calculate the asymmetric brake command, δa_c, based on an *a priori* turn rate mapping $f()$. The limited state feedback requires a more conservative control algorithm over the highly adaptive method presented previously. However, potential avenues to fuse adaptive control strategies with beacon-based guidance and navigation is discussed below.

$$\delta a_c = f(\dot{\psi}_c); \quad \delta a_c \in [-1, 1] \tag{7.1}$$

A saturation limit of 15 deg/s is placed on the commanded turn rate to ensure excessive rocking or spiral behavior is not excited. The command turn rate is calculated by the guidance and navigation algorithms, presented for each method in the sections below.

7.1.1 Method 1: $\{\psi_B, z\}$ Feedback

The use of relative heading to the beacon and altitude above ground reanalyzes the initial work done in the field of guided airdrop systems [7, 8]. Relative heading can be measured in real-time using any a direction-sensing antenna like Watson-Watt antenna which uses orthogonal dipole antennas to conduct radiolocation [51]. This method provides high measurement rates over scanning monopole antennas used previously [7, 8].

The $\{\psi_B, z\}$ method is considered to have the most limited state observability. Both distance to target and atmospheric winds which represent the primary metric to minimize and the greatest disturbance, respectively, are unobservable. Given limited parameters of importance, a basic strategy of homing towards the target is implemented. The control logic always attempt to align the vehicle heading with the direction of the RF beacon. Equations (7.2) & (7.3) calculate the heading error fraction and commanded turn rate, respectively. Here, $\Delta\psi_{max}$ is a tuning parameter defined as the heading error at which the maximum turn rate is commanded, and $\psi_{B,des} = \pi$.

$$\Delta\psi_k = \frac{\psi_{B,des} - \psi_{B,k}}{\Delta\psi_{max}} \tag{7.2}$$

$$\dot{\psi}_c = \begin{cases} \dot{\psi}_{max} & \text{for } \Delta\psi_k \geq 1 \\ -\dot{\psi}_{max} & \text{for } \Delta\psi_k \leq -1 \\ \dot{\psi}_{max}\Delta\psi_k\sqrt{|\Delta\psi_k|} & \text{else} \end{cases} \tag{7.3}$$

This controller provides excellent tracking towards the beacon when the system is far from the IP and loiters in a circular pattern near the IP for the majority of the descent. These loops are created when the vehicle reaches the IP and the heading error becomes maximum causing a 360 degree turn to re-orient towards the target. Without an estimate of the vehicle's position with respect to the target, the algorithm is unable to coordinate reaching the IP when the vehicle altitude tends to zero. As a result, the vehicle remains in the circling pattern until landing.

7.1.2 Method 2: $\{R, z\}$ Feedback

This algorithm uses measurements of spherical range from the IP and altitude above ground to steer the vehicle to the target. The use of beacon range instead of relative heading provides a direct metric to minimize improving the accuracy. This enables the vehicle to time the landing event to be when the system reaches the IP. This is conducted through a two stage homing and loiter guidance strategy in which the vehicle attempts to either home towards or maintain a constant distance from the target. During a typical flight, the algorithm uses homing to reach the dropzone, the loiter phase to maintain proximity to the IP, and will re-enter homing to land at the target.

After deployment, the system enters homing in which the maximum inward radial velocity is tracked to reach the target. The inward velocity of the vehicle depends on the heading of the vehicle and the effects of the atmospheric winds. A bang-bang controller is used to track inward direction because a change in the vehicle's inward velocity can only be used to find the magnitude heading error, but not the sign of the error. This controller switches between maximum left and right turn rate to enforce an oscillation around the inward direction. The control effort is reversed when the radially inward acceleration goes negative indicating that the vehicle is tending away from correct heading. The controller is tuned to track the maximum inward velocity which can vary as a result of changing atmospheric wind conditions.

When the vehicle is aligned towards the target during homing, an estimate of the radial component of the atmospheric winds is calculated by comparing the radial velocity and assumed vehicle airspeed. Though this estimate is subject to errors from a misalignment between vehicle heading and radial direction during the oscillatory approach, it significantly aids in the transition to the loiter period.

During the loiter period the vehicle maintains proximity to the target by flying in a large circular trajectory with radius r_L and centered at the IP. To transition from homing to the loiter path, a 90-degree turn must executed at the transition radius to ensure the vehicle range is close to the loiter radius at the conclusion of the turn. The transition radius is given in Equation (7.4) where the final term is associated with the drift of the vehicle due to atmospheric winds.

$$r_T = r_L + r_{TR}\left(1 - \frac{\pi/2}{V_0}V_{Wr}\right) \tag{7.4}$$

Once in loiter, a proportional derivative (PD) controller is used to track the loiter path and reject errors induced primarily by atmospheric winds. A zero error turn rate associated with the curved path is added to the heading controller (Equation (7.5)).

$$\dot{\psi}_c = \dot{\psi}_L + K_P(r - r_L) + K_D\dot{r} \tag{7.5}$$

$$\dot{\psi}_L = \frac{V_0}{r_L} \tag{7.6}$$

The atmospheric winds directly impact the radial distance and rate of the vehicle which allows their effects to be minimized the PD controller even though they are unobservable. Similar to the previous scheme, this guidance strategy requires the vehicle to penetrate any headwinds to fully circle the desired impact point which limits this method to atmospheric wind speeds at or below the vehicle airspeed. This is generally not a requirement for systems with full GPS feedback which can decouple the atmospheric wind from the total vehicle motion and land accurately at the target if they are deployed sufficiently upwind of the IP.

The system re-enters homing when the current altitude is equal to the altitude required to reach the target, defined by the radial distance, airspeed, and descent rate. This metric is similar to the altitude margin, but does not account for the influence of atmospheric winds making it a worse predictor.

$$z_{transition} = r\frac{\dot{z}}{V_0} \tag{7.7}$$

This transition point is defined independent of the atmospheric winds making the approach to the target and overall landing accuracy heavily dependent on the low altitude atmospheric winds.

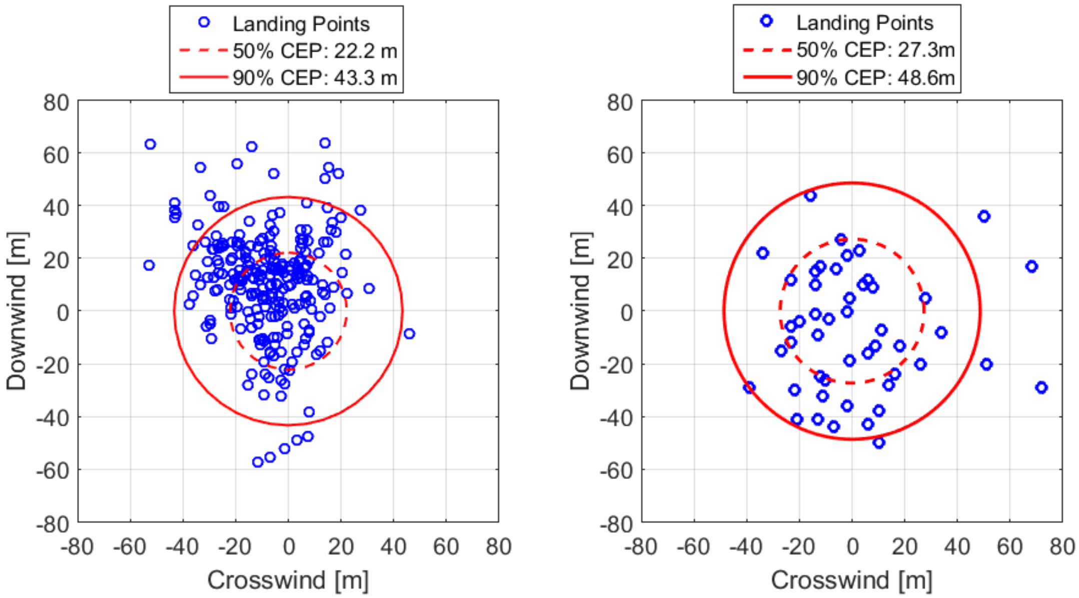
7.2 Simulated and Experimental Results

To study the capability of the proposed beacon based algorithms, a series of simulation and experimental tests were conducted. Simulation results focused on Monte Carlo tests where parameters of the atmospheric wind model were changed between flights such that wind magnitudes varied from 0 to 6 m/s and the angle between the upper and lower air masses varied from 0 to 180 degrees. Experimental flight testing was conducted in a large wooded clearing with gently rolling hills outside of Atlanta, GA.

The RF beacon-based algorithms are benchmarked against the landing capabilities of the Conventional GPS-based GNC algorithm presented in Chapter 4. Monte Carlo simulations and experimental flight test results of the Conventional GPS algorithm are presented in Figure 7.2. This conventional GPS-based algorithm exhibited 50% and 90% CEPs in simulation of 22.2m and 43.3m, respectively. Experimental results closely match with 50% CEP of 27.3m and 90% CEP of 48.6m. Note that the axes of Figure 7.2 are aligned with the wind direction such that a vehicle approaching the target into the wind would be traveling down the vertical axis.

7.2.1 $\{\psi_B$, z$\}$ Feedback

The $\{\psi_B, z\}$ algorithm is simple in nature with only a homing controller used to fly directly towards the target. The system is able to quickly align with and track the inward direction during the initial approach but transitions into a turning loiter phase to maintain proximity to the target. An example simulation of the $\{\psi_B, z\}$ algorithm is presented in Figure 7.3 with the trajectory plot zoomed to near the target for clarity. The tracking error becomes maximum when the vehicle reaches the target near $t = 35$s and the system enters the turning phase. After only two loops, the vehicle approach to the target is aligned with the wind. This occurs because



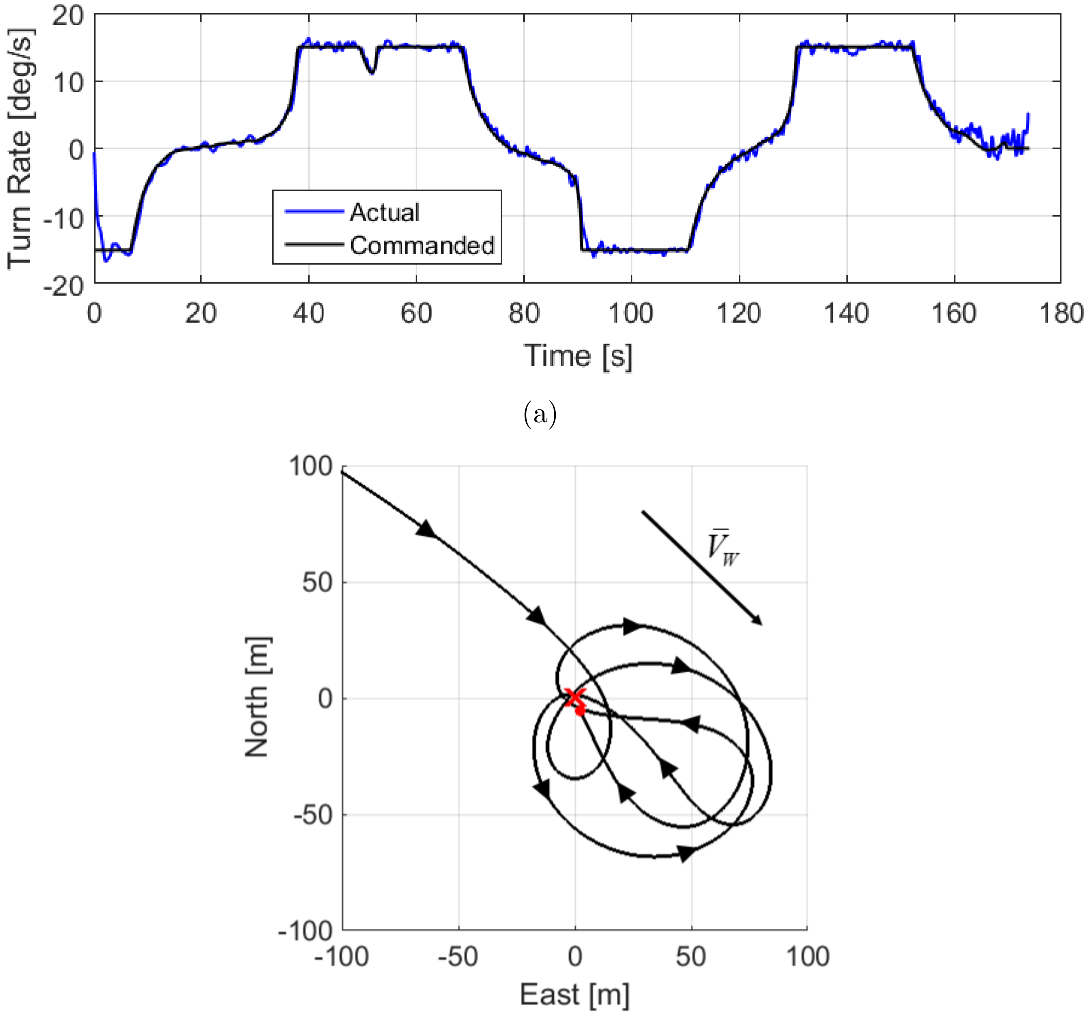
(a)

(b)

Figure 7.2: Landing dispersion of GPS based feedback algorithm in (a) simulation and (b) experimental flight testing.

the vehicle heading, not course direction (direction of the ground track velocity), is aligned to the target. The wind constantly perturbs the vehicle downwind of the target resulting in the inward trajectory aligning anti-parallel with the wind vector. When the vehicle flies over the target, the heading error becomes maximum and the vehicle makes a sharp turn to reorient towards the target.

This algorithm, though simple in nature, exhibits beneficial emergent behavior. The unobservable atmospheric wind magnitude regulates how much a given looping trajectory is stretched downwind of the target. This governs the maximum distance the vehicle gets from the IP, effectively creating an upper bound on the final miss distance. For slow atmospheric winds, the stretching behavior is not pronounced and the vehicle creates a flower pedal shaped trajectory around the IP. As atmospheric wind speed increases, the upper bound increases but also increases the probability the vehicle will land into the wind. Intuitively this occurs because the vehicle spends



(a)

(b)

Figure 7.3: Example simulated trajectory of the $\{\psi_B, z\}$ feedback algorithm

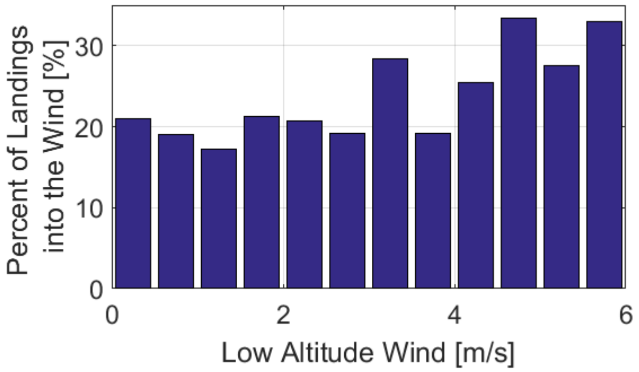
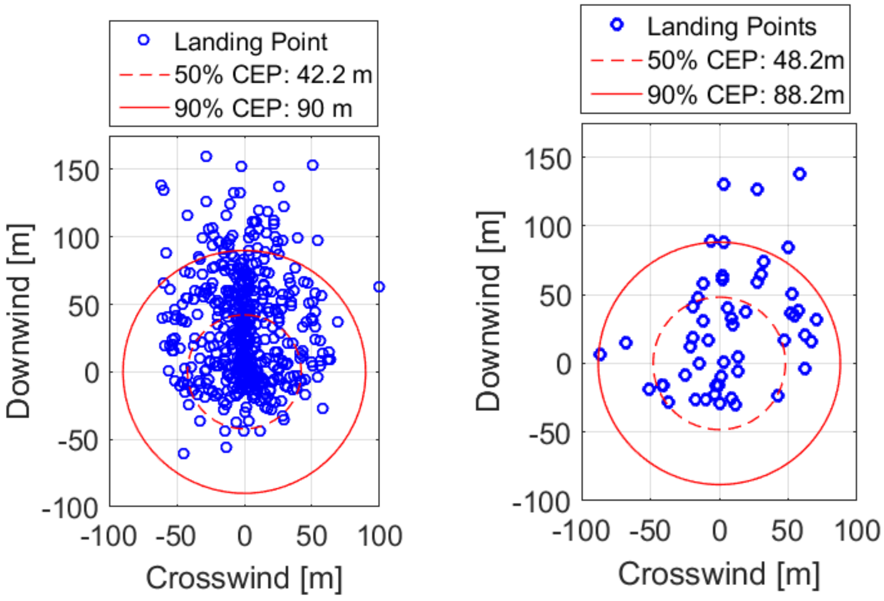


Figure 7.4: Percent of landings aligned with the wind as a function of low altitude wind speeds.

longer flying towards the target than turning to re-orient towards it. The time required to re-orient is constant (based on the 15 deg/s maximum allowed turn rate) whereas the time to fly towards the target increases as oncoming head winds slow the ground speed of the vehicle.

A Monte Carlo simulation analysis was conducted to evaluate the accuracy of this feedback method and provide quantitative justification of the emergent behavior. Results presented in Figure 7.4 indicate that the vehicle lands aligned with the wind under 20% of the time in low winds and approximately 30% of the time with greater wind speeds. While not nearly as effective as the Conventional GPS at aligning with the wind, this emergent behavior helps payload survivability.

Landing accuracy of the algorithm itself is shown in Figure 7.5 with both simulation and experimental results. Simulation results indicate that the $\{\psi_B, z\}$ feedback algorithm has a 42.2m 50% CEP and 90.0m 90% CEP. This represents a decrease in accuracy over the concurrently tested GPS algorithm by 91% and 108%, for each CEP metric respectively. Experimental results of 52 landings show similar results with a 48.2m 50% CEP and 88.2m 90% CEP. As expected, a large number of misses occur along the downwind axis which primarily are associated with flights that land flying toward the target.

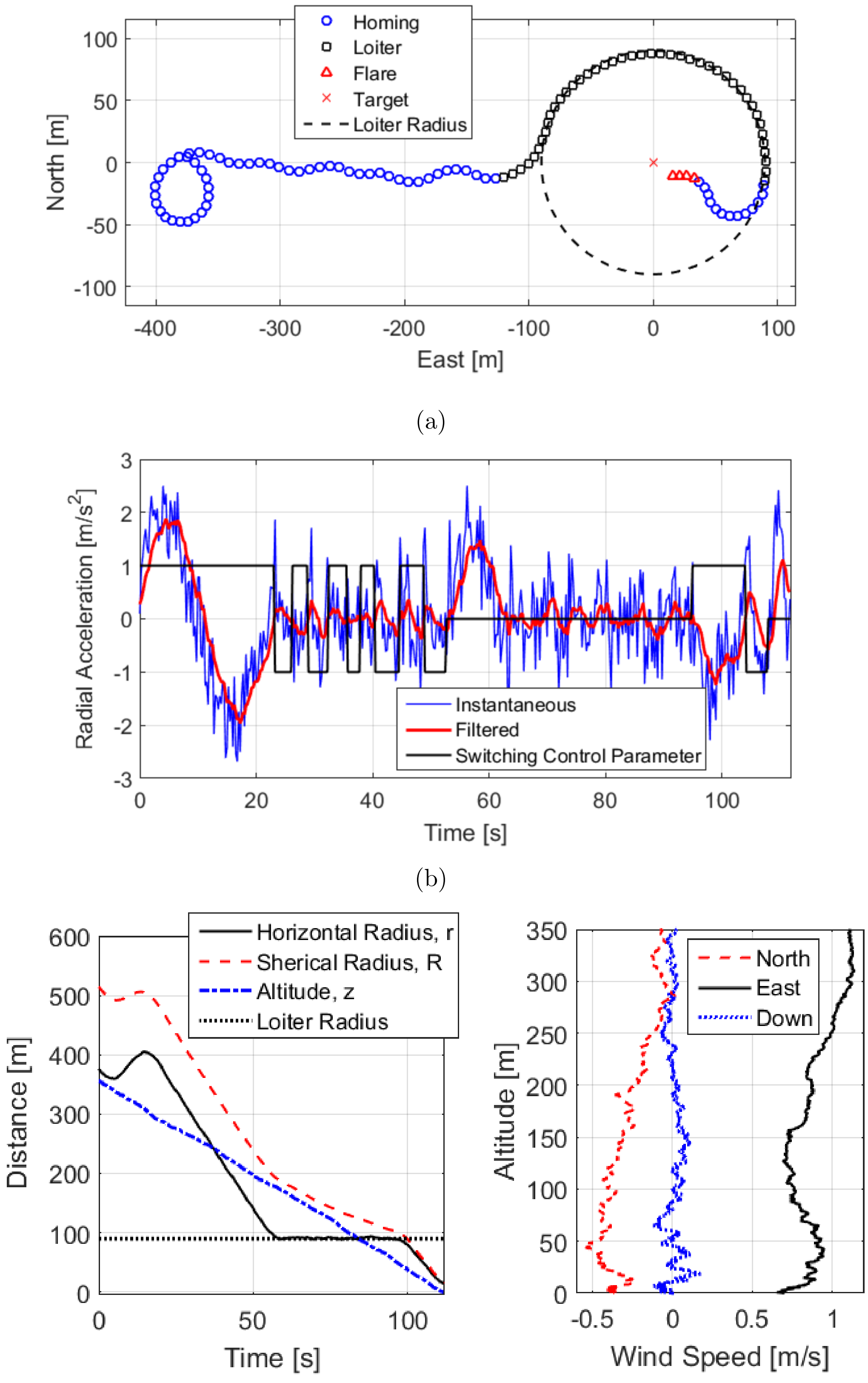


(a) (b)

Figure 7.5: Landing dispersion of the $\{\psi_B, z\}$ feedback algorithm based on (a) simulation results and (b) experimental flight testing.

7.2.2 {R, z} Feedback

Using beacon range instead of beacon heading, this algorithm is able to improve accuracy by actively timing when to land at the target. An example simulated trajectory of an airdrop system employing this GNC algorithm is presented in Figure 7.6. The vehicle is deployed upwind of the target following standard airdrop procedure and immediately enters a turn to find the maximum inward velocity. The vehicle aligns towards the target $t = 22$s when the switching controller being to oscillate as shown in Figure 7.6(b). After, the system accurately homes towards the target and transitions smoothly to the loiter period. During loiter, the PD controller is able to reject the effects of the atmospheric winds by tracking the circular path. The loiter radius was selected at 90 m based on a simulation parameter study which identified a trade-off between minimizing loiter radius and maximizing nominal control effort. A small loiter radius minimizes the distance the vehicle must fly during the final stage



(a)

(b)

(c)

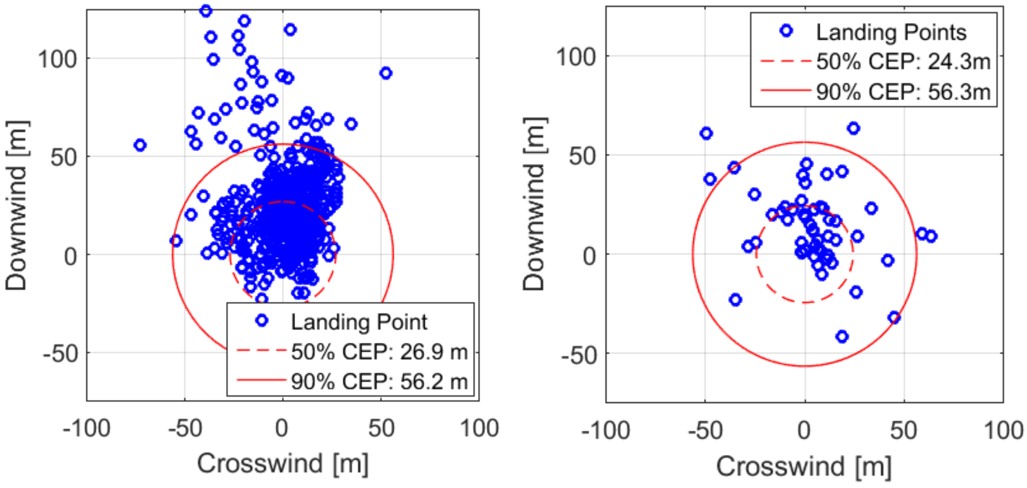
(d)

Figure 7.6: Simulated trajectory of an autonomous airdrop system employing $\{R, z\}$ feedback in a windy environment.

of homing, minimizing the impact of the wind on the final landing point. However, a small turn radius also increases the turn rate necessary to maintain the circular path ($\dot{\psi}_{loiter}$ in Equation (7.6)) and limits the amount of control authority the PD controller has available before actuator saturation. The final homing period is able to fly towards the target but lands short and downwind of the target.

The accuracy of the $\{R, z\}$ beacon feedback method has been extensively tested in simulation through Monte Carlo analysis described above and experimental flight testing. Results of each analysis are presented in Figure 7.7. In simulation, the $\{R, z\}$ beacon system has a 50% CEP of 25.9 m and 90% CEP of 62.3 m. These results indicate only a 21% decrease in accuracy when the beacon system is used instead of GPS and a 36% improvement over the $\{\psi_B, z\}$ controller. 53 drops were conducted during experimental flight testing resulting in a 50% CEP landing accuracy of 24.3m which closely matches and validates the simulation results. As noted previously, the landing accuracy of this algorithm is heavily dependent on the unobservable low altitude winds. The correlation between miss distance and low altitude winds is presented in Figure 7.8. As expected, the system is most accurate when ground winds are mild. At high atmospheric wind magnitudes, the vehicle can be perturbed off of the loiter trajectory. In this case, the vehicle re-enters homing but cannot always return to the target based on the altitude.

The aggregate results of the beacon-based methods are presented in Table 7.1. Overall, the beacon methods are able to maintain high levels of landing accuracy with limited feedback. The effect of the unobservable winds are minimized or used advantageously, though these methods cannot consistently land into the wind.



(a)

(b)

Figure 7.7: Results of the $\{R, z\}$ beacon feedback method tested in (a) simulation and (b) experimental flight testing.

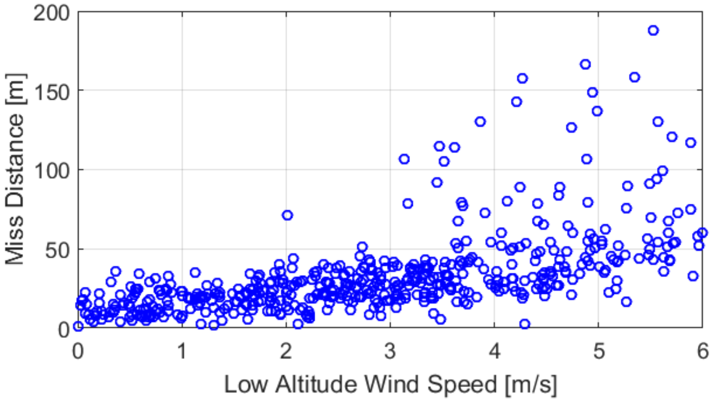


Figure 7.8: Dependence of landing accuracy on low altitude wind speeds for the $\{R, z\}$ beacon feedback method.

Table 7.1: Landing statistics for Beacon-based GNC algorithms.

GNC Type	Simulation		Experimentation	
	50% CEP	90% CEP	50% CEP	90% CEP
Conventional GPS	22.2m	43.3m	27.3m	48.6m
Method 1: $\{\psi_B, z\}$	42.2m	90.0m	48.2m	88.2m
Method 2: $\{R_B, z\}$	26.9m	56.2m	24.3m	56.3m

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Conventional guidance, navigation, and control strategies depend heavily on GPS feedback and an *a priori* nominal vehicle model to steer the precision airdrop system to the target. However, the assumption that GPS feedback is available and that the vehicle performs similar to nominal known behavior is not valid in degraded scenarios which can have significant negative impact on the landing accuracy of the vehicle. This thesis has expanded the set of degraded conditions that a guided airdrop system can successfully negotiate through creation of a novel adaptive control scheme and development of guidance schemes using a radio frequency beacon signal.

A hyperadaptive guidance, navigation, and control algorithm was developed to overcome high model uncertainty typically exhibited by airdrop systems. At its core is a model identification algorithm that can fully characterize the vehicle turn rate dynamics and control sensitivity mapping during the course of a single flight. This method presents a data-driven, non-parametric approach to modeling the control sensitivity to handle a wide array of damage. Additionally, estimation of the dynamic response enabled an improved Dubins path planning algorithm to always provide feasible paths to the control algorithm. Extensive simulation results and flight test data shows that the hyperadaptive GNC is robust to all forms of model uncertainty introduced. Minimal to no loss of landing accuracy was noted for changes in atmospheric winds, vehicle airspeed, turn rate biases, actuator speed, and many linear and nonlinear changes to the control sensitivity mapping including non-monotonic and deadband behavior.

As a comparison, two GNC algorithms based on currently fielded flight software were studied. Both algorithms performed equivalently for small levels of damage which result in changes to the vehicle airspeed or turn rate bias. However, as more heavily damaged situations are considered, the comparison algorithms show decreasing accuracy. The worst performance resulted from considering a control reversal where over an order of magnitude decrease in landing accuracy was noted. This commonly induced human errors destabilizes the comparator algorithms causing undesired spiral behavior but the data-driven approach of the hyperadaptive algorithm is not affected. When compared directly using a highly uncertain airdrop vehicle, the hyperadaptive algorithm most successfully rejected damage, exhibiting a 40% decrease in landing accuracy in comparison to the nominal vehicle. The comparison algorithms were more adversely affected with 100% and 300% increases in the median miss distance. Due to instability of control reversals, the impact on 90% CEP is even greater. After extensive flight simulation of the hyperadaptive algorithm, over 100 experimental flight tests were conducted on a small-scale vehicle to test the hyperadaptive algorithm. The computation requirements of this algorithm were actively controlled to ensure the ability to run in real-time on a 80MHz microprocessor. Flight test data indicated only a 30% decrease in landing accuracy when controlling highly uncertain vehicle in comparison a conventional GNC tested of a vehicle with *no* damage. Additionally, when applied to the nominal vehicle, the hyperadaptive GNC showed very comparable landing accuracy to that of a GNC tuned specifically for that vehicle implying no base accuracy loss by switching to a fully adaptive system.

A beacon-based strategy was employed to overcome situations where GPS is unavailable or severely degraded. Two novel GNC algorithms were developed to strategically use limited information to home towards and land near the target. These methods were tested through extensive simulation analysis and over 50 experimental flight tests for each algorithm. Very close alignment was noted between the simulation and experimental results, validating the testing procedure and exposure to a large variety of wind fields. With radial range, the beacon-based algorithm can sense the radial component of the wind while homing, loiter in proximity to the target and land with only a slight decrease in landing accuracy as compared to a GPS-based algorithm. The main disadvantage is the inability to coordinate a landing into the wind which could hamper payload survivability. The second beacon-based strategy analyzed heading to the beacon as a source of feedback. Far from the target this provides excellent homing, and once at the target, flies in a looping pattern until landing. This simple procedure exhibits beneficial emergent behavior that aligns the approach to the target into the wind vector. This method experienced a 90% decrease in landing accuracy in comparison to a GPS-based algorithm but still provides a high level of accuracy in situations where GPS is not available.

8.2 Future Work

The work presented here has several limitations and areas for improvement. A list of key developments for this project are:

1. The current hyperadaptive scheme requires the ability to turn both left and right at the desired maximum turn rate. Expanded logic to handle scenarios where the vehicle can only turn in one direction should be investigated. The algorithm can be expanded to include Dubins paths with unidirectional constraints as explored by Choi [94] and the SNLM estimator can identify when this guidance constraint needs to be employed.

2. Improvements to the control strategy can also be studied. The estimation of the linear dynamic model opens the door to more advanced control algorithms. Model predictive control is the most commonly applied method in the literature for application with Hammerstein models [90, 91, 95, 96] and has also been

studied for the precision airdrop system [97,98].

3. Research that advances the sensing and feedback capabilities for airdrop systems can enable both the hyperadaptive and beacon algorithms to be expanded.

   (a) With improved feedback, many of the steady state longitudinal parameters could be estimated and controlled in real time. This would take advantage of the landing accuracy improvements gained with dual channel control.

   (b) Preliminary work on a full state observer for beacon-based feedback shows promise, but depends on high quality inertial heading measurements for good tracking behavior. This strategy has been analyzed for autonomous underwater vehicles [48,49,57,100] and is viable for precision airdrop systems, but does exhibit observability issues when the vehicle is aligned to the target [57]. If developed, this method could provide a truly redundant feedback method for GPS blackout as it enables the same guidance and control algorithms to steer the vehicle to the target.

4. Finally, there is potential to use the hyperadaptive strategy with a variety of conventionally non-adaptive GNCs because the core Hammerstein model identification algorithm and integration process is modular. This includes strategies employing optimal path planning and even beacon-based navigation. As long as an airdrop vehicle is equipped with heading rate feedback, the lateral turn rate mapping procedure can be employed. This aligns well with the beacon heading method$\{\psi_B, z\}$ because it only depends on an accurate mapping of the control sensitivity to land near the target.

APPENDIX A

XFLY ADAPTIVE FLIGHT SOFTWARE

The XFly GNC incorporates an Model Reference Adaptive Controller (MRAC) into the Conventional GNC flight software presented in Chapter 4. The PI heading controller can apply a constant offset to the *a priori* turn rate sensitivity mapping $f(\dot{\psi}_c)$ but cannot account for changes in the form or linear gain of the function. The MRAC approximates the turn rate mapping as a linear function and updates the parameters to drive the tracking error to zero.

$$\delta a_c = f(\dot{\psi}_c) + \delta a_{bias} \tag{A.1}$$

$$\delta a_c \approx \delta a_{Gain} \dot{\psi}_c + \delta a_{bias} \tag{A.2}$$

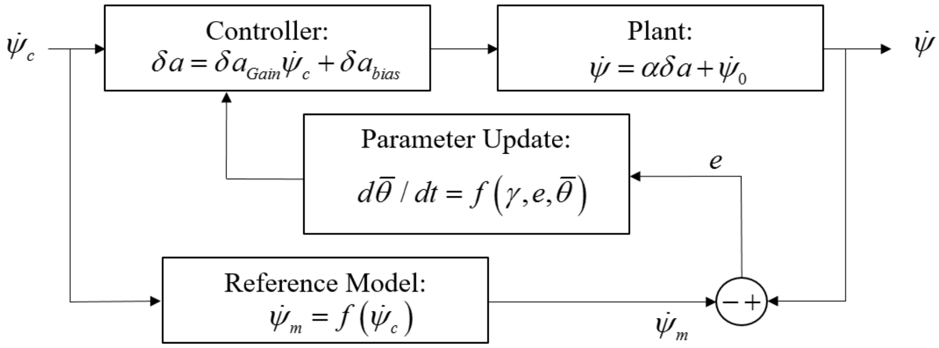
An illustration of the controller is presented in Figure A.1 where both control parameters are updated by studying the error between the reference model turn rate and measured turn rate. This controller assumes the plant takes a linear form given as:

$$\dot{\psi} = \alpha \delta a + \dot{\psi}_0 \tag{A.3}$$

To drive the error to zero, the controller must become an inverted estimate of the plant. Inspection of the error given in Equation (A.5) reveals the optimal values of the control parameters. For convenience, the two control parameters are combined into a parameter vector $\bar{\theta} = [\delta a_{Gain}, \delta a_{bias}]^T$.

$$\begin{aligned} e(\bar{\theta}) &= \dot{\psi} - \dot{\psi}_m && \text{(A.4)} \\ &= \alpha(\theta_1 \dot{\psi}_c + \theta_2) + \dot{\psi}_0 - \dot{\psi}_m && \text{(A.5)} \end{aligned}$$

$$\theta_1^* = 1/\alpha; \quad \theta_2^* = -\dot{\psi}_0/\alpha; \tag{A.6}$$



$\dot{\psi}_c$
Controller:
$\delta a = \delta a_{Gain}\dot{\psi}_c + \delta a_{bias}$
Plant:
$\dot{\psi} = \alpha\delta a + \dot{\psi}_0$
$\dot{\psi}$
Parameter Update:
$d\bar{\theta}/dt = f(\gamma, e, \bar{\theta})$
e
Reference Model:
$\dot{\psi}_m = f(\dot{\psi}_c)$
$\dot{\psi}_m$
$-$
$+$

Figure A.1: Block diagram of Model Reference Adaptive Controller.

Convergence to the optimal parameters is accomplished using the quadratic error criterion.

$$J(\bar{\theta}) = \frac{1}{2}e(\bar{\theta})^2 \tag{A.7}$$

The goal of the parameter update is to decrease the error by updating the parameters in the direction of the negative gradient of $J(\bar{\theta})$. This definition ensures Lyapunov stability by having a positive definite cost function and negative cost derivative [101]. The parameter derivative can be calculated using the error sensitivity, $\partial e/\partial\bar{\theta}$.

$$\frac{d\bar{\theta}}{dt} \equiv -\gamma\frac{\partial J(\bar{\theta})}{\partial\theta} = -\gamma e\frac{\partial e(\bar{\theta})}{\partial\bar{\theta}} \tag{A.8}$$

Solving Equation (A.8) yields the parameter update laws which are converted to the discrete domain for digital implementation.

$$\frac{d\theta_1}{dt} = -\gamma_1\alpha\dot{\psi}_c\left(\dot{\psi} - \dot{\psi}_m\right) \tag{A.9}$$

$$\frac{d\theta_2}{dt} = -\gamma_2\alpha\left(\dot{\psi} - \dot{\psi}_m\right) \tag{A.10}$$

Given that α is an unknown parameter, it can be combined with the adaption gain, γ.

REFERENCES

[1] "In the news," `https://info.aiaa.org/tac/AASG/ADSTC/Web%20Pages/In%20the%20news.aspx`, 2010, Accessed: 2016-10-15.

[2] Tschida, T., "X-38 Prototype Lands on Rogers Dry Lakebed," 07/22/2015 2000.

[3] Culpepper, S., Ward, M., and Costello, M., "Adaptive Control of Damaged Parafoils," 2013.

[4] Ward, M., Costello, M., and Slegers, N., "Specialized System Identification for Parafoil and Payload Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 588–597.

[5] Scheuermann, E., *Autonomous Control of Parafoil and Payload Systems Using Upper Surface Canopy Spoilers*, Ph.D. thesis, 2015.

[6] Brown, G. and Benney, R., "Precision Aerial Delivery Systems in a Tactical Environment," *Proceedings of the 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2005, pp. 23–26.

[7] Kane, M. T., Dicken, H., and Buehler, R., "A Homing Parachute System," Tech. rep., Sandia Corp., Albuquerque, N. Mex., 1961.

[8] Barton, W. and Knapp, C., "Controlled recovery of payloads at large glide distances, using the para-foil," *Journal of Aircraft*, Vol. 5, No. 2, 1968, pp. 112–118.

[9] Goodrick, T., "Estimation of Wind Effect on Gliding Parachute Cargo Systems Using Computer Simulation," *AIAA Aerodynamic Decelerator Systems Conference*, 1970, pp. 14–16.

[10] Goodrick, T. F. and Murphy, A., "Analysis of various automatic homing techniques for gliding airdrop systems with comparative performance in adverse winds," *American Institute of Aeronautics and Astronautics, Aerodynamic Deceleration Systems Conference, 4 th, Palm Springs, Calif*, p. 1973.

[11] Goodrick, T., "Hardware Options for Gliding Airdrop Guidance Systems," *6th AIAA Aerodynamic Decelerator and Balloon Technology Conference*, 1979.

[12] Benney, R., Barber, J., McGrath, J., McHugh, J., Noetscher, G., and Tavan, S., "The joint precision airdrop system advanced concept technology demonstration," *AIAA paper*, Vol. 1601, 2005, pp. 23–26.

[13] Wright, R., Benney, R., and McHugh, J., "Precision airdrop system," *AIAA Paper*, Vol. 1644, 2005, pp. 23–26.

[14] Smith, J., Bennett, B., and Fox, R., "Development of the NASA X-38 parafoil landing system," *Proceedings of the 3rd AIAA Workshop on Weakly Ionized Gases Workshop*, 1999, pp. 1–5.

[15] Stein, J. M., Madsen, C. M., and Strahan, A. L., "An overview of the guided parafoil system derived from X-38 experience," *Proceedings of the 18th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 1999, pp. 23–26.

[16] Bennett, T. and Fox, R., "Design, Development and Flight Testing of the NASA X-38 7500 ft2 Parafoil Recovery System," *AIAA Paper*, Vol. 2107, 2003.

[17] Hogue, J. R. and Jex, H. R., "Applying parachute canopy control and guidance methodology to Advanced Precision Airborne Delivery Systems," *AIAA Aerodynamic Decelerator Systems Technology Conference*, 1995, pp. 15–18.

[18] Murray, J. E., Sim, A. G., Neufeld, D. C., Rennich, P., Norris, S., and Hughes, W., *Further development and flight test of an autonomous precision landing system using a parafoil*, Vol. 4599, National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program, 1994.

[19] Calise, A. and Preston, D., "Design of a stability augmentation system for airdrop of autonomous guided parafoils," *AIAA Paper*, Vol. 6776, 2006.

[20] Calise, A. J. and Preston, D., "Swarming/flocking and collision avoidance for mass airdrop of autonomous guided parafoils," *Journal of guidance, control, and dynamics*, Vol. 31, No. 4, 2008, pp. 1123–1132.

[21] Slegers, N. and Costello, M., "Aspects of control for a parafoil and payload system," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 898–905.

[22] Ward, M. and Costello, M., "Adaptive glide slope control for parafoil and payload aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1019–1034.

[23] Ward, M., Cacan, M., Scheuermann, E., Costello, M., Bergeron, K., Noetscher, G., and Shurtliff, M., "Flight Test Results of Recent Advances in Precision Airdrop Guidance, Navigation, and Control Logic," 2015.

[24] Brown, G., Haggard, R., Almassy, R., Benney, R., and Dellicker, S., "The Affordable Guided Airdrop System (AGAS)," *15th CEAS/AIAA Aerodynamic Decelerator Systems Technology Conference*, 1999, pp. 8–11.

[25] Jorgensen, D. S. and Hickey, M. P., "The AGAS 2000 precision airdrop system," *Proceedings of Infotech@ Aerospace*, 2005, pp. 26–29.

[26] Brown, G. J., “Parafoil steady turn response to control input,” *AIAA Paper*, Vol. 1241, 1993, pp. 10–13.

[27] Ward, M., Culpepper, S., and Costello, M., “Parafoil Control Using Payload Weight Shift,” *Journal of Aircraft*, Vol. 51, No. 1, 2014, pp. 204–215.

[28] Scheuermann, E., Ward, M., Cacan, M., Costello, M., Bergeron, K., and by Institutions, F., “Flight Testing of Autonomous Parafoils Using Upper Surface Bleed Air Spoilers,” 2015.

[29] Slegers, N., Beyer, E., and Costello, M., “Use of variable incidence angle for glide slope control of autonomous parafoils,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 585–596.

[30] Bergeron, K., Fejzic, A., and Tavan, S., “AccuGlide 100: Precision airdrop guidance and control via glide slope control,” *21st AIAA Aerodynamic Decelerator Systems Technology Conference*, 2011.

[31] Bergeron, K., Noetscher, G., Shurtliff, M., Deazley, F., and by Institutions, F., “Longitudinal Control for Ultra Light Weight Guided Parachute Systems,” 2015.

[32] Calise, A. J., Hovakimyan, N., and Idan, M., “Adaptive output feedback control of nonlinear systems using neural networks,” *Automatica*, Vol. 37, No. 8, 2001, pp. 1201–1211.

[33] Calise, A. J., Yang, B.-J., and Craig, J. I., “Augmenting adaptive approach to control of flexible systems,” *Journal of guidance, control, and dynamics*, Vol. 27, No. 3, 2004, pp. 387–396.

[34] Jann, T., *Advanced Features for Autonomous Parafoil Guidance, Navigation and Control*, Aerodynamic Decelerator Systems Technology Conferences, American Institute of Aeronautics and Astronautics, 2005, doi:10.2514/6.2005-1642.

[35] Carter, D., George, S., Hattis, P., McConley, M., Rasmussen, S., Singh, L., and Tavan, S., "Autonomous large parafoil guidance, navigation, and control system design status," *AIAA Paper*, Vol. 2514, 2007.

[36] Ward, M., Costello, M., and Slegers, N., "On the Benefits of In-Flight System Identification for Autonomous Airdrop Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1313–1326.

[37] Fontana, R. D., Cheung, W., Novak, P. M., and Stansell, T., "The new L2 civil signal," *Proceedings of ION GPS*, 2001.

[38] Leonard, J. J. and Durrant-Whyte, H. F., "Mobile robot localization by tracking geometric beacons," *Robotics and Automation, IEEE Transactions on*, Vol. 7, No. 3, 1991, pp. 376–382.

[39] Kohlbrecher, S., Von Stryk, O., Meyer, J., and Klingauf, U., "A flexible and scalable slam system with full 3d motion estimation," *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, IEEE, 2011, pp. 155–160.

[40] Ding, W., Wang, J., Han, S., Almagbile, A., Garratt, M. A., Lambert, A., and Wang, J. J., "Adding Optical Flow into the GPS/INS Integration for UAV navigation," *Proc. of International Global Navigation Satellite Systems Society Symposium*, Citeseer, 2009, pp. 1–13.

[41] Giachetti, A., Campani, M., and Torre, V., "The use of optical flow for road navigation," *Robotics and Automation, IEEE Transactions on*, Vol. 14, No. 1, 1998, pp. 34–48.

[42] Kendoul, F., Fantoni, I., and Nonami, K., "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, Vol. 57, No. 6, 2009, pp. 591–602.

[43] Barrows, G. L., "Optic flow sensors for MAV navigation," *Fixed, flapping and rotary vehicles at very low Reynolds Numbers, Proceedings of the Conference, Notre Dame, IN*, 2000, pp. 464–476.

[44] Yun, S., Lee, Y. J., and Sung, S., "IMU/Vision/Lidar integrated navigation system in GNSS denied environments," *Aerospace Conference, 2013 IEEE*, IEEE, 2013, pp. 1–10.

[45] Kelly, J. and Sukhatme, G. S., "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, Vol. 30, No. 1, 2011, pp. 56–79.

[46] Conte, G. and Doherty, P., "An integrated UAV navigation system based on aerial image matching," *Aerospace Conference, 2008 IEEE*, IEEE, 2008, pp. 1–10.

[47] Chitre, M., "Path planning for cooperative underwater range-only navigation using a single beacon," *Autonomous and Intelligent Systems (AIS), 2010 International Conference on*, IEEE, 2010, pp. 1–6.

[48] Vaganay, J., Baccou, P., and Jouvencel, B., "Homing by acoustic ranging to a single beacon," *OCEANS 2000 MTS/IEEE Conference and Exhibition*, Vol. 2, IEEE, 2000, pp. 1457–1462.

[49] Casey, T., Guimond, B., and Hu, J., "Underwater vehicle positioning based on time of arrival measurements from a single beacon," *OCEANS*, IEEE, 2007, pp. 1–8.

[50] Priyantha, N. B., Miu, A. K., Balakrishnan, H., and Teller, S., "The cricket compass for context-aware mobile applications," *Proceedings of the 7th annual international conference on Mobile computing and networking*, ACM, 2001, pp. 1–14.

[51] Watson-Watt, R., "The evolution of radiolocation," *Electrical Engineers-Part IIIA: Radiolocation, Journal of the Institution of*, Vol. 93, No. 1, 1946, pp. 11–19.

[52] Kurth, D., Kantor, G., and Singh, S., "Experimental results in range-only localization with radio," *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Vol. 1, IEEE, 2003, pp. 974–979.

[53] Kantor, G. and Singh, S., "Preliminary results in range-only localization and mapping," *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, Vol. 2, IEEE, 2002, pp. 1818–1823.

[54] Pierce, J. A., McKenzie, A., and Woodward, R. H., *LORAN, long range navigation*, Vol. 4, McGraw-Hill Book Co., 1948.

[55] Farnsworth, B. D. and Taylor, D. W., "High-precision 2.4 GHz DSSS RF ranging," *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation.*

[56] Ensco Inc., `www.ensco.com/national-security/agile-radio-frequency-rf-ranging`, Accessed: 2016-10-15.

[57] Gadre, A. S. and Stilwell, D. J., "A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location," *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, 2005, pp. 1420–1425.

[58] Doraiswami, R., “A novel Kalman filter-based navigation using beacons,” *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 32, No. 2, 1996, pp. 830–840.

[59] Goodrick, T., “Theoretical study of the longitudinal stability of high-performance gliding airdrop systems,” *American Institute of Aeronautics and Astronautics, Aerodynamic Deceleration Systems Conference, 5 th, Albuquerque, N. Mex*, 1975.

[60] Crimi, P., “Lateral stability of gliding parachutes,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 6, 1990, pp. 1060–1063.

[61] Iosilevskii, G., “Center of gravity and minimal lift coefficient limits of a gliding parachute,” *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1297–1302.

[62] Mooij, E., Wijnands, Q., and Schat, B., “9 DoF Parafoil/Payload Simulator Development and Validation,” 2003.

[63] Prakash, O. and Ananthkrishnan, N., “Modeling and simulation of 9-DOF parafoil-payload system flight dynamics,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2006, pp. 21–24.

[64] Gorman, C. and Slegers, N., “Comparison and Analysis of Multi-Body Parafoil Models with Varying Degrees of Freedom,” *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2011, pp. 23–26.

[65] Barrows, T. M., “Apparent mass of parafoils with spanwise camber,” *Journal of Aircraft*, Vol. 39, No. 3, 2002, pp. 445–451.

[66] Lissaman, P. and Brown, G. J., “Apparent mass effects on parafoil dynamics,” *AIAA Paper*, Vol. 1236, 1993, pp. 10–13.

[67] "Military Standard, Flying Qualities of Piloted Aircraft," Vol. MIL-STD-1797A, 2004.

[68] Gage, S., "Creating a unified graphical wind turbulence model from multiple specifications," *2003 AIAA Modeling and Simulation Technologies Conference and Exhibit*.

[69] Kaminer, I. I. and Yakimenko, O. A., *Development of control algorithm for the autonomous gliding delivery system*, Defense Technical Information Center, 2003.

[70] Ewing, E., Bixby, H., and Knacke, T., "Recovery systems design guide," Report, Irving Industries Inc., 1978.

[71] Dubins, L. E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, 1957, pp. 497–516.

[72] Ljung, L., "System Identification: Theory for the User, PTR Prentice Hall Information and System Sciences Series," 1999.

[73] Bai, E.-W., "Identification of systems with hard input nonlinearities," *Perspectives in robust control*, Springer, 2001, pp. 1–12.

[74] Bai, E.-W., "Decoupling the linear and nonlinear parts in Hammerstein model identification," *Automatica*, Vol. 40, No. 4, 2004, pp. 671–676.

[75] Chan, K. H., Bao, J., and Whiten, W. J., "Identification of MIMO Hammerstein systems using cardinal spline functions," *Journal of Process Control*, Vol. 16, No. 7, 2006, pp. 659–670.

[76] Ding, F. and Chen, T., "Identification of Hammerstein nonlinear ARMAX systems," *Automatica*, Vol. 41, No. 9, 2005, pp. 1479–1489.

[77] Rangan, S., Wolodkin, G., and Poolla, K., “Identification methods for Hammerstein systems,” *Proceedings of Control and Decision Conference*, 1995, pp. 697–702.

[78] Vörös, J., “Parameter identification of discontinuous Hammerstein systems,” *Automatica*, Vol. 33, No. 6, 1997, pp. 1141–1146.

[79] Billings, S. and Fakhouri, S., “Identification of systems containing linear dynamic and static nonlinear elements,” *Automatica*, Vol. 18, No. 1, 1982, pp. 15–26.

[80] Greblicki, W., “Nonlinearity estimation in Hammerstein systems based on ordered observations,” *Signal Processing, IEEE Transactions on*, Vol. 44, No. 5, 1996, pp. 1224–1233.

[81] Pawlak, M., “On the series expansion approach to the identification of Hammerstein systems,” *Automatic Control, IEEE Transactions on*, Vol. 36, No. 6, 1991, pp. 763–767.

[82] Bai, E.-W., “An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems,” *Automatica*, Vol. 34, No. 3, 1998, pp. 333–338.

[83] Boutayeb, M., Aubry, D., and Darouach, M., “A robust and recursive identification method for MISO Hammerstein model,” *Control’96, UKACC International Conference on (Conf. Publ. No. 427)*, Vol. 1, IET, 1996, pp. 234–239.

[84] Chang, F. and Luus, R., “A noniterative method for identification using Hammerstein model,” *IEEE Transactions on Automatic control*, Vol. 16, No. 5, 1971, pp. 464–468.

[85] Hsia, T., "A multi-stage least squares method for identifying Hammerstein model nonlinear systems," *1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, No. 15, 1976, pp. 934–938.

[86] Jeng, J.-C. and Huang, H.-P., "Nonparametric identification for control of MIMO Hammerstein systems," *Industrial & Engineering Chemistry Research*, Vol. 47, No. 17, 2008, pp. 6640–6647.

[87] Saha, S., Das, S., Pakhira, A., Mukherjee, S., and Pan, I., "Comparative studies on decentralized multiloop PID controller design using evolutionary algorithms," *Engineering and Systems (SCES), 2012 Students Conference on*, IEEE, 2012, pp. 1–6.

[88] Regalia, P. A., Mitra, S. K., and Vaidyanathan, P., "The digital all-pass filter: a versatile signal processing building block," *Proceedings of the IEEE*, Vol. 76, No. 1.

[89] O'Reilly, O. M., *Engineering dynamics: a primer*, Springer Science & Business Media, 2013.

[90] Bloemen, H., Van Den Boom, T., and Verbruggen, H., "Model-based predictive control for Hammerstein-Wiener systems," *International Journal of Control*, Vol. 74, No. 5, 2001, pp. 482–495.

[91] Fruzzetti, K., Palazolu, A., and McDonald, K., "Nolinear model predictive control using Hammerstein models," *Journal of process control*, Vol. 7, No. 1, 1997, pp. 31–41.

[92] Norquay, S. J., Palazoglu, A., and Romagnoli, J., "Model predictive control based on Wiener models," *Chemical Engineering Science*, Vol. 53, No. 1, 1998, pp. 75–84.

[93] Yakimenko, O., Slegers, N., and Tiaden, R., "Development and testing of the miniature aerial delivery system snowflake," *AIAA Paper*, Vol. 2980, 2009, pp. 2009.

[94] Choi, H., *Time-Optimal Paths for a Dubins Car and Dubins Airplane with a Unidirectional Turning Constraint*, Ph.D. thesis, 2014.

[95] Al-Duwaish, H. and Naeem, W., "Nonlinear model predictive control of Hammerstein and Wiener models using genetic algorithms," *Control Applications, 2001.(CCA'01). Proceedings of the 2001 IEEE International Conference on*, IEEE, 2001, pp. 465–469.

[96] Xuefeng, Z. and Seborg, D. E., "Nonlinear predictive control based on Hammerstein models," *Control Theory and Applications*, Vol. 11, No. 5, 1994, pp. 564–575.

[97] Slegers, N. and Costello, M., "Model predictive control of a parafoil and payload system," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 816–821.

[98] Slegers, N., Kyle, J., and Costello, M., "Nonlinear model predictive control technique for unmanned air vehicles," *Journal of guidance, control, and dynamics*, Vol. 29, No. 5, 2006, pp. 1179–1188.

[99] Slegers, N., Scheuermann, E., Costello, M., and Bergeron, K., "High Fidelity In-Flight Pressure and Inertial Canopy Sensing," 2015.

[100] Ferreira, B., Matos, A., and Cruz, N., "Single beacon navigation: Localization and control of the MARES AUV," *Proceedings of MTS/IEEE conference OCEANS 2010*, 2010, pp. 1–9.

[101] Khalil, H. K. and Grizzle, J., *Nonlinear systems*, Vol. 3, Prentice hall New Jersey, 1996.

VITA

Martin Cacan was born and raised in the Southern California beach town of Santa Barbara. He received his Bachelor of Science degree in Mechanical Engineering from the University of California, Berkeley in 2011. Upon graduation, Martin moved to the Southeast to begin graduate studies at the Georgia Institute of Technology. In 2013, Martin received a Masters of Science degree in Mechanical Engineering and continued to pursue a Doctor of Philosophy. Martin's research interests include dynamic modeling, state estimation, and control system design.