# Robust Planning for Autonomous Parafoil

by

Ian Sugel

S.B., Aeronautics and Astronautics
Massachusetts Institute of Technology (2011)

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

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Aeronautics and Astronautics
August 22, 2013

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Jonathan P. How
Richard C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

# Robust Planning for Autonomous Parafoil

by

Ian Sugel

Submitted to the Department of Aeronautics and Astronautics
on August 22, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

Parafoil trajectory planning systems must be able to accurately guide the highly non-linear, under-actuated parafoil system from the drop zone to the pre-determined impact point. Parafoil planning systems are required to navigate highly complex terrain scenarios, particularly in the presence of an uncertain and potentially highly dynamic wind environment.

This thesis develops a novel planning approach to parafoil terminal guidance. Building on the chance-constrained rapidly exploring random tree (CC-RRT) [1] algorithm, this planner, CC-RRT with Analytic Sampling, considers the non-linear dynamics, as well as the under-actuated control authority of the parafoil by construction. Additionally, CC-RRT with Analytic Sampling addresses two important limitations to state-of-the-art parafoil trajectory planners: (1) implicit or explicit constraints on starting altitude of the terminal guidance phase, and (2) a reactive or limitedly-proactive approach to handling the effect of wind uncertainty.

This thesis proposes a novel formulation for the cost-to-go function, utilizing an approximation of the reachability set for the parafoil to account for the effect of vehicle heading on potential future states. This cost-to-go function allows for accurate consideration of partially planned paths, effectively removing strict constraints on starting altitude of the terminal guidance phase. The reachability set cost-to-go function demonstrates considerably improved performance over a simple LQR cost function, as well as cost-to-go functions with a glide-slope cone bias, demonstrating the effectiveness of utilizing the reachability set approximation as a means for incorporating heading dynamics. Furthermore, this thesis develops a multi-class model for characterizing the uncertain effect of wind. The wind model performs an online classification based on the observed wind measurements in order to determine the appropriate level of planner conservatism. Coupling this wind model with the method for sampling the analytic uncertainty distribution presented in this thesis, the CC-RRT with Analytic Sampling planner is able to efficiently account for the future effect of wind uncertainty and adjust trajectory plans accordingly, allowing the planner to operate in arbitrary terrain configurations without issue.

CC-RRT with Analytic Sampling performs exceptionally well in complex terrain scenarios. Simulation results demonstrate significant improvement on complex terrain relative to the state-of-the-art Band-Limited Guidance (BLG) [2], drastically reducing the worst case and average target miss distances. Simulation results demonstrate the CC-RRT with Analytic Sampling algorithm remains un-affected as terrain complexity increases, making it an ideal choice for applications where difficult terrain is an issue, as well as missions with targets with drastically different terrain conditions. Moreover, CC-RRT with Analytic Sampling is capable of starting terminal guidance at significantly higher altitudes than conventional approaches, while demonstrating no significant change in performance.

Thesis Supervisor: Jonathan P. How
Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics

# Acknowledgments

First, I would like to thank Professor How for his guidance, support, and rigorous questioning. For any problem that I faced, he had numerous suggestions for how to tackle it, and for any solution, he had questions and investigative directions to improve its quality. Professor How has been invaluable in my personal growth as a researcher and engineer.

I would like to thank Draper Laboratories for the funding for this research, through their Independent Research & Development (IR&D) program.

I would also like to thank the members of the Draper Laboratories airdrop program, specifically Amer Fejzic, Dave Carter and Chris Dever for their patient explanations and willingness to help me understand the parafoil system. Additionally, I would like to give special thanks to Louis Breger for his willingness to help and advise, as well as his invaluable guidance.

An enormous thanks to Brandon Luders, with whom I have worked so closely on this project, and without whose help, guidance, and advice this work would not have been possible. Additionally, for agreeing to provide comments to every section of this thesis, I cannot thank him enough.

I would like to thank Michelle Princi, for her love, support and reassurance throughout my entire graduate career.

Finally, I would like to thank my family. My sister Jacquie, for consistently reminding me of where I came from. My parents, for their endless love and support, and for making me into the person that I am today.

# Contents

**1 Introduction** **15**

1.1 Problem Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

1.1.1 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . 18

1.1.2 Glide-Slope Cone . . . . . . . . . . . . . . . . . . . . . . . . 19

1.2 Literature Review . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

1.3 Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

1.3.1 Success Criteria . . . . . . . . . . . . . . . . . . . . . . . . . 24

1.3.2 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

1.3.3 Planning Under Uncertainty . . . . . . . . . . . . . . . . . . 25

1.3.4 RRT Overview . . . . . . . . . . . . . . . . . . . . . . . . . . 26

1.3.5 Testing Terrains . . . . . . . . . . . . . . . . . . . . . . . . . 36

1.3.6 Comparison Algorithm: Band Limited Guidance . . . . . . . . 37

1.4 Contributions and Structure of Thesis . . . . . . . . . . . . . . . . . 39

**2 Cost-To-Go Function** **43**

2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

2.2 One step look-ahead . . . . . . . . . . . . . . . . . . . . . . . . . . 44

2.2.1 Description of Approach . . . . . . . . . . . . . . . . . . . . . 44

2.2.2 CDF and Percentile Plots . . . . . . . . . . . . . . . . . . . . 46

2.2.3 Cost-To-Go function Parameters . . . . . . . . . . . . . . . . . 50

2.3 Comparison to Alternative Approaches . . . . . . . . . . . . . . . . 59

2.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60

3 Wind Modeling 63
3.1 Introduction . . . 63
3.1.1 Model Goals . . . 64
3.2 Model Development and Description . . . 65
3.2.1 Model Form . . . 65
3.2.2 Wind Model Classifications . . . 66
3.2.3 Wind Model Components . . . 69
3.3 Classification . . . 76
3.3.1 Support Vector Machine . . . 76
3.4 Results and Comparison . . . 78
3.4.1 Miss-Classification Effects . . . 78
3.5 Summary . . . 80

4 Analytic Chance Constraints 81
4.1 Introduction . . . 81
4.1.1 Algorithm Naming Convention . . . 81
4.2 Motivation . . . 82
4.2.1 Results of RRT with Replan vs BLG . . . 82
4.2.2 Deficiency of CC-RRT with Particles . . . 84
4.3 Analytic Covariance Sampling . . . 86
4.3.1 Deriving the Distribution . . . 87
4.3.2 Parameters: Covariance Buffer . . . 91
4.4 Computational Comparison of CC-RRT with Analytic Sampling to CC-RRT Mode AB . . . 92
4.5 Summary . . . 93

5 Simulation Results 95
5.1 Simulation Configuration . . . 96
5.1.1 Wind Profiles . . . 96
5.1.2 CC-RRT Simulations . . . 96
5.1.3 BLG Simulation . . . 97

5.2 Valley Terrain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97

5.3 Flat Terrain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

5.4 CC-RRT Invariance to Terrain . . . . . . . . . . . . . . . . . . . . . . 104

5.5 High Altitude . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107

5.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108

**6 Conclusion and Future Work 111**

6.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111

6.2 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112

6.2.1 Analytic Sampling Extension . . . . . . . . . . . . . . . . . . . . 112

6.2.2 Cost Function Augmentations . . . . . . . . . . . . . . . . . . . . 113

**References 120**

# List of Figures

1-1 Army Circular Parachute System . . . . . . . . . . . . . . . . . . . . . 15

1-2 PADS Candidate System . . . . . . . . . . . . . . . . . . . . . . . . . . 17

1-3 FireFly Parafoil flying the JPADS . . . . . . . . . . . . . . . . . . . . . 17

1-4 Glide-Slope Cone Visualization . . . . . . . . . . . . . . . . . . . . . . 20

1-5 Illustration of the Sampling Strategy . . . . . . . . . . . . . . . . . . . 32

1-6 Illustration of the Sampling Distribution . . . . . . . . . . . . . . . . . 32

1-7 Illustration of Nearest Node Distance . . . . . . . . . . . . . . . . . . . 34

1-8 Illustration of the Reference Generation . . . . . . . . . . . . . . . . . . 35

1-9 Valley Terrain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

1-10 Obstacle Terrain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

2-1 Arcing Trajectories for Reachability Set Approximation . . . . . . . . 46

2-2 Miss Distance of Combined Cost Function vs Constituent Pieces . . . 49

2-3 Cost-to-Go Function Propagation Time Comparisons . . . . . . . . . 54

2-4 Visualization of Heading in Cost-To-Go Function Heat Maps . . . . . 56

2-5 Heat Map of $N = 1$ at $\theta = 0$ . . . . . . . . . . . . . . . . . . . . . . . . 57

2-6 Heat Map of $N = 3$ at $\theta = 0$ . . . . . . . . . . . . . . . . . . . . . . . . 57

2-7 Heat Map of $(N = 1) - (N = 3)$ at $\theta = 0$ . . . . . . . . . . . . . . . . 57

2-8 Heat Map of $(N = 5) - (N = 3)$ at $\theta = 0$ . . . . . . . . . . . . . . . . 58

2-9 Heat Map of $(N = 5) - (N = 3)$ at $\theta = \pi/4$ . . . . . . . . . . . . . . 58

2-10 Heat Map of $(N = 5) - (N = 3)$ at $\theta = \pi/2$ . . . . . . . . . . . . . . 58

2-11 Combined Cost-to-Go vs Glide-Slope Cost with Heading Scale CDF . 60

3-1 Regularized Cost for Filter Width Optimization . . . . . . . . . . . . 72

3-2 Tuned Wind Models for Classes 1-3 . . . . . . . . . . . . . . . . . . . . . . 77

3-3 Miss Distance CDF for Miss-Classification Test . . . . . . . . . . . . . 80

4-1 Miss Distance CDF: RRT with Replanning vs BLG . . . . . . . . . . 83

4-2 CC-RRT Mode A Worst Case Step 1 . . . . . . . . . . . . . . . . . . . 84

4-3 CC-RRT Mode A Worst Case Step 2 . . . . . . . . . . . . . . . . . . . 84

4-4 Miss Distance CDF of Mode A vs Mode AB . . . . . . . . . . . . . . 85

4-5 Zoomed Miss Distance CDF of Mode A vs Mode AB . . . . . . . . . 86

4-6 Nominal CC-RRT Mode A Plan Unacceptably Close to Obstacle Terrain 91

4-7 Crash Percentage at Standard Deviation Levels . . . . . . . . . . . . 92

5-1 Valley Terrain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98

5-2 Miss Distance CDF on Valley Terrain . . . . . . . . . . . . . . . . . . 99

5-3 Miss Distance CDF on Valley Terrain Zoom to 50m−400m . . . . . . 100

5-4 BLG Planned Trajectory vs Executed Trajectory . . . . . . . . . . . 101

5-5 Miss Distance CDF on Flat Terrain . . . . . . . . . . . . . . . . . . . 102

5-6 Miss Distance CDF on Flat Terrain zoom to 25m−200m . . . . . . . 103

5-8 Miss Distance CDF for BLG Against Various Terrain Scenarios . . . 106

5-9 Miss Distance CDF for CC-RRT with Analytic Sampling Against Various Terrain Scenarios . . . . . . . . . . . . . . . . . . . . . . . . . 107

5-10 Miss Distance CDF for CC-RRT with Analytic Sampling with Various Starting Altitudes . . . . . . . . . . . . . . . . . . . . . . . . . . . 109

# List of Tables

2.1 Miss Distance Data of Combined Cost Function vs Constituent Pieces 48

2.2 Miss Distance Data for Propagation Time Comparisons . . . . . . . . 53

2.3 Miss Distance Data for Cost-To-Go Formulation Comparisons . . . . 60

3.1 Miss Distance Data for Miss-Classification Test . . . . . . . . . . . . 79

4.1 Miss Distance Table: RRT with Replanning vs BLG . . . . . . . . . . 83

4.2 Miss Distance Data of Mode A vs Mode AB . . . . . . . . . . . . . . 86

4.3 CC-RRT Mode AB and Analytic Sampling Node Generation Times . 93

5.1 Miss Distance Data for Valley Terrain Comparison . . . . . . . . . . . 100

5.2 Miss Distance Data for Flat Terrain Comparison . . . . . . . . . . . . 103

5.3 Miss Distance Data for BLG Terrain Comparison . . . . . . . . . . . 106

5.4 Miss Distance Data for CC-RRT with Analytic Sampling Terrain Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107

5.5 Miss Distance Data for CC-RRT with Analytic Sampling with Various Starting Altitudes . . . . . . . . . . . . . . . . . . . . . . . . . . . 109

# Chapter 1

# Introduction

Airdrop systems have been used since World War II to resupply remote military positions as well as provide humanitarian relief. The first airdrop systems consisted of nothing more than crates with additional padding in order to prevent damage on impact [3]. Shortly thereafter small parachutes were added to the payloads in order to decelerate the vehicles before touchdown. An example of an Army parachute system at touchdown [4] is shown in Figure 1-1. Accurate resupply by these approaches requires the payload to be dropped from very low altitudes [5].

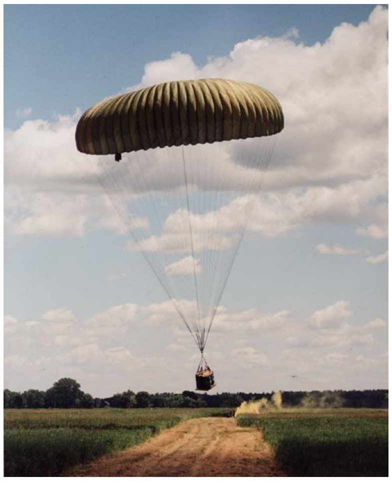


Figure 1-1: Army Circular Parachute System [4]

Higher altitude drops required significant increase in predictability of the parachute ballistic trajectory, leading to the development of the "Computed Air Release Point"

(CARP) [6] system. The CARP system integrates wind forecasting and atmospheric data, as well as parachute ballistic properties, descent rate and canopy opening trajectories learned from the significant number of previous drops into determining a release area for the parachute which is likely to result in the best landing conditions [6]. The Precision Aerial Delivery System (PADS) was developed in order to automate this process using wind data provided by weather balloons or drop team members at the target location [7].

Although incorporation of the CARP system allowed for release altitudes on the order of 5,500-7,620 m, the average landing accuracy exceeded 1,000m. Landing areas of this size either severely limit the effectiveness of airdrop as a viable supply system, or subject the system to a number of potentially devastating effects. Airdrops systems that miss the target to such an extent are subject to system and payload damage due to unintended collisions with man-made or natural hazards, theft of cargo by enemy combatants, or the airdrop system itself could cause damage. The incorporation of the PADS system resulted in average landing accuracy between 250m and 310m, depending on the transport aircraft [8, 9]. Such landing accuracy is still unacceptable in combat or in disaster relief zones. Therefore, as an alternative to the previously standard round parachute, using a parafoil instead allowed for a greater degree of control and therefore more accurate landings. The original parafoil design was based the work of Domina Jalbert [10] and researchers at Notre Dame extended the idea, and coined the term parafoil [11]. An example of a PADS candidate parafoil system at the U.S. Army Natick Soldier Systems Center [12] is shown in Figure 1-2, and a FireFly class parafoil, employing the extended Joint Precision Airdrop System [13] is shown in Figure 1-3.

To fully utilize the capabilities of the parafoil systems, many groups began developing terminal guidance systems. The terminal guidance paradigm utilizes a combination of a homing phase, designed to steer the parafoil directly toward the target, and an energy management phase, designed to descend above the target until an appropriate altitude for terminal guidance. Once terminal guidance is active, it utilizes on-board sensors as well as a feedback guidance algorithm to control the parafoil to-

Figure 1-2: PADS Candidate System [12]

Figure 1-3: FireFly Parafoil flying the JPADS [13]

ward the target. Two state of the art approaches to parafoil terminal guidance are Band Limited Guidance (BLG) [2] and massively parallel GPU processing [14]. The terminal guidance parafoil problem will be posed in the following sections and is the subject of this thesis.

## 1.1 Problem Overview

The parafoil terminal guidance problem is to generate a trajectory from an initial position $\{x_0, y_0, z_0\}$ and heading configuration $\psi_0$ to a goal position, $\{x_g, y_g, z_g\}$, or position and heading, $\{x_g, y_g, z_g, \psi_g\}$, configuration. The parafoil terminal guidance problem presents significant technical challenges, particularly for the large scale systems considered in this thesis:

1. Parafoil dynamics are highly non-linear, relying on aerodynamic drag effects for lateral control [15].

2. The parafoil is an under-actuated system, with lateral turn rates necessitating turning circles over $100m$ in diameter, and with severely limited, or no vertical control, resulting in a descent rate determined by system kinematics.

3. Parafoil drop locations have arbitrary, non-convex terrain maps, and although

these terrains are mapped in advance, the non-convexity of the terrains pose a significant problem for constraint satisfaction.

4. Parafoils are subject to uncertain and variable wind environments, which, if uncompensated, often result in unacceptably large errors between predicted and actual trajectories.

5. High-risk military and humanitarian applications often have tight landing restrictions, such as narrow valleys, adjacent to water or other terrain hazards, or along the edge of a cliff. Such restrictions prevent loss of supplies or unacceptable recovery efforts from individuals at the target location.

### 1.1.1 Problem Statement

The parafoil terminal guidance problem is a specific case of a more general trajectory planning problem. At each time step, the path planner attempts to solve the optimal control problem,

$$\begin{aligned}
\min_{\mathbf{u}(t)} \quad & \phi_f(\mathbf{x}_g, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} \phi(\mathbf{x}_g, \mathbf{x}(t), \mathbf{u}(t)) \, \mathrm{d}t, \qquad (1.1) \\
\text{s.t.} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \tilde{\mathbf{w}}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \\
& \dot{\mathbf{x}}_C(t) = f(\mathbf{x}_C(t), \mathbf{u}(t), \mathbf{w}(t)), \quad \mathbf{x}_C(t_0) = \mathbf{x}_0 \\
& \dot{\mathbf{w}} = f_w(\mathbf{w}(t), \overline{\mathbf{w}}, \boldsymbol{\nu}), \quad \mathbf{w}(t_0) = \mathbf{w}_0 \\
& \dot{\tilde{\mathbf{w}}} = f_w(\tilde{\mathbf{w}}(t), \overline{\mathbf{w}}, \mathbf{0}), \quad \tilde{\mathbf{w}}(t_0) = \mathbf{w}_0 \\
& P(\mathbf{x}_C(t = t') \in X) \geq p_{safe} \\
& \mathbf{u}(t) \in U \quad \forall t,
\end{aligned}$$

where $\mathbf{x}(t)$ is the nominal vehicle state, $\mathbf{u}(t)$ is the control input, $\mathbf{w}(t)$ are the wind disturbances to the system, $\overline{\mathbf{w}}$ are the nominal (mean) wind disturbances to the system, $\tilde{\mathbf{w}}(t))$ is the deterministic system wind, $\mathbf{x}_C(t)$ is the state represented as a distribution over the wind disturbances, $f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t))$ are the vehicle dynamics to be discussed further in Section 1.3.4, $\phi_f(\mathbf{x}_g, \mathbf{x}(t_f))$ is the cost at the terminal state, and $\phi(\mathbf{x}_g, \mathbf{x}(t), \mathbf{u}(t))$ is the cost along the path. The set $X$ is the entire set of feasible states, $U$ is the set of possible controls, $\mathbf{x}_g$ is the goal state, $t_0$ and $t_f$ are the initial and final times, respectively, $f_w(\mathbf{w}(t), \overline{\mathbf{w}}, \boldsymbol{\nu})$ is a generalized model of wind state dynamics (further discussed in Chapter 3), and $\boldsymbol{\nu}$ is the random noise governing the evolution of the wind state. Finally, the final time of the mission $t_f$ is determined by the time at which the vehicle intersects with the terrain.

Given a terrain map $T(x, y)$, which is a scalar function of $x$ and $y$ representing the height of the terrain, an initial state $\mathbf{x}_0$, an altitude time evolution $\dot{z}(\mathbf{x}(t), \mathbf{u}(t), \tilde{\mathbf{w}}(t))$, and a control sequence $\mathbf{u}(t)$, the terminal time is defined such that,

$$t_f = \left\{ t \;\middle|\; z_0 + \int\limits_{t_0}^{t} \dot{z}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tilde{\mathbf{w}}(\tau))\ \mathrm{d}\tau = T\left( \mathbf{x}_0 + \int\limits_{t_0}^{t} f(\mathbf{x}(\tau), \mathbf{u}(\tau), \tilde{\mathbf{w}}(\tau))\ \mathrm{d}\tau \right) \right\}. \tag{1.2}$$

### 1.1.2 Glide-Slope Cone

The glide-slope cone for a parafoil system is important in describing the characteristics of the parafoil guidance problem. The glide-slope surface is defined as all states $\{x, y, z, \psi\}$ such that applying no control leads the parafoil to terminate at the goal state. Assuming no wind, this is defined as

$$X_{GSS} = \left\{ (x, y, z, \psi) \;\middle|\; z = \frac{\sqrt{x^2 + y^2}}{L_D}\ ,\ \psi + \pi = \operatorname{atan2}(y, x) \right\}, \tag{1.3}$$

where atan2$(y, x)$ is the 4 quadrant arc-tangent, returning a value in $[-\pi, \pi)$ expressing the angle between the positive $x-$axis and the point $(x, y)$, and $\{x, y, z\}$ are goal relative position states. Additionally, there is an implicit $z > 0$ constraint imposed by the physical understanding that the parafoil can only descend.

This 4D surface is difficult to visualize, therefore, the glide-slope of the vehicle is typically collapsed into 3D, where it forms a cone shown in Figure 1-4 and defined by

$$X_{GSC} = \left\{ (x, y, z, \psi) \;\middle|\; z = \frac{\sqrt{x^2 + y^2}}{L_D} \right\}. \tag{1.4}$$

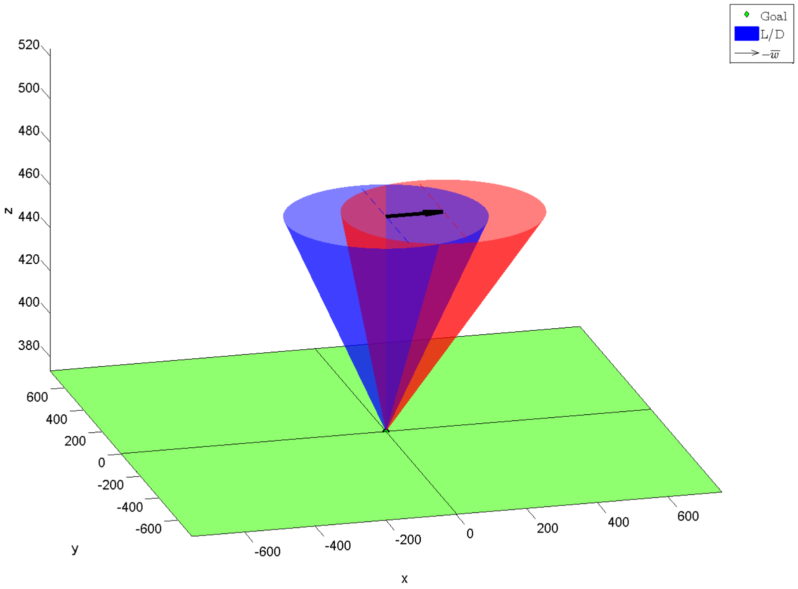


Figure 1-4: Visualization of the glide-slope cone under no wind (blue) as well as a constant wind (red). The goal is represented by the green diamond at $(0, 0)$m, and the negative of the constant wind is represented by the black arrow.

The cone representation of the glide-slope is useful as it allows for visualization of the planning space relative to this cone. The space defined by $z > \dfrac{\sqrt{x^2 + y^2}}{L_D}$ is the region where the goal remains attainable, and moreover, the parafoil maintains margin against unexpected wind disturbances further on in the trajectory. The space defined by $z < \dfrac{\sqrt{x^2 + y^2}}{L_D}$ implies a lower bound on the attainable miss distance proportional to $\left|\dfrac{\sqrt{x^2 + y^2}}{L_D} - z\right|$; moreover, in this space, no disturbances further down in the path can be addressed, without increasing this lower bound.

## 1.2 Literature Review

Since autonomous resupply is an active area of military research, there has been considerable work concerning the parafoil terminal guidance problem. Broadly speaking, these approaches can be broken down into two categories. The first category, glide-slope-based planning, utilizes the concept of the parafoil glide-slope discussed in Section 1.1.2 as a reference trajectory and builds control schemes designed to track this trajectory. The second category is the trajectory-based approach. This approach, instead of relying on a pre-determined trajectory such as the glide-slope, generates reference trajectories to the target and utilize general control schemes to track these trajectories.

Glide-slope-based approaches rely on utilizing the parafoil glide-slope as a reference trajectory to improve parafoil mission performance. Ref. [16] demonstrates an approach utilizing a series of scripted maneuvers intended to estimate the parameters required to accurately compute the glide-slope, and then executes a series of turning maneuvers to drive the parafoil to the glide-slope. This approach demonstrated the usefulness of the glide-slope as an approach trajectory as well as a set of requirements for accurately computing the surface. However, the approach heavily constrains the solution space. Moreover, this approach forced glide-slope tracking from a relatively large initial altitude above and lateral distance from the goal location, resulting in trajectory plans which are greatly affected by uncertainty in the vehicle dynamics or environment conditions (wind). Ref. [17] replaces the use of pre-scripted turning maneuvers with nonlinear Model Predictive Control (MPC) to track the glide-slope. Incorporating MPC improves the ability of the system to reject small-scale disturbances while tracking the glide-slope. However, the MPC approach suffers from the same major deficiency as the turning approach, requiring long-term tracking of the glide-slope, a surface that can be subject to environmental effects (wind) that cause changes orthogonal to the control authority of the vehicle.

Ref. [18] builds on the glide-slope tracking approach and uses feedback control based on the glide-slope and wind estimates driving the parafoil to approach the goal along the computed glide-slope estimate. This approach, known as Glide-Slope Guidance (GSG), controls the parafoil around the glide-slope in such a way as to minimize the effect of coupled system uncertainty, the combination of wind uncertainty and vehicle uncertainty, on the vehicle trajectory. This is accomplished by ensuring a maximum heading deviation from the estimated wind direction. While this approach takes some measures to account for the effect of wind uncertainty on the parafoil landing position; the approach offers no robustness to interaction with terrain obstacles, nor does it overcome the fundamental constraint of the solution space imposed by the glide-slope approach paradigm.

The glide-slope-based maneuvers, discussed above, rely on the notion of the glide-slope for planning. Trajectory-based approaches [2, 19, 20], on the other hand, generate arbitrary reference trajectories online to optimize a pre-determined cost function. Ref. [19] formulates the parafoil terminal guidance problem as an optimal control problem and leverages several optimal control techniques to determine the necessary conditions for the optimal solution. Additionally [19] notes that the optimal control conditions for the parafoil terminal guidance problem can only generate near-optimal solutions through approximation methods. However, [19] assumes a constant wind throughout the mission, and would therefore require replanning to account for wind effects during the mission.

Ref. [2] utilized a Band Limited Guidance (BLG) strategy which guarantees satisfaction of control bandwidth constraints, ensuring that the trajectory can be accurately followed. The BLG algorithm is a direct optimization technique, using the Nelder-Mead simplex search to minimize a cost function of the predicted terminal vehicle state. The BLG approach is computationally efficient, replanning at 1 Hz to account for unexpected wind effects. As a terminal guidance approach, BLG is extremely effective in practice under nominal wind and terrain conditions. However, the algorithm requires performing an optimization where the objective function is based on propagating a vehicle state from the initial state to terrain collision. In order for an approach of this type to retain computational efficiency, an upper limit on planner starting altitude must be enforced. Additionally, BLG incorporates no notion of varying future wind in the trajectory planner, and relies on reactive replanning to address varying future wind conditions.

Ref. [20] utilizes inverse dynamics (IDVD) to automatically connect the initial vehicle state to the target terminal state. This approach is computationally efficient, allowing for rapid replanning of the trajectory when encountering disturbances. Moreover, this approach guarantees satisfaction of terminal conditions by construction. The algorithm, however, cannot guarantee bandwidth constraints on the control input space, requiring iteration in order to ensure that the trajectory plan can be tracked by the controller. Moreover, the IDVD approach does not explicitly handle changing future wind, relying on rapid replanning in order to mitigate the wind effect on the resulting trajectory. Similar to BLG, this approach to wind is reactive, relying on replanning to address off-nominal wind effects.

Recently, planning techniques have begun to consider methods which are explicitly formulated to be robust to changing wind conditions. While Refs. [2, 14, 20] demonstrated approaches which allow for rapid re-planning to account for wind, Ref. [14] leverages massively parallel graphics processing unit (GPU) to consider a Monte Carlo simulation of possible future winds, based on the measurement mean and standard deviation. Using these trajectories to determine a landing distribution, and any potential path is costed based on that distribution. This is an important step in the development of parafoil terminal guidance algorithms. In contrast to the other approaches presented here, Ref. [14] considers a set of possible future winds in order to create a nominal plan robust to a number of possible wind scenarios. However, this approach requires significant computation power in the form of a GPU in order to utilize the replanning capability online. Additionally, while a collection of constant wind profiles covers the trajectory-wide effect (i.e. wind effect on landing location), it is an optimistic assumption; failing to consider dynamic wind changes may result in ignoring significant terrain interaction.

As outlined above, the general body of parafoil terminal guidance work suffers from some combination of the following major limitations.

1. Solution space artificially constrained to an a priori defined subset of the true solution space based on pre-conceived notions of the form of the solution [16–18].

2. Implicit or explicit constraints on starting altitude cause a reliance on homing and energy management phases to result in initial conditions which allow for successful terminal guidance phases [2, 14, 16–18, 20].

3. A reactive approach to handling the effect of wind uncertainty [2, 16–18, 20].

In order to address these limitations, this thesis presents an implementation of a chance-constrained rapidly exploring random tree (CC-RRT), which is a real-time general planning approach that is capable of planning from an arbitrary altitude as well as incorporating a model for future uncertainty into the planning framework.

## 1.3 Approach

### 1.3.1 Success Criteria

Given the problem statement defined in Section 1.1.1, the effectiveness of the trajectory planner developed in this thesis is measured in terms of the following success criteria:

1. The planner should result in a *mean arrival accuracy* that is comparable to the state of the art algorithm, BLG, while improving *worst case arrival accuracy.*

2. The planner should satisfy *all hard constraints*, such as turn rate constraints.

3. At all time steps, the planner should maintain a *probabilistically feasible* path, specifically accounting for wind uncertainty in the plan, the degree of feasibility to be determined a priori.

4. The planner should not be constrained by initial altitude of terminal phase.

### 1.3.2 Overview

There have been several approaches proposed for solving the parafoil terminal guidance problem. This thesis investigates the use of a chance constrained formulation of the rapidly exploring random tree algorithm (CC-RRT) [21, 22] as a solution to the problem.

### 1.3.3 Planning Under Uncertainty

The problem of motion planning under uncertainty represents a large class of problems, as described in [23, 24]. Such problems address two general types of uncertainty, process noise, which represents uncertainty in the vehicle model, and obstacle uncertainty, which represents uncertainty in the environment through sensing errors and/or uncertainty in the states of other agents in the system.

The partially-observable Markov decision processes (POMDPs) [25, 26] framework is a powerful tool for representing and planning in an uncertain environment. POMDPs develop optimal policies for partially-observable systems. However, while the POMDP approach is guaranteed to generate the optimal policy, even in uncertain systems, for state-action spaces of comparable size to the parafoil guidance problem, a real-time application is computationally intractable.

Ref. [27] demonstrates a framework for Gaussian overlap for chance constrains with an underlying $A^*$ graph search. Similar to the POMDP, to the level of chosen discretization, this approach produces a true optimal policy for partially-observable systems. This approach, however, suffers from the same scalability problem as the POMDP approach, where increased size of the $A^*$ graph leads to intractable computation times, ultimately prohibiting this approach from modelling the domain of the parafoil guidance problem.

Other approaches include the use of randomized planners to generate probabilistic roadmaps (PRMs) [28, 29]. These approaches use pre-processed maps to explore the space and provide probabilistic guarantees on path feasibility. However, PRMs are composed of linear path segments, segments which the control scheme for the parafoil is not necessarily able to accurately track, particularly in the uncontrollable dimensions (altitude).

The main algorithm of interest for this thesis is the rapidly exploring random tree (RRT) algorithm originally proposed in Ref. [30]. The RRT approach creates a tree of connected nodes and utilizes low level control to connect the nodes of the tree in a dynamically feasible manner. Similar to the PRM approach discussed above, this approach is unsuitable to be applied to the parafoil terminal guidance problem as the parafoil controller is unable to reliably and efficiently connect two arbitrary states within the space. A variant of the RRT algorithm, closed loop RRT (CL-RRT), is presented in Ref. [31, 32]. This variation converts samples into control sequences in order to create new tree nodes. These nodes do not require connection via low level control, as they are the product of simulated control sequences. This bypasses the problem of connecting piecewise linear trajectories, as well as the treatment of uncontrollable dimensions. Refs. [21, 22] extend the CL-RRT approach by using chance constraints on path feasibility, making CC-RRT an ideal approach for solving the parafoil terminal guidance problem. This thesis develops a variation of the CC-RRT approach applied to the parafoil terminal guidance problem.

### 1.3.4 RRT Overview

This section outlines the underlying RRT algorithm implemented in this thesis to solve the parafoil terminal guidance problem. In general, there are a number of components which must be addressed to implement an RRT approach. This section outlines the overarching RRT algorithm, as well as defines the required components for the RRT framework which are not contributions of this thesis, but are required for a full definition of the approach. The basic steps for tree growth outlined in [30] are:

1. Sample some state from the environment.
2. Identify the "nearest node" in the tree using a pre-defined distance metric.
3. Attempt to connect to the sample from the nearest node with a feasible trajectory.

The chance constrained formulation of the RRT algorithm, rather than generating a tree of feasible trajectories, generates a tree of state distributions that satisfy a minimum probability of feasibility.

A path (or series of nodes) from that tree is then selected in order to connect the tree root (initial state $\mathbf{x}_0$) to the goal region $X$, based on some cost function. The cost function typically has components related to the terminal state $\phi_t(\mathbf{x}_{t_f})$ as well as cost accumulated traversing the planned path through the space $\phi_p(\mathbf{x}_t, \mathbf{u}_u)$, as is typically expressed by

$$J = \phi_t(\mathbf{x}_{t_f}) + \sum_{t=0}^{t=t_f} \phi_p(\mathbf{x}_t, \mathbf{u}_u). \tag{1.5}$$

If a path being evaluated has not yet reached the goal region, then a cost-to-go is used in place of the true cost, as discussed in Chapter 2.

### Tree Growth

**Algorithm 1** Tree Growth: CC-RRT [1]

```
 1: Take a sample $x_{\text{samp}}$ from the environment
 2: Identify the $m$ nearest nodes using heuristics
 3: for  $m \leq M$ sorted nearest nodes do
 4:        $N_{near} \leftarrow$ current node, $x_{t+k} \leftarrow$ final state of $N_{near}$
 5:    while $p_{\text{safe}}(x_{t+k}) \geq p_{\text{safe}}$ and $x_{t+k}$ has not reached $x_{\text{samp}}$ do
 6:       Select input $u_{t+k}$
 7:       Simulate $x_{t+k+1}$ using (1.11)
 8:       Create intermediate nodes as appropriate
 9:       $k \leftarrow k + 1$
10:       Compute node feasibility probability $p_{\text{safe}}(x_{t+k})$
11:    end while
12:    for there each feasible node $N$ do
13:       Update cost estimates for $N$
14:       Add $N$ to the tree
15:       if Exists a probabilistically feasible connection to goal region then
16:          Update upper-bound cost-to-go of $N$ and ancestors
17:       end if
18:    end for
19: end for
```

The algorithm for generating the tree of probabilistically feasible paths, as shown in Ref. [1], is given in Algorithm 1. Similar to the original RRT algorithm from Ref. [30], Algorithm 1 generates a sample (line 1) from the environment, identifies the nodes nearest to the sample (line 2), then attempts to connect to the sample via a probabilistically feasible trajectory (lines 4-12). Heuristics are utilized to generate probabilistically feasible paths to the goal, and moreover generate "better" paths (in terms of (1.5)). The important design choices required for a CC-RRT implementation, are listed below:

1. *Cost-to-go* is an approximation of the potential remaining cost to the goal.
2. *Sampling strategy* is the methodology used to generate candidate samples. It can be general or problem specific.
3. *Vehicle and environment model* are the dynamic models governing the vehicle state and the environment scenario.
4. *Nearest node* is the metric determining the order in which nodes from the tree attempt to make connections to the sample generated in line 1.
5. *Reference generator* is the CL-RRT method of mapping a sample to the reference input space.
6. *Feasibility calculation* is the methodology used by the chance constrained algorithm to determine feasibility.

### Execution

The execution step for CC-RRT, given in Ref. [1], is reproduced in Algorithm 2. The real-time CC-RRT algorithm operates at a fixed planning frequency. Between each growth cycle of the tree, the algorithm propagates the vehicle state, and roots the tree at that state. Following that, the algorithm expands the tree using the tree growth algorithm (Algorithm 1) for the remainder of the available time. After the growth, the best path is selected based on the cost of the nodes (either cost (1.5) or cost-to-go). The trajectory associated with the chosen path is simulated to ensure probabilistic feasibility. If the path is feasible, the algorithm executes the next control associated with that path. Otherwise, the infeasible segment is pruned from the tree, and the algorithm searches for a new path.

**Algorithm 2** Execution: CC-RRT [1]

```
Input: Initial state distribution $\mathcal{N}(\hat{x_0}, P_{x_0})$, goal region $X$
 1: $t \leftarrow 0$, $x_t \leftarrow x_0$
 2: Initialize tree with node at $x_0$
 3: while $x_t \notin X$ do
 4:    Update current vehicle state $x_t$
 5:    Use measurements, if any, to repropagate state distributions
 6:    Propagate the mean state $x_t$ by the computation time $\rightarrow x_{t+\Delta t}$ using (1.11)
 7:    while time remaining for this time step do
 8:       Expand the tree by adding nodes (Algorithm 1)
 9:    end while
10:    Use cost estimates to identify best path $\{N_{\text{root}}, \dots, N_{\text{target}}\}$
11:    if no paths exist then
12:       Apply safe and goto line 19
13:    end if
14:    Repropagate the best path from $x_{t+\Delta t}$ using (1.11)
15:    if repropagated best path is probabilistically feasible then
16:       Apply best path
17:    else
18:       Remove infeasible portion of best path and goto line 9
19:    end if
20:    $t \leftarrow t + \Delta t$
21: end while
```

### Vehicle Model

The airspeed of a parafoil in steady level flight [33] is given by (1.6), with the atmospheric density model based on isothermal scale height [34] given by (1.7),

$$\begin{aligned} v(z) &= v_0\sqrt{M\frac{\rho_0}{\rho(z)}}, && (1.6) \\ \rho(z) &= \rho_{sl}e^{-z/\tau_z}, && (1.7) \end{aligned}$$

where $v_0$ is the ground level airspeed of the parafoil set to $17.8m/s$ [35], $\rho_0$ is the atmospheric density computed using (1.7), $M$ is the weight ratio of the parafoil to the 9000 kg nominal weight, $\rho_{sl}$ is the sea level atmospheric density set to $1.225\frac{\text{kg}}{\text{m}^3}$ [34], and $\tau_z$ is the isothermal scale height of earth, 10000m [34]. This airspeed is used in this work as part of the 3D extension of a Dubins vehicle used to model the position, $\{x, y, z\}$, and heading, $\psi$, of the parafoil [36]. The heading rate of the parafoil is modeled based on a second-order approximation of the Dutch roll mode of the canopy. Ref. [36] determines appropriate ranges for the second order Dutch roll model, and our approach selects randomly from those ranges in order to generate a specific model. Additionally, the differential toggle control input mechanism is modelled with a first order lag with a time constant determined in Ref. [35]. The controller is a PID controller with feedforward as proposed in Ref. [36], with gains tuned to achieve performance similar to that in Ref. [36]. The states of the lag dynamics, including the controller, are denoted by the 5th order state $\mathbf{r}$. The state vector and the control input are

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x & y & z & \mathbf{r}^T & \psi \end{bmatrix}^T, \qquad (1.8) \\ \mathbf{u} &= \dot{\psi}_d. \qquad (1.9) \end{aligned}$$

The continuous time dynamics of the system are given by combining the 3D Dubins model with the 5th order lag dynamics,

$$\begin{aligned} \dot{x} &= v(z)\cos\psi + w_x(z), \\ \dot{y} &= v(z)\sin\psi + w_y(z), \\ \dot{z} &= \frac{-v(z)}{L_D} + w_z(z), \\ \dot{\mathbf{r}} &= A\mathbf{r} + B\mathbf{u}, \\ \dot{\psi} &= \text{sat}(C\mathbf{r} + D\mathbf{u}, \omega_{min}, \omega_{max}), \qquad (1.10) \end{aligned}$$

where $L_D$ is the lift-to-drag ratio of the parafoil, 2.8 [35], $\{w_x(z), w_y(z), w_z(z)\}$ is the 3D wind profile associated with the environment model, the matrices $A, B, C, D$ are the continuous time representation of the lag dynamics, and the saturation function "sat" takes minimum and maximum turn rate arguments, $\omega_{min}$ and $\omega_{max}$ and ensure that the output lies within $[\omega_{min}, \omega_{max}]$, set to $\pm 0.2094 rad/s$ [35] (this implies a minimum turning radius of $R = v_0/\omega_{\max} = 85$m for the parafoil). In order to implement a real-time version of the CC-RRT algorithm, the dynamics in (1.10) must be discretized. Using the discretization time step, $\Delta t$, and the indexing $k$, the discrete time dynamics are

$$
\begin{aligned}
x_{k+1} &= x_k + \Delta t \left( v(z_k)\cos\psi_k + w_{x,k} \right), \\
y_{k+1} &= y_k + \Delta t \left( v(z_k)\sin\psi_k + w_{y,k} \right), \\
z_{k+1} &= z_k + \Delta t \left( \frac{-v(z_k)}{L_D} + w_{z,k} \right), \\
\mathbf{r}_{k+1} &= \mathbf{r}_k + \Delta t \left( A\mathbf{r}_k + B\mathbf{u}_k \right), \\
\psi_{k+1} &= \psi_k + \Delta t sat \left( C\mathbf{r}_k + D\mathbf{u}_k, \omega_{\min}, \omega_{\max} \right). \qquad (1.11)
\end{aligned}
$$

### Sampling Strategy

In sample-based planning algorithms, the weighting of sample space is extremely important in the ultimate outcome of the planner [23]. The sampling strategy can be broken up into two parts: random sampling, which generates samples randomly across the entire planning space, and directed sampling, which creates samples geared toward incorporating specific planning options into the tree.

**Random Sampling** 4 regions of interest within the 3-D Euclidean planning space have been identified. When a new sample is required, the algorithm randomly selects a region based on tunable probabilities assigned a priori. The regions are

1. **Goal Sampling** Generates a sample around the goal based on a hemisphere, where the radius is sampled from a Folded-Normal distribution, with mean and standard deviation determined by tunable parameters and azimuth and elevation angles drawn uniformly from $[-\pi, \pi]$ and $[0, \pi/2]$, respectively.

2. **Local Sampling** Generates a sample in a sphere around the parafoil, where the radius is sampled from a Folded-Normal distribution, with mean and standard deviation determined by tunable parameters and azimuth and elevation angles drawn uniformly from $[-\pi, \pi]$ and $[-\pi/2, \pi/2]$, respectively.

3. **Line Sampling** Generates a sample in an sphere around a convex combination of the parafoil location and the goal location, with combination parame-

ter $\lambda$ drawn uniformly from $[0, 1]$, with the radius is sampled from a Folded-Normal distribution, with mean and standard deviation determined by tunable parameters and azimuth and elevation angles drawn uniformly from $[-\pi, \pi]$ and $[-\pi/2, \pi/2]$, respectively.

4. **Global Sampling** Generates a sample uniformly across the entire space.

Figure 1-5 shows an illustration of the sampling regions, as well as the probability of selecting a sample from each region. Figure 1-6 shows a heat map of the sampling distribution, showing the relative sample density. In this example, the parafoil is located at $(-300\text{m}, -300\text{m})$ and the goal at $(0, 0)$m.

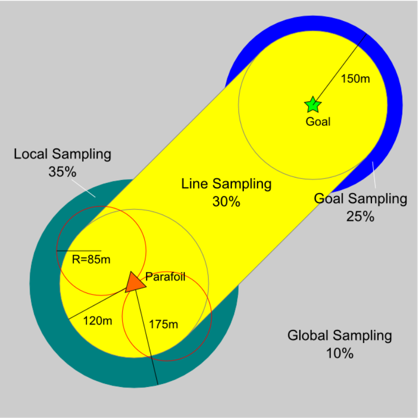


Figure 1-5: Illustration of the sampling strategy. Each of the sampling regions (goal sampling, local sampling, line sampling, global sampling) are indicated, along with the parafoil (red triangle), parafoil minimum-radius turning circles (red circles) and the goal (green star)

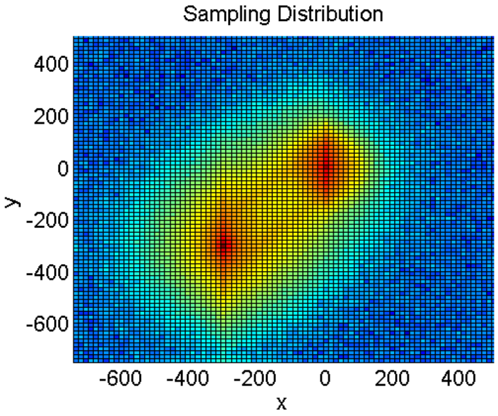


Figure 1-6: Heat map of the sampling distribution. The parafoil is located at $(-300m, -300m)$ and the goal at $(0, 0)$.

**Directed Sampling** Two types of direct sampling have been identified that are useful for the parafoil guidance problem. After every sample generated by random sampling, the resulting nodes attempt to make a connection based on the following two direct sampling methods

1. **Connect to Goal** Generates a sample at the goal location and attempts to make a connection to it.

2. **Turn Around** With probability

$$P\left(\chi \;<\; \frac{\sqrt{x^2+y^2}(\Delta\psi)^2}{r_{ds}R\pi^2},\;\; \chi \sim U(0,1)\right),$$

   where $x$ and $y$ are states of the node, $\Delta\psi$ is the heading of the node relative to pointing toward the goal location, $R = v_0/\omega_{\max} = 85\text{m}$ is the minimum turning radius of the parafoil, $r_{ds}$ is the tuning parameter, and $\chi$ is randomly drawn from a uniform distribution on $[0,1]$, places a sample at $(-R, \text{sign}(\dot{\psi})R)$ in the frame parallel and perpendicular to the parafoil. This sample forces a turn around maneuver. This is an advantageous option for the parafoil under a combination of two conditions, (1) the parafoil is far from the goal (2) the parafoil is facing away from the goal. These conditions are expressed in the probability of executing this direct sample.

### Nearest Node

Standard nearest node metrics are distance functions in a metric space, which are used as a method of determining which node in the tree a new sample is connected to. Examples of such functions are Hamming distance, Manhattan distance and Euclidean distance. For this work, the distance $d$ is a variant of Euclidean distance which does not penalize altitude above the sample based on propagation along the glide-slope

$$d = \sqrt{(x_n - x_s)^2 + (y_n - y_s)^2 + \left(z_n - z_s - \frac{\sqrt{(x_n - x_s)^2 + (y_n - y_s)^2}}{L_D}\right)^2}, \qquad (1.12)$$

where $\{x_n, y_n, z_n\}$ is the position of the node, $\{x_s, y_s, z_s\}$ is the position of the sample, and $L_D$ is the lift-to-drag ratio of the parafoil. Figure 1-7 illustrates the effect of the nearest node distance relative to the true Euclidean distance. Additionally, this metric has the effect of biasing toward connections when a sample is below the node. This is an advantageous bias, as the parafoil has no method of vertical control, and therefore must descend to all future states.

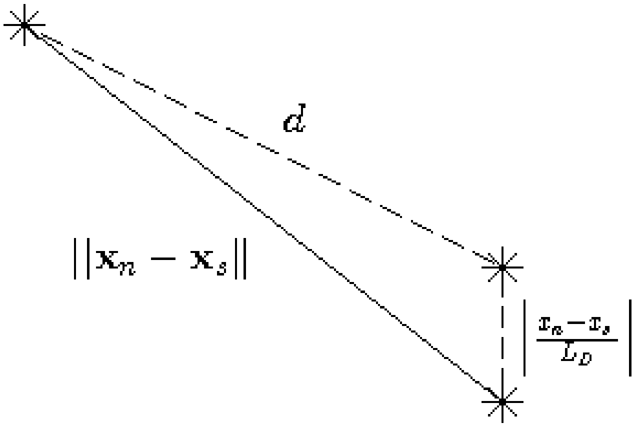

$d$
$||\mathbf{x}_n - \mathbf{x}_s||$
$\left|\frac{z_n - z_s}{L_D}\right|$

Figure 1-7: Illustration of Nearest Node Distance

### Reference Generator

The underlying CL-RRT planner requires samples generated be in the space of the reference input, instead of a waypoint to which the vehicle is controlled. For this work, a circular arc connecting the nearest node with specified orientation, $\{x_n, y_n, z_n, \psi_n\}$, to the sample location $\{x_s, y_s\}$ is used. This is accomplished by determining the radius, $R$ of the circle connecting the two points, shown in Figure 1-8:

$$\begin{aligned} \delta_x &= x_s - x_n, \\ \delta_y &= y_s - y_n, \\ \delta &= \sqrt{\delta_x^2 + \delta_y^2}, \\ R &= \frac{\delta^2}{2(\delta_y \cos\psi_n - \delta_x \sin\psi_n)}. \end{aligned} \tag{1.13}$$

(1.13) is not a true radius, as it can be negative, however, the sign of $R$ does encode the turn direction. The angular rate reference $\dot{\psi}_d$ is computed using the horizontal velocity model (1.6) and the radius $R$,

$$\dot{\psi}_d = \frac{v_a(z_n)}{R}. \tag{1.14}$$

The duration of the reference command is computed by determining the angle $\theta$ traversed about the circle connecting the node and the sample using

$$\theta = 2\sin^{-1}\left(\frac{\delta}{|2R|}\right). \tag{1.15}$$

The angle $\theta$ is then used to compute the time, $t_d$, over which the reference is commanded,

$$t_d = \frac{\theta}{\dot{\psi}_d}. \tag{1.16}$$

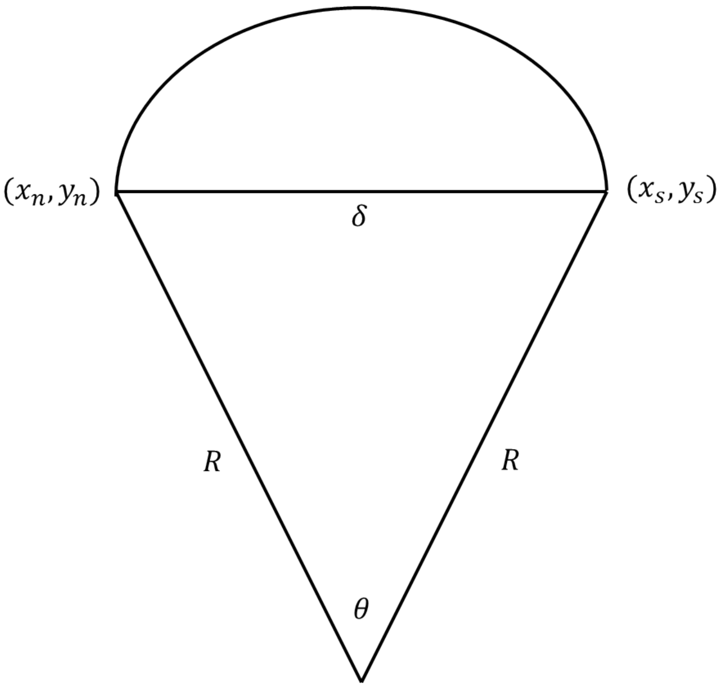
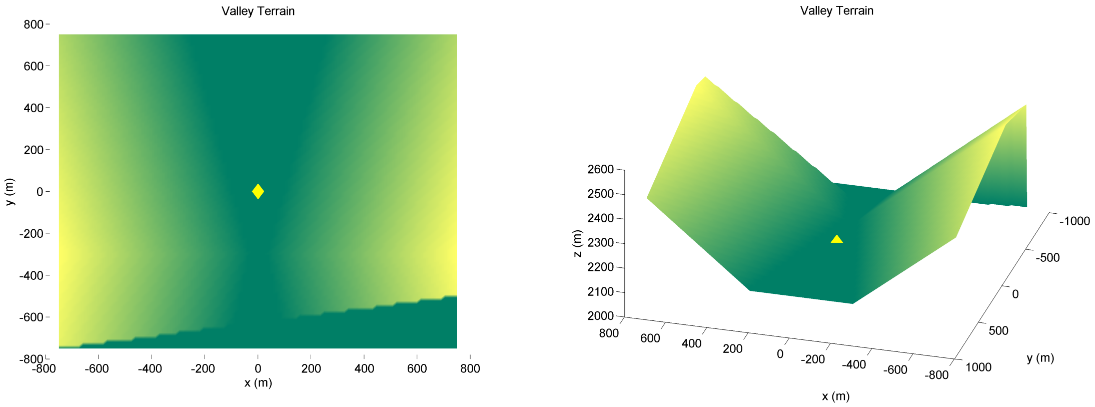


Figure 1-8: Illustration of the Reference Generation

### 1.3.5 Testing Terrains

#### Valley Terrain

Figures 1-9(a) and 1-9(b) show the difficult valley terrain used to validate the CC-RRT algorithm, where green terrain is at low altitude, yellow terrain is high altitude, and the yellow diamond is the goal location. This terrain is difficult based on three design features. These features are: (1) the slope of the valley is greater than the glide-slope of the parafoil, severely limiting the planning options for the parafoil at lower altitudes, (2) the parafoil goal is located at a terrain "bottleneck", making feasible plans away from the goal relatively simple, while making feasible plans toward the goal relatively more difficult, (3) a global minimum (bottom right of Figure 1-9(a)) far from the goal, with a drastic elevation change between the minimum and the goal for the planner to avoid when leaving the global-minimum region.



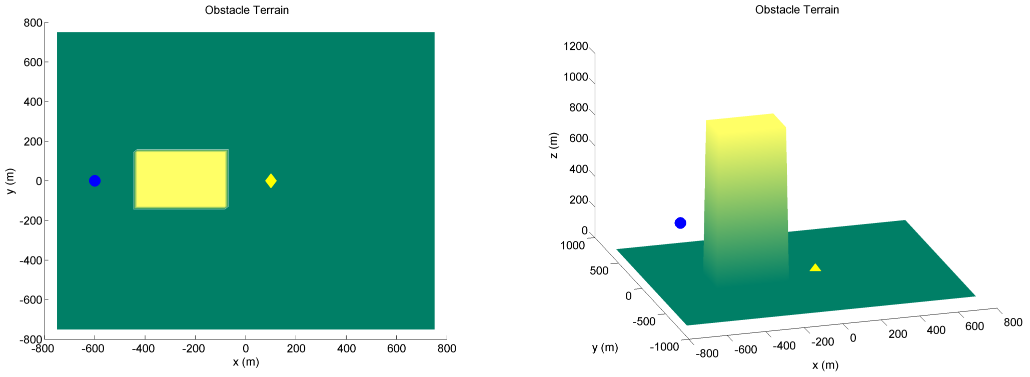
(a) Valley Terrain Top View

(b) Valley Terrain Skew-Side View

Figure 1-9: Valley Terrain Images. Green indicates lower altitude, with yellow shades indicating higher altitude. The parafoil goal is located at the yellow diamond.

#### Obstacle Terrain

Figures 1-10(a) and 1-10(b) show the obstacle terrain used to analyze the value of robustness of the CC-RRT approach. This terrain situates a terrain obstacle between the parafoil starting location (blue circle) and the goal (yellow diamond). Testing on this terrain is intended to be nearly pathological. This terrain has a start/goal pair, $(-700\text{m}, 0\text{m}, 350\text{m})$ and $(100\text{m}, 0\text{m})$, respectively, that encourages paths that closely approach the obstacle with a wind scenario that constantly forces the parafoil toward the $y = 0$ line. This scenario is intended to highlight one of the specific cases in which robustness to wind effects and terrain play a critical role in the success of a parafoil mission.



(a) Obstacle Terrain Top View

(b) Obstacle Terrain Skew-Side View

Figure 1-10: Obstacle Terrain Images. Parafoil start location shown as a blue circle; goal location shown as a yellow diamond.

### 1.3.6 Comparison Algorithm: Band Limited Guidance

The BLG algorithm proposed and evaluated in Ref. [2] will be used as a point of comparison for this thesis. The BLG algorithm determines an optimal control by choosing coefficients $\psi'_k$ for the heading rate profile,

$$\psi'(h) = \sum_{k=0}^{N} \psi'_k \frac{\sin(\pi(h - k\Delta z)/\Delta z)}{\pi(h - k\Delta z)/\Delta z}, \tag{1.17}$$

where $\Delta z$ and $N$ are the band-limiting parameters described in Ref .[2]. The simplified dynamics,

$$\begin{aligned} x' &= -L_D\cos(\psi) + w_x/\dot{z}, \\ y' &= -L_D\sin(\psi) + w_y/\dot{z}, \\ (\cos(\psi))' &= -\psi(z)'\sin(\psi), \\ (\sin(\psi))' &= \psi(z)'\cos(\psi), \end{aligned} \tag{1.18}$$

expressed as differential equations with respect to altitude by $(\cdot)'$, are simulated forward and a cost function consisting of a weighted sum of $x^2$, $y^2$, and $(\sin(\Delta\psi/2))^2$, where $\Delta\psi$ is the difference between final heading and desired heading is optimized [2]. The vehicle model used in this planning algorithm is fundamentally different from the model presented in Section 1.3.4 in the way that heading rate is handled. In the vehicle model used in the thesis (Section 1.3.4), the heading rate is the result of a linear model, whereas the BLG vehicle model assumes full control over the heading rate with no lag, provided that the controls are bounded by the BLG control equation (1.17). This approach has been implemented in MATLAB in order to provide comparison to the CC-RRT simulation set-up.

### Algorithm Implementation

**Initial Guess** The parafoil terminal guidance problem in general is a non-convex optimization, and therefore, there is significant chance for solution convergence to a local minimum. It is, therefore, important to select an appropriate initial guess for the optimization algorithm. For this implementation of the BLG algorithm, the following procedure to determine the initial guess is used.

1. The BLG algorithm is initialized with all coefficients $\psi'_k$ initialized to the same constant value. This value is chosen from that list of candidate constants.

2. Generate a list of candidate constants. This list of constants, $\Psi$, is generated by first defining the following quantities,

$$\psi'_a = \frac{1}{z_0}\cos^{-1}\left(\frac{-x_0\cos\psi_0 - y_0\sin\psi_0}{\sqrt{x_0^2+y_0^2}}\right) \tag{1.19}$$

$$\psi'_b = \frac{1}{z_0}\left(2\pi - \cos^{-1}\left(\frac{-x_0\cos\psi_0 - y_0\sin\psi_0}{\sqrt{x_0^2+y_0^2}}\right)\right) \tag{1.20}$$

where, $\{x_0, y_0, z_0, \psi_0\}$ represent the initial state of the parafoil, and $\cos^{-1}$ is the arc-cosine function. $\psi'_a$ and $\psi'_b$ represent minimum angle opposite direction constant turn-rate turns which align the parafoil heading with the goal at touchdown. The list of candidate solutions is then generated as,

$$\Psi = \begin{bmatrix} \omega_{\min} & -\psi'_b & -\psi'_a & 0 & \psi'_a & \psi'_b & \omega_{\max} \end{bmatrix}. \tag{1.21}$$

3. Using the BLG parafoil dynamics model (1.18), each of the potential initializations from the candidate list is simulated until termination and then a cost is assigned to each of the candidate constants.

4. The BLG algorithm is then initialized using the element of the candidate list which results in the lowest cost.

This procedure considers initial guesses in many regions of the solution space, allowing the guess to be close to the global or an attractive local minima.

**Optimization Algorithm** The BLG algorithm requires a minimization of a non-convex objective function. For this application, the MATLAB `fmincon` function is used. The `fmincon` function uses sequential quadratic programming (SQP) ([37]). The SQP method optimizes a quadratic approximation of the objective function subject to a linearization of the system constraints. MATLAB uses the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method quasi-Newton with line search ([37]) which utilizes objective function gradient change information to iteratively build a Hessian matrix approximation. Details of SQP, as well as the BFGS method are found in Chapter 18 of Ref. [38].

## 1.4 Contributions and Structure of Thesis

This thesis presents a novel variation of the CC-RRT algorithm applied to parafoil terminal guidance. Parafoil missions are subject to significant disturbances, with winds over 95% of the vehicle airspeed. Moreover, many parafoil missions require navigating aggressive terrain scenarios. Each chapter offers contributions toward fulfilling the success criteria from Section 1.3.1.

- **Chapter 2:** A novel cost-to-go function for the parafoil problem is proposed, developed, and analyzed. This cost-to-go approach is based on a fixed-horizon discrete approximation of a reachability set. This cost-to-go approach allows for the consideration of partial paths in the planning process, improving the planners ability to adapt to varying mission configurations as well as reducing the impact of planner start altitude on solution quality. Specific parameters of the cost-to-go function are analyzed to determine the most favorable configuration.
- **Chapter 3:** A multi-class colored noise wind uncertainty model is developed. The model uses real-time observed wind data to classify the wind uncertainty environment online. The model of each cluster considers a non-zero-mean random process; where the mean is estimated using a moving average of the observations, and the additional transient is modeled using an altitude-driven colored noise. This model differs significantly from those proposed in the literature as it considers the spacial and temporal evolution of the statistical characteristics of the wind along the parafoil trajectory, while the literature considers primarily the wind estimation problem, wind prediction over large areas (using sensor arrays) or long time-scale prediction. The effectiveness of the proposed multi-class model is then characterized and compared to utilizing a single classification.
- **Chapter 4:** Utilizing the multi-class wind model developed in Chapter 3, Chapter 4 presents an analytic method for determining the uncertainty distribution of the vehicle state. The method utilizes the analytic uncertainty distribution to create a finite set of samples which uniformly cover the uncertainty space, resulting in an efficient method for constraint checking, which unlike the "particle" formulation proposed in Ref. [1], does not requires dynamic state propagation.
- **Chapter 5:** Demonstrates simulation results of the new CC-RRT with Analytic Sampling algorithm on the valley terrain scenario, and compares these results with a nominal CL-RRT formulation (with only mean wind knowledge), with a CC-RRT formulation utilizing a particle approximation of the uncertainty distribution, as well as with the state-of-the-art BLG formulation. This chapter

also demonstrates the ability of the CC-RRT with Analytic Sampling algorithm to operate with various initial altitudes

- **Chapter 6:** This chapter presents conclusions and recommendations for future work.

# Chapter 2

# Cost-To-Go Function

## 2.1 Introduction

One of the main advantages of the RRT framework is the capability to select a path which has not yet terminated in planning, use it as the basis for vehicle execution, and complete the plan for the path during a later planning cycle. Critical to this capability is an informative cost-to-go function, which must provide the ability to compare two prospective paths which have not yet terminated, as well as to compare paths that have not terminated with actual path costs of paths that have terminated in planning. Cost-to-go functions are utilized in many different works, but are rarely the main focus of the work. [39, 40] consider a cost-to-go function in the context of a mixed integer linear program. This cost-to-go function, very naturally, follows the form of the objective function that the work seeks to optimize and takes the form of a weighted $L_1$ norm. [41] extends the previous approach to include decent consideration as well as visibility of the target point. Extending these ideas to the parafoil, one might consider a cost-to-go function as a weighted sum of the position state. However, as we shall see in Section 2.6, such an approach is insufficient for this application, as it does not consider the effect of heading on the outcome of the cost-to-go. This chapter presents a cost-to-go formulation which combines the cost at the point of consideration with a discrete approximation of the reachability set.

## 2.2 One step look-ahead

### 2.2.1 Description of Approach

We seek a method for determining the cost-to-go for a given state $\{x, y, z, \theta, \mathbf{r}\}$, defined in (1.8). Our approach combines the use of a point cost primitive with a reachability set approximation to incorporate the effect of approach direction on the cost. For this cost-to-go function, we ignore the lag dynamic states $\mathbf{r}$ so as to simplify computation.

#### Cost at a Point

A straightforward choice for the cost-to-go for a single point is the Euclidean 2-norm of the vector connecting the point to the goal location (assumed to be $\mathbf{0}$). We have incorporated a slight modification in accounting for the drift associated with the persistent wind, discussed in Section 3.2.3,

$$J_i = \sqrt{(x_i - t_g \overline{w}_x)^2 + (y_i - t_g \overline{w}_y)^2 + z_i^2}. \tag{2.1}$$

where $t_g$ represents the time to descend from the current altitude to the goal altitude. The use of this cost-to-go function represents the desire to terminate at the goal, but accounts for the persistent wind effect, as well as demonstrating the preference for nodes closer in altitude to the termination point, as captured by the $z_i^2$ term.

#### Reachability Set Approximation

The use of a reachability set addresses the intention to incorporate the effect of approach direction on the cost of a particular node. The full reachability set of a state is defined as all possible future states of the system, which, in the case of the parafoil, can only be constructed by propagating all input sequences from the initial state until intersection with the terrain. This is extremely difficult to achieve in practice, therefore, in order to simplify the computation as well as regulate the overall effect of heading on the cost function we consider only an approximation of the reachability set described by a propagation of a finite number of possible input sequences for a finite time.

In order to estimate the reachability set for a given number of input sequences, $N$, and propagation time $t$, we elect to use a set of equi-spaced, constant control inputs. Constant inputs allow for analytic determination of the points in the reachability set approximation, allowing for a computationally efficient approach. Choosing equally spaced points allows the approximation to cover the largest portion of the space for a given number of points in the approximation. If $N = 1$, then a constant input of $\omega = 0$ is used; if $N > 1$, then the control inputs are determined for index $i \in \{1, \ldots, N\}$ by

$$\omega_i = \omega_{\min} + \frac{\omega_{\max} - \omega_{\min}}{N - 1} i, \tag{2.2}$$

where $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum allowable turn rates of the vehicle, as defined in Section 1.3.4.

Using the resulting set of control inputs, $\omega_i$, as well as the time $\tau$, the minimum of the propagation time $t$ and the time to descend to goal altitude, each tuple $\{x_i, y_i, z_i\}$ representing a point in the reachability set approximation is computed using

$$\begin{aligned}
x_i &= x + \left|\frac{v}{\omega_i}\right| \left(\cos(\psi + \mathrm{sign}(\omega_i)\frac{\pi}{2}) + \cos(\psi + (2 + \mathrm{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i \tau)\right), & (2.3)\\
y_i &= y + \left|\frac{v}{\omega_i}\right| \left(\sin(\psi + \mathrm{sign}(\omega_i)\frac{\pi}{2}) + \sin(\psi + (2 + \mathrm{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i \tau)\right), & (2.4)\\
z_i &= z - \frac{v}{L_D}\tau. & (2.5)
\end{aligned}$$

Figure 2-1 illustrates the result of the reachability set approximation using $N = 3$ and $t = T/4$. In this framework, each of the $\omega_i$ represent a control choice that the planner is able to make, and each of the points $x_i$ represent the result of each of those choices. An increase in $N$ implies that the approximation is aware of more options for planner control choice (i.e. a larger and more closely spaced set of $\omega_i$), allowing for a more densely packed representation of the reachability set. A larger $t$ represents a reachability set which looks further forward, implying that the planner must hold constant inputs for longer before it is able to make another planning choice.

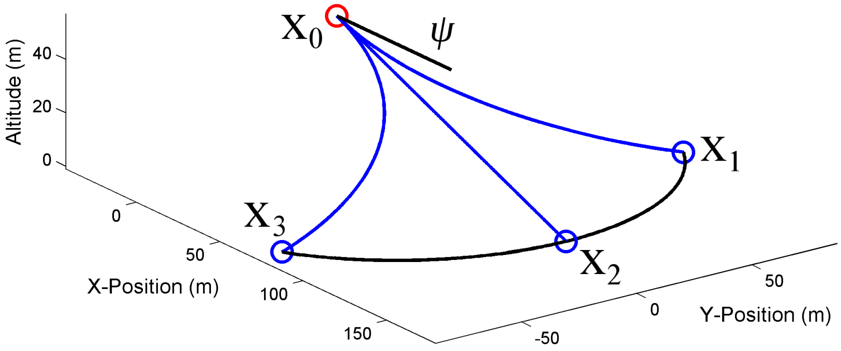


Figure 2-1: Arcing Trajectories for Reachability Set Approximation.From the point considered ($x_0$, red circle), the reachability set approximation ($x_1, x_2, x_3$) are computed relative to the parafoil heading $\psi$.

### 2.2.2 CDF and Percentile Plots

Aggregate cumulative density function (CDF) plots, individual percentile plots, as well as tabular representation of the percentile data are used throughout this thesis to demonstrate the relative performance and effectiveness of various strategies. Due to the inherent randomization in the RRT, and by extension CC-RRT, algorithm, the outcome and planner decisions of a single simulation trial are useful in explaining a particular trait of the algorithm, but cannot be generalized to infer operating conditions on a larger scale. Therefore, we consider aggregate representations of many runs of the algorithm in different scenarios. The CDF and percentile plots are an efficient, meaningful represntation of this data.

In the subsequent section, Figure 2-2(a) shows the CDF aggregating 500 trials of the options for the cost-to-go function. For any miss distance $d$, this plot shows the percentage of simulated results landing within a circle of radius $d$. Additionally, the plot shows 95% confidence bounds, computed using the MATLAB function `norminv`. Figure 2-2(b) shows the mean and standard deviation of the miss distance for each of the cost-to-go functions being compared, the miss distance values associated with specific percentages, as well as the 95% confidence bounds. Table 2.1 reproduces the percentage data in tabular form.

In considering data presented in this form, there are 4 key features when interpreting the data:

1. *Shape of the CDF curve*: The closer the CDF curve is to a vertical line at miss distance of $0m$, the more successful the algorithm or algorithm parameter.

2. *Mean miss distance*: The mean miss distance characterizes the nominal performance of the configuration.

3. *Mid-Valued Percentile*: The mid-value allow feature (1) to be quantified by assigning numerical values to intermediate points of the CDF plot.

4. *High Percentile and Worst Case*: The worst case and high percentile miss distances characterize how poorly the algorithm can perform.

### Combining Cost-To-Go Pieces

Now that we have developed a reachability set approximation, and have costed each of the points of the approximation, it is necessary to combine all parts into a single cost-to-go function, so that each may be compared to the others in a meaningful way. The cost-to-go function combines the costs by taking the maximum between the cost of the initial point, $J_0$, and the minimum of the cost of the points of the reachability set approximation (2.2.1),

$$J = \max(J_0, \min(J_1, J_2, \ldots, J_N)). \tag{2.6}$$

Each of the two pieces of the cost-to-go function incorporate a different understanding about the parafoil planning problem. The first piece, $J_0$, at every altitude, suggests to the planner to situate the vehicle directly above the goal, which is as far from the glide-slope as possible, while remaining inside it. Planning using a this type of cost-to-go function allows for high amount of disturbance rejection later in the path planning process.

The second piece, $\min(J_1, J_2, \ldots, J_N)$, represents the most favorable option of the reachability set approximation. The cost-to-go function considers the minimum cost propagated point (as opposed to the maximum or the average) as the planner has

Table 2.1: Miss Distance Data of Combined Cost Function vs Constituent Pieces

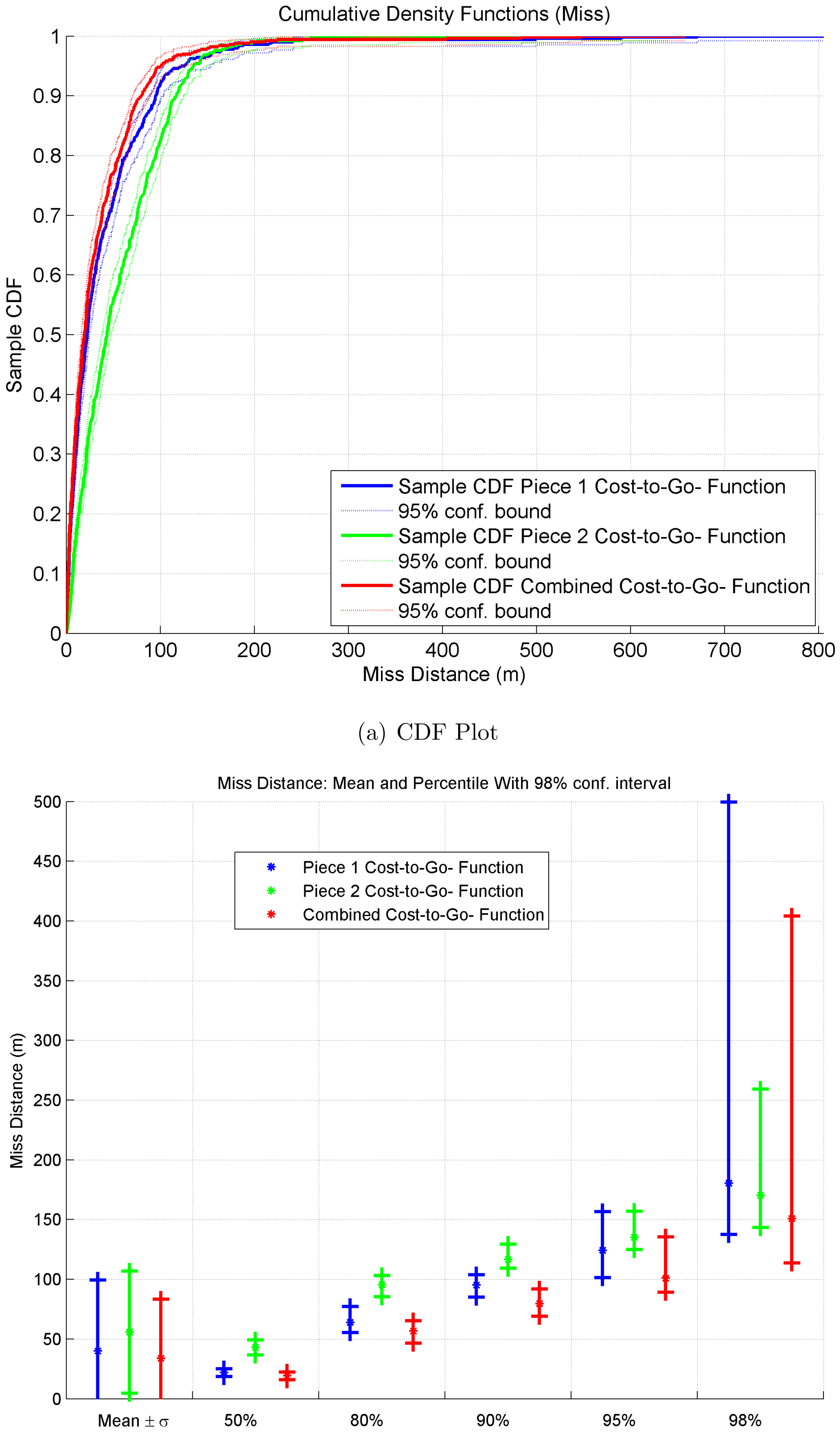
| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** |
|---|---|---|---|---|---|---|---|---|
| Piece 1 Cost-to-Go Function | 500 | 40.0 | 59.3 | 22.0 | 64.1 | 95.1 | 124 | 181 |
| Piece 2 Cost-to-Go Function | 500 | 55.9 | 51.1 | 43.2 | 95.3 | 117 | 135 | 170 |
| Combined Cost-to-Go Function | 500 | 33.9 | 49.3 | 19.3 | 56.6 | 79.7 | 101 | 151 |

the control authority to choose the executed trajectory and can therefore choose to execute the control leading to the lowest cost. When this piece of the cost-to-go function is active, i.e. $\min(J_1, J_2, \dots, J_N) > J_0$, this implies that all possible choices available to the planner are less desirable than remaining at the current state.

Figures 2-2(a) and 2-2(b) show the CDF and percentiles (with 95% error bounds), respectively, for the simulated miss distance of the full cost-to-go compared with individual pieces; Table 2.1 represents the data in tabular form. This simulation was conducted using the following simulation parameters

1. Valley Terrain described in Section 1.3.5
2. Cost-to-go function parameters $N = 3$ and $t = T/4$, where $T = \dfrac{2\pi}{\omega_{\max}}$ is the duration of one turning circle
3. Nominal RRT planner, utilizing only re-planning capabilities, and no chance constraint formulation

Figures 2-2(a) and 2-2(b) demonstrate the relative effectiveness of each of the individual pieces of the cost-to-go function, as well as the combined cost-to-go function. Considering first "Piece 2" of the cost-to-go function, $\min(J_1, J_2, \dots, J_N)$; we see that the "Combined" cost-to-go function out-performs the individual "Piece 2" in each of the important characteristics relevant to the data. "Piece 2" as a stand-alone cost-to-go function suggests optimism about the future path quality of a state for states close to the glide-slope surface (1.3), and pessimism about the future path quality of states



(a) CDF Plot

(b) Percentile Plot

Figure 2-2: Miss Distance of Combined Cost Function vs Constituent Pieces

which are far from the glide-slope surface, most notably points on the glide-slope cone (1.4), but which have a heading $\psi$ incompatible with the glide-slope surface. "Piece 1" tracks consistently with the "Combined" cost-to-go function, it fails to consider several worst case scenarios in which approaching the glide-slope cone with an incorrect heading results in significantly worse path quality, as evidenced by the high percentile ad worse case miss distances.

### 2.2.3 Cost-To-Go function Parameters

**Propagation Time**

As described in Section 2.2.1, the propagation time determines where, along its respective pre-determined trajectory, each point of the reachability set approximation is placed. As seen in Section2.6, this will effect the cost-to-go function by altering the $\min(J_1, J_2, \ldots, J_N)$ term. It is useful, therefore, to consider $t$ as it effects the symmetry of the cost-to-go function, as a cost-to-go function which is symmetric in 3D space for a particular choice of heading cannot capture the behavior of the true glide-slope surface, as discussed in Section 1.1.2. When $t = 0$, it is trivial to see that the cost-to-go function is symmetric, as all of terms in $\min(J_{i\in\{1,\ldots,N\}})$ are the same as the $J_0$ term. Consider now, the case in which $t = \frac{2\pi}{\omega_{\max}}$; moreover, assume that $N > 1$, and therefore, $\omega_1 = \omega_{\min}$ and $\omega_N = \omega_{\max}$; for a typical parafoil, $\omega_{\min} = -\omega_{\max}$, so, without loss of generality, we can consider only $\omega_{\max}$. Because propagation is done using a constant turn rate, in $x/y$ space the parafoil traces a circle, thus the $x$ and $y$ terms are periodic, contributing in the same way to the cost at different altitudes. However, since altitude is strictly decreasing, after $t = \frac{2\pi}{\omega_{\max}}$, the cost-to-go of the propagated point is strictly lower than the initial point, since both points have the same $x$ and $y$, but the propagated point has a lower $z$. This implies that for any state with $z \geq v\frac{2\pi}{\omega_{\max}}$, there exists at least one $x_i$ such that $J_i < J_0$, and therefore the node cost-to-go is $J_0$.

The exact glide-slope surface exists only when the relative heading toward the goal $\theta = 0$. Without loss of generality, the case where $\psi = 0$, $y = 0$ and $x = -zL_D$

can be considered. Additionally if the point $y = 0$, $x = +zL_D$, is considered, this point will also be on the glide-slope cone, but will not be on the glide-slope surface. As established previously, there exist two choices for propagation time $t = 0$ and $t = \dfrac{2\pi}{\omega_{\max}}$ for which these two points will have the same cost-to-go function value. Clearly, it is a poorly tuned cost-to-go function if a point on the glide-slope surface and a point which had no control option which allows it to stay within the glide-slope cone evaluate to the same cost. Consider next a propagation time $t = \epsilon$. For a sufficiently short propagation time, no matter what the commanded turn rate, the first order approximation for all propagated points is directly forward along the direction of the heading, implying that the cost of the state with $x < 0$ will remain $J_0$ as propagated points will all move toward the goal in both lateral and vertical position, and the cost-to-go of the state with $x > 0$ will increase, as any decrease in cost-to-go due to lower altitude is more than offset by an increase in cost-to-go due to lateral position. We have now established that there exists a propagation time $t = \epsilon$ such that $\min(J_1, J_2, \ldots, J_N) > J_0$ and a propagation time $t = \dfrac{2\pi}{\omega_{\max}}$ such that $\min(J_1, J_2, \ldots, J_N) < J_0$, therefore since, in this case, $\min(J_1, J_2, \ldots, J_N)$ is continuous in $t$, as the points $x_i$ trace continuous paths in $t$, there must exist a point $t = t^*$ at which $\min(J_1, J_2, \ldots, J_N) = J_0$.

The point $t = t^*$ represents the point at which the point $(x = -zL_D, y = 0)$ and $(x = zL_D, y = 0)$ have the same cost-to-go, and therefore, points on the glide-slope cone, but not on the glide-slope surface, have a symmetric cost-to-go to the points which are on the glide-slope surface. In order to prevent this cost-to-go symmetry, we seek an upper bound on $t$, namely $t^*$ such that an asymmetry between these two classes of points can be guaranteed in the cost-to-go function. This problem is written as

$$\begin{aligned} \max_{t} \quad & t \qquad\qquad (2.7) \\ \text{s.t.} \quad & J_i > J_0 \ \forall \ i \in \{1, \ldots, N\} \\ & x_0 = -z_0 L_D \end{aligned}$$

$$y_0 = 0 \text{ and } \psi_0 = 0$$
$$0 < t < \frac{2\pi}{\omega_{\max}}$$

The optimization in (2.7) can be solved analytically. For $N = 1$, this problem is trivial, as there is no $t > 0$ for which $J_N < J_0$, therefore, we assume that $N > 1$. Because $N > 1$, there exist at least $N-1$ non-zero control inputs in the discretization set $\omega_i$, and moreover, those control inputs are symmetric about 0. Therefore, the chosen control input $\omega$ can be assumed to be greater than 0. In order to solve this problem, we must find the value $t = t^*$ such that $J_0 = J_N$, where $J_N$ is the cost-to-go of the propagated point, not necessarily the first propagated point. First, we consider the propagated point $\{x_N, y_N, z_N\}$, computed using

$$\begin{aligned} x_N &= x + \frac{v}{\omega}\sin(\omega t), && (2.8)\\ y_N &= \frac{v}{\omega}(1-\cos(\omega t)), && (2.9)\\ z_N &= z - \frac{v}{L_D}t, && (2.10) \end{aligned}$$

as well as the cost-to-go associated with that point, computed using

$$\begin{aligned} J_N^2 &= x_N^2 + y_N^2 + z_N^2, && (2.11)\\ &= x^2 + 2\frac{v}{\omega}x\sin(\omega t) + \frac{v^2}{\omega^2}\sin^2(\omega t) + \frac{v^2}{\omega^2}\left(1 - 2\cos(\omega t) + \cos^2(\omega t)\right) \\ &\quad + z^2 - 2\frac{v}{L_D}zt + \frac{v^2}{L_D^2}t^2. && (2.12) \end{aligned}$$

This cost-to-go can then be equated to the cost-to-go at the initial point $\{x, y, z\}$,

$$J_0^2 = x^2 + z^2, \tag{2.13}$$

to determine a single functional relationship between $x$, $z$ and $t$ for which the cost-to-go of the initial point and the cost-to-go of the propagated point are equal,

$$J_0^2 = J_N^2 \tag{2.14}$$

Table 2.2: Miss Distance Data for Propagation Time Comparisons

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| $t = 1/4t^*$ | 200 | 37.0 | 44.3 | 23.6 | 60.7 | 84.3 | 116.9 | 133.1 | 430.4 |
| $t = 1/2t^*$ | 200 | 31.8 | 29.7 | 21.9 | 51.6 | 76.7 | 93.0 | 116.9 | 146.4 |
| $t = 3/4t^*$ | 200 | 33.0 | 32.1 | 23.2 | 53.0 | 73.4 | 101.4 | 135.9 | 170.4 |
| $t = t^*$ | 50 | 39.2 | 37.1 | 30.1 | 62.5 | 98.6 | 105.8 | 151.7 | 170.4 |

$$\begin{aligned} x^2 + z^2 &= x^2 + z^2 + 2\frac{v}{\omega} x \sin(\omega t) + \frac{v^2}{\omega^2}\sin^2(\omega t) \ldots \\ &\quad + \frac{v^2}{\omega^2}\left(1 - 2\cos(\omega t) + \cos^2(\omega t)\right) - 2\frac{v}{L_D} z t + \frac{v^2}{L_D^2} t^2, \end{aligned} \tag{2.15}$$

$$\Rightarrow z = \frac{2\frac{v^2}{\omega^2}\left(1 - \cos^2(\omega t)\right) + \dfrac{v^2}{L_D^2} t^2}{2\dfrac{v}{L_D} t} + \frac{\frac{v}{\omega}\sin(\omega t)}{\dfrac{v}{L_D} t} x. \tag{2.16}$$

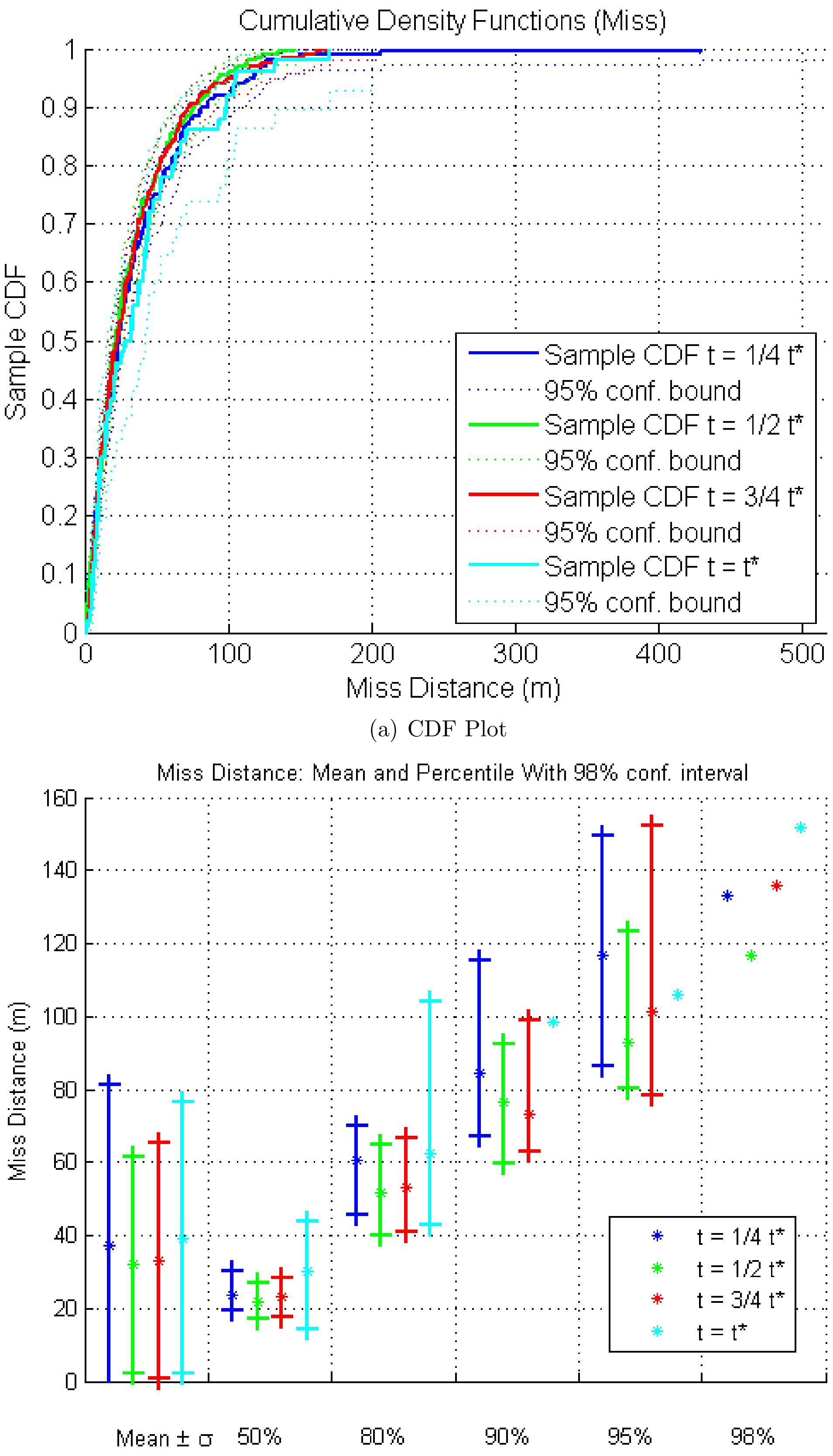
(2.16) defines an affine relationship between $x$ and $z$. The desired value $t = t^*$ is achieved when the slope of (2.16) is parallel to the glide-slope cone,

$$\frac{1}{L_D} = \frac{dz}{dx} = \frac{\dfrac{v}{\omega}\sin(\omega t^*)}{\dfrac{v}{L_D} t^*}, \tag{2.17}$$

$$\Rightarrow t^* = \frac{L_D^2}{\omega}\sin(\omega t^*). \tag{2.18}$$

(2.18) is a transcendental equation and, therefore, cannot be solved analytically for the variable $t^*$. Several numerical solution methods, including Newton's Method, Taylor series approximation as well as the secant method are used to solve such a system. Employing a numerical implementation in MATLAB and using the parameters of the dynamics model in 1.3.4 yields $\omega t^* \approx 0.906\pi$. In order to find the minimum value of $t^*$, as stated in the problem statement (2.7), the largest possible value of $\omega$ should be used, and therefore $t^* \approx 0.453T \approx 13.891$s.

Figures 2-3(a) and 2-3(b) show the miss distances for varying values of $t$, ranging from $t^*/4$ to $t^*$. There is little difference between any two values of $t$, with $t^*/2$ having the lowest overall miss distances by a slight margin.



(a) CDF Plot

(b) Percentile Plot

Figure 2-3: Cost-to-Go Function Propagation Time Comparisons

### Propagation Number

Determining the propagation number, $N$, for the reachability set approximation involves balancing the information/computation trade-off. Increasing the number of points in the approximation provides more control options for consideration in determining the the best case propagated path, but comes at the cost of increased computation cost. Moreover, since all inputs must fall within the range $[\omega_{\min}, \omega_{\max}]$, a larger value of $N$ will only discretize the a space of finite size more finely, decreasing the marginal benefit of each added propagation point. It is therefore, the goal of this section to determine the value of $N$ which accurately captures the possible future states while remaining as small as possible, in order to minimize the computation cost.

First, it is important to note that only odd values of $N$ should be included in the search. This is due to the fact that while for all values of $N$ the propagated directions are symmetric about the no-turn ($\omega = 0$) propagation option, and therefore all even values of $N$ do not consider the no-turn propagation option. This option leads to, among other things, the understanding that the vehicle being on the glide-slope facing toward the goal is a desirable location. Therefore, only odd values of $N$ should be considered for the reachability set approximation.

The following analysis considers heat maps of the cost-to-go function using different values of $N$. The cost-to-go function is determined by $t$, $N$, the position (reduced to altitude and lateral distance from the goal), as well as heading. Figure 2-4 shows a lateral cross section of the glide-slope cone as viewed from above a distance $z$ above the goal (goal shown in yellow), with the glide-slope cone a lateral distance $r$ from the goal center. Each blue arrow represents a parafoil with a true heading $\psi = \frac{\pi}{2}$, but with various headings relative to the goal location $\theta$. In order to perform this analysis, we consider heat map representations of the cost-to-go function. For each of these heat maps, the cost-to-go (2.6) is computed for a range of altitudes (y-axis), as well as lateral distances (x-axis) where a negative lateral distance can be interpreted, relative to Figure 2-4, as locations from the goal in the direction of the parafoil, and

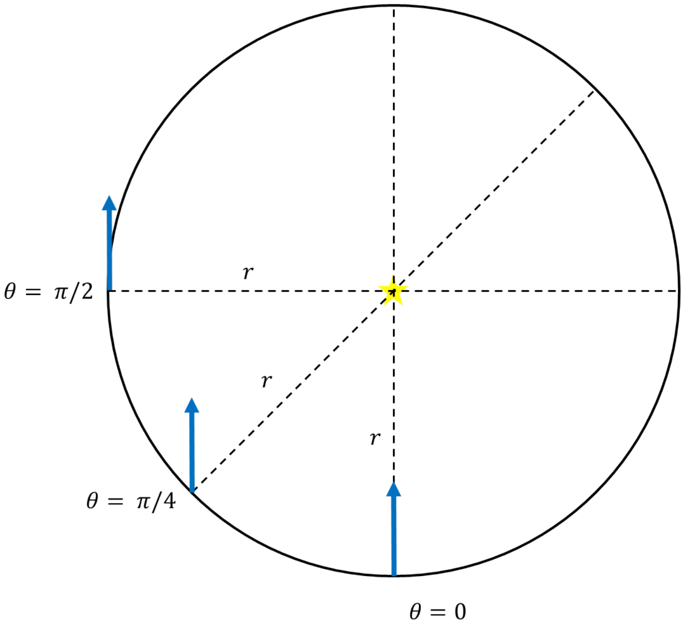


Figure 2-4: Visualization of Heading in Cost-To-Go Function Heat Maps

goal relative headings $\theta$. The heat maps presented here are 2D cross-sections of the 3D heat map $r, z, \theta$ taken at particular goal relative headings $\theta$.

First, consider the step from $N = 1$ to $N = 3$. We desire to understand what information is given to the planner by adding the two additional propagation points. Figure 2-5 depicts the heat map for $N = 1$, while Figure 2-6 shows the heat map for $N = 3$, both of them with $\theta = \frac{\pi}{2}$. The main difference between these two heat maps lies in the structure of the central region (the region directly around the $r = 0$ line). This difference can be seen even more clearly by the difference between the two heat maps, shown in Figure 2-7, where the areas of larger cost difference (more red) show a greater decrease in cost from $J_{N=1}$ to $J_{N=3}$. The addition of these lower cost regions allow for the addition of an important class of paths to be added to the planner, helical trajectories which circle about the $r = 0$ line in order to decrease the parafoil altitude before making a final turn into the goal.

Next, consider the step from $N = 3$ to $N = 5$. Similar to the previous analysis, we desire to understand what information or planning options are introduced by increasing the number of propagation points from $N = 3$ to $N = 5$. In order to do this, we consider the following 3 cost function heat maps, each showing the difference

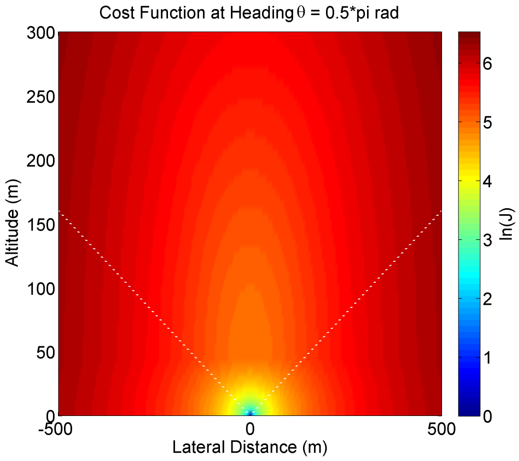


Figure 2-5: Heat Map of $N = 1$ at $\theta = 0$

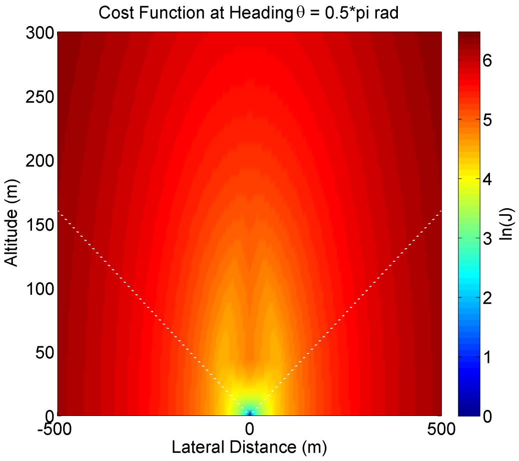


Figure 2-6: Heat Map of $N = 3$ at $\theta = 0$

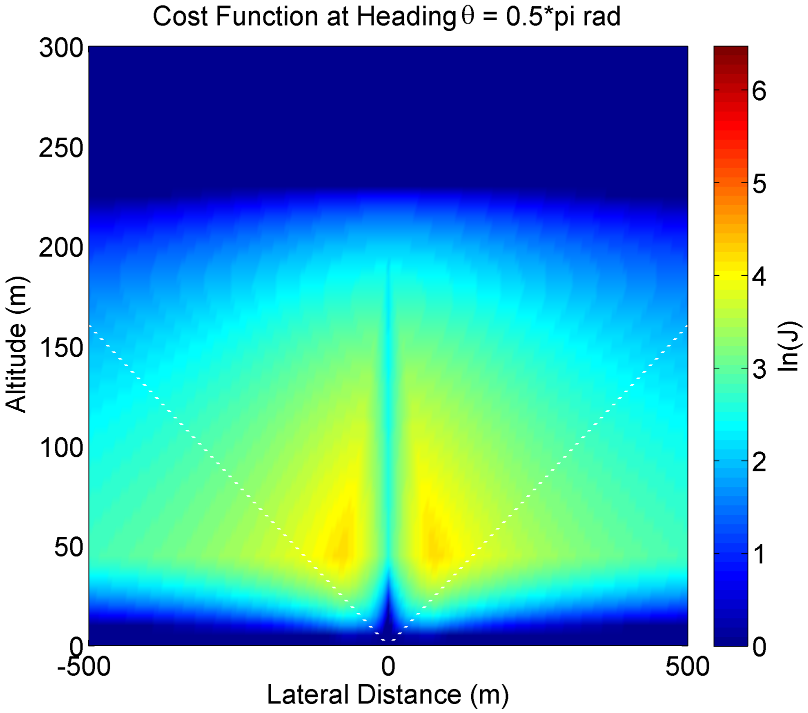


Figure 2-7: Heat Map of $(N = 1) - (N = 3)$ at $\theta = 0$

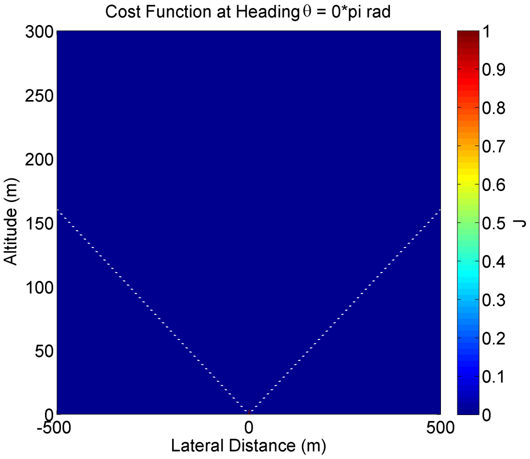


Figure 2-8: Heat Map of $(N = 5) - (N = 3)$ at $\theta = 0$

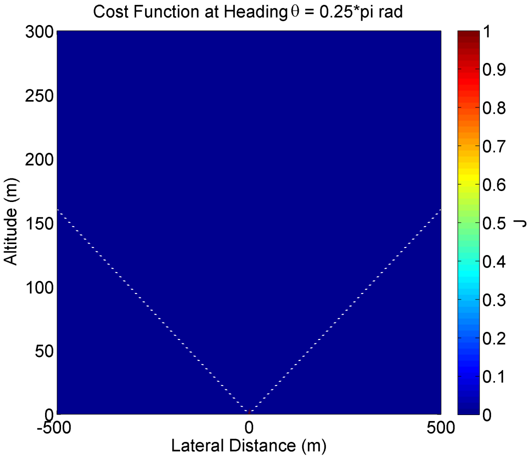


Figure 2-9: Heat Map of $(N = 5) - (N = 3)$ at $\theta = \pi/4$

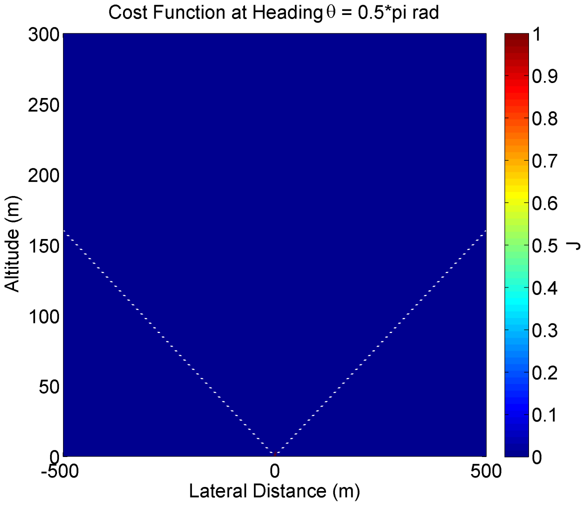


Figure 2-10: Heat Map of $(N = 5) - (N = 3)$ at $\theta = \pi/2$

between the costs for $N = 3$ and $N = 5$, with Figure 2-8 showing the cost difference when $\theta = 0$, Figure 2-9 showing the cost difference when $\theta = \frac{\pi}{4}$, and Figure 2-10 showing the cost difference when $\theta = \frac{\pi}{2}$. These figures show that none of the information incorporated by adding the 4th and 5th propagation point is utilized by the cost function, leading to the choice for number of propagation point to be $N = 3$.

## 2.3 Comparison to Alternative Approaches

**Glideslope Cone with heading scaling** Refs. [16–18] utilize the glide slope as a reference input for trajectory control of a parafoil. Following along from this idea, it would be logical to compare the approach presented in this chapter with one that biases the parafoil toward the glide slope.

$$J = \left(1 + C_3 \left(\frac{\Delta\psi}{\pi}\right)^2\right)^2 J_{\text{cone}}(x, y, z) \tag{2.19}$$

$$J_{\text{cone}}(x, y, z) = \begin{cases} z\left[(1 - C_1)\left(\frac{r(x,y)}{r_{\text{cone}}(z)}\right)^2 + C_1\right], & r < r_{\text{cone}}(z), \\ z + \frac{1}{2}C_2(r(x, y) - r_{\text{cone}}(z)), & r \geq r_{\text{cone}}(z) \end{cases} \tag{2.20}$$

(2.20) represents a cost function which is lowest along the glide slope cone, and increases proportionally to the deviation from the cone. (2.19) represents a quadratic heading penalty augmentation based on the deviation from the heading required to fully utilize the glide slope cone to arrive at the goal location. $r_{\text{cone}}(z)$ computes the radius of the glide slope cone at an altitude $z$, $r(x, y)$ computes the radius from the goal at a position $\{x, y\}$, and $\{C_1, C_2, C_3\}$ are constants.

Figure 2-11 and Table 2.3 present the results of simulation comparing the glide slope cost-to-go with the cost to go developed in this chapter. The combined cost-to-go function developed in this chapter demonstrates globally superior performance to the glide slope inspired cost-to-go function; the nominal performance improves by nearly 30%, and similar improvement can be seen for the mid and even high percentile ranges. There is no significant difference in the absolute worst case miss distances

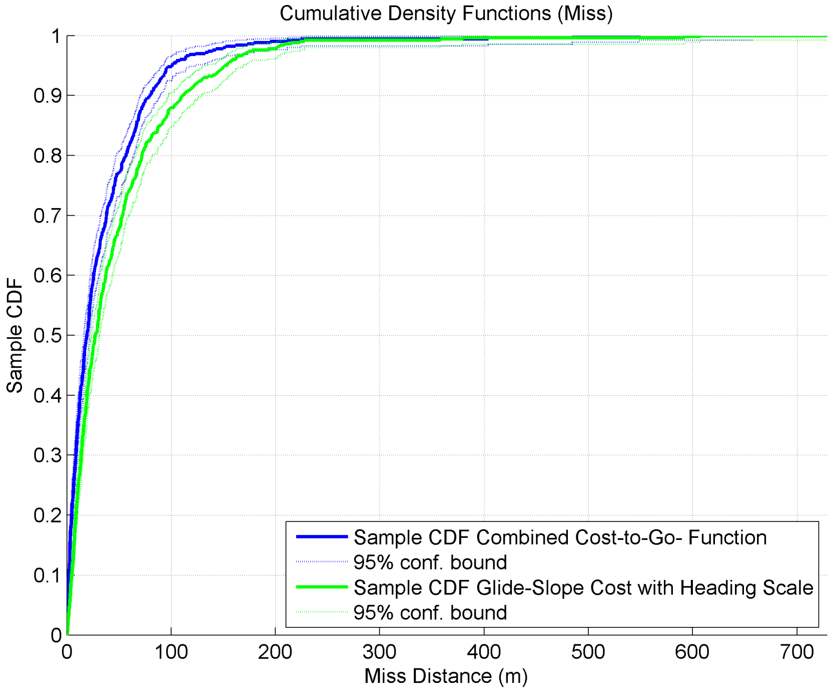


Figure 2-11: Combined Cost-to-Go vs Glide-Slope Cost with Heading Scale CDF

Table 2.3: Miss Distance Data for Cost-To-Go Formulation Comparisons

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** |
|---|---|---|---|---|---|---|---|---|
| Combined Cost-to-Go-Function | 500 | 33.9 | 49.3 | 19.3 | 56.6 | 79.7 | 101 | 151 |
| Glide-Slope Cost with Heading Scale | 500 | 47.0 | 60.5 | 27.0 | 72.1 | 111 | 152 | 202 |

for the two cost functions. This similarity is due to the lack of robustness to wind effects. Chapters 3 and 4 will address this deficiency.

## 2.4 Summary

This chapter presented a cost-to-go formulation which combines the cost at the point of consideration with a discrete approximation of the reachability set. Using the reachability set approximation, the cost-to-go function is able to accurately account for the potential effects of heading on the viability of future states. The effectiveness of this strategy is demonstrated through comparison with alternative approaches, such as a LQR cost-to-go function (“Piece 1” of Section 2.2.2), and two cost-to-go functions which favor glide-slope based trajectories (“Piece 2” of Section 2.2.2, “Glide-Slope Cost with Heading Scale” of Section 2.3).

# Chapter 3

# Wind Modeling

## 3.1 Introduction

Wind modeling is an important consideration for many different applications including aerospace, power generation, and civil engineering. As such, there has been considerable work on developing wind prediction and estimation models. Several approaches have addressed the problem of online estimation of wind. Ref. [42] utilizes a Kalman filter and an unscented Kalman filter to accurately estimate wind vectors from radar data. The approach requires either accurate airspeed measurement, or specific vehicle maneuvers (no fewer than 2 turns during the trajectory) to accurately estimate the wind vectors. Similarly, Ref. [43] also utilized a Kalman Filter approach to perform online estimates of 3D wind for an unmanned aerial vehicle (UAV) in an efficient, real-time manner. In contrast to Ref. [42], Ref. [43] utilizes only onboard sensing, GPS, IMU as well as UAV trajectory information to estimate accurate wind vectors at each point along the trajectory. Additionally, Ref. [43] demonstrates functionality of the algorithm regardless of the UAV trajectory.

While these approaches represent the solution to the very important estimation problem, they offer no manner of wind prediction forward in time (or space) for UAV systems. There have been approaches considered for other domains that begin to address the issue of predictability. In the meteorological community, Ref. [44] generated time series models for “short-term” wind prediction using wavelets. However, “short-term" refers to time scales of days, whereas a typical parafoil drop lasts minutes. Ref. [45] utilizes adaptive Gaussian processes in order to generate predictive models for wind speed. The predictive models developed require data spanning months in order to train accurately, an assumption that may be unreasonable in combat or disaster zones. Additionally, this approach utilizes the correlation which arises from continuous time-series data. Even if numerous parafoil drops have been recorded, it is unclear if and to what degree two arbitrary drops are correlated.

Current wind modeling literature has developed strategies for estimating wind along a vehicle trajectory [42, 43], as well as prediction over time scales as short as 10 minutes [44, 45]. However, none of these approaches address prediction over the time scales considered in this thesis, and moreover do not consider spatial prediction of the wind speed. This chapter fits an uncertainty model to the wind as a function of the altitude (the relevant parameter for a parafoil mission) which can be incorporated into the planner to enforce robustness. Moreover, this chapter discusses the implementation of an on-line classification scheme. This scheme uses a set of pre-determined wind classes to regulate the amount of conservatism in the planner. Performing a wind classification on-line allows the planner to dynamically adjust the level of conservatism by changing the wind classification based on observed wind conditions.

### 3.1.1 Model Goals

The wind model developed in this section is intended to be utilized by the planner to improve performance on the parafoil terminal guidance problem. The model is tuned to match the 194 wind profiles collected by Draper Laboratories [46], utilizing the wind estimation method in Ref. [2]. In order to address this, development of the wind model is broken into the following 3 objectives:

1. Maintain a simple model of wind uncertainty. A simple uncertainty model is desirable for two reasons. First, it ensures that the model is not over-fit to the data. Second, a simple model can be incorporated into the real-time operation

of the planner without utilizing significant computational resources.

2. Improve wind effect predictability over zero assumed wind. Improved predictability, especially in wind scenarios in which there is significant prevailing wind, will mitigate the amount of replanning required and greatly improve the quality of solutions generated by the RRT. Moreover, improved predictability justifies the computation effort put into growing a tree of possible solutions, as that tree is more useful as a planning tool.

3. Capture the uncertainty of future wind effects. Arguably the main purpose of this wind model is to characterize the possible effects of the wind into an uncertainty model. Developing such a model and incorporating it into the planning framework gives the planner knowledge of a distribution over possible outcomes of a planned trajectory. Utilizing this distribution in a chance constrained planner (Section 1.3.4) allows for constraint checking for probabilistic feasibility, allowing planned trajectories to ensure a level of probabilistic robustness to terrain obstacles.

## 3.2 Model Development and Description

### 3.2.1 Model Form

The 3-D wind estimate at step $k$,

$$\mathbf{w}_k = \overline{\mathbf{w}}_k + \delta\mathbf{w}_k, \tag{3.1}$$

is composed of two components: a 3-D persistent estimate, $\overline{w}_k$, and a 2-D variational estimate, $\delta w_k$. The persistent estimate,

$$\overline{\mathbf{w}}_k = \frac{1}{m}\sum_{i=k-m+1}^{k} \mathbf{w}_i, \tag{3.2}$$

is modeled using a finite impulse response filter (Section 3.2.3) to reflect the notion that there exists a prevailing wind which acts on the parafoil throughout the entire mission and must be accounted for during the state prediction. The variational estimate,

$$\delta\mathbf{w}_k = \delta\mathbf{w}_{k-1} + v_0 \Delta t \left( A_j \delta\mathbf{w}_{k-1} + B_j \boldsymbol{\nu}_{k-1} \right) \quad \boldsymbol{\nu}_{k-1} \sim N(0,1) \tag{3.3}$$

is modeled as a colored noise process (Section 3.2.3), where $v_0$ is the nominal vertical airspeed of the parafoil, $\Delta t$ is the discretization time step, $A_j$ and $B_j$ are the tuned parameters of the colored noise model for the $j^{\text{th}}$ wind classification, and $N(0,1)$ is the zero-mean, unit-variance Gaussian noise. A colored noise process was chosen to reflect the idea that, while wind at lower altitudes is correlated with the wind measured at the current altitude, the further that separation becomes, the lower the correlation.

### 3.2.2 Wind Model Classifications

Human jumpers utilize classification when planning and executing jumps. They utilize weather forecasting to determine jump windows or if a jump is feasible at all. Additionally, professional jumpers develop intuition about the effects of wind during a jump. In order to give the planner a methodology for tailoring the amount of conservatism to the observed wind condition, we have implemented an on-line learning algorithm to determine, in real time, the class of wind scenario being experienced by the parafoil. Such a class determines the parameters of the variational estimate (Section 3.2.3), ultimately guiding the actions of the planner, permitting higher risk-reward maneuvers or encouraging safe, reliable trajectories when the situation warrants it. The variational model associated with each class is tuned to capture the amount of uncertainty associated with the wind profiles within the class, while not incorporating unnecessary conservatism.

### Feature Selection

Wind profiles of the form $\{w_x(z), w_y(z), w_z(z)\}$ are data representations which pose significant problems for clustering and classification algorithms. Clustering and classification algorithms were designed to operate on observations, whereas wind profiles represent functions of altitude. The process of feature selection reduces the dimension of system models, and allows for the use of many efficient clustering algorithms and classification schemes [47]. For this work, the following features were chosen to represent a wind profile,

$$\Phi = \begin{bmatrix} \overline{\rho} & \max \rho & \overline{|\rho'|} & \max |\rho'| & \overline{|\theta'|} & \max |\theta'| \end{bmatrix}, \tag{3.4}$$

where $\rho = \sqrt{w_x(z)^2 + w_y(z)^2 + w_z(z)^2}$, $\theta = \text{atan2}(y, x)$, the overline notation denotes the average over all discrete data points in the wind profile, the max notation denotes the maximum over all discrete data points in the wind profile, and the derivative with respect to altitude $z$, denoted $(\cdot)'$, is computed point-wise forward and backward on the data by $x'_k = \dfrac{\mathrm{d}x_k}{\mathrm{d}z} = \dfrac{x_{k+1} - x_{k-1}}{z_{k+1} - z_{k-1}}$. These features represent the amount of power believed to be in the profile, captured by the $\overline{\rho}$ and $\max \rho$ terms, as well as the possible rate of influx of power, captured by the $\overline{|\rho'|}$ and $\max |\rho'|$ terms. The features $\overline{|\theta'|}$ and $\max |\theta'|$ represent the average and maximum horizontal plane direction shift experienced by a descending parafoil over the course of the wind profile.

### Clustering of Wind Profiles

After defining the relevant features, the problem of clustering the wind profiles becomes a standard clustering problem of partitioning observations $\{x_1, x_2, \ldots, x_n\}$, where the observations are the features described in the above Section for each of the $n$ recorded wind profiles, into $\kappa < n$ clusters $\mathbf{S} = \{S_1, S_2, \ldots, S_\kappa\}$. The partition is chosen so as to minimize the squared sum of the distance from the mean within each cluster, $\mu_i$,

$$\underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2. \tag{3.5}$$

The optimization problem posed in (3.5) has been extensively addressed in the literature [48, 49]. The modern, efficient, algorithm, known as the *k-mean* clustering algorithm, was proposed in Ref. [50].

The *k-means* algorithm is divided into two steps, an assignment step and an update step. The assignment step assigns the observation to the $S_l$ cluster by solving the optimization,

$$l = \underset{i}{\operatorname{argmin}} ||x_j - \mu_i||^2, \tag{3.6}$$

where $\mu_i$ are, again, the cluster centers. Once all observations have been assigned one of the $\kappa$ clusters, the cluster centers are re-computed using

$$\mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j. \tag{3.7}$$

The algorithm repeats the update and assignment steps until convergence.

The *k-means* clustering algorithm described above has been used in many machine learning and autonomy applications. The main drawback of the standard *k-means* approach is that it requires either (1) *a priori* knowledge of the number of clusters $\kappa$, or (2) an arbitrary amount of time/resources to determine the appropriate number of clusters through successive guessing of the value of $\kappa$ [51].

When approaching such a problem, it is common to augment the optimization (3.5) with a regularization term,

$$\underset{\mathbf{S}}{\operatorname{argmin}} \left( \sum_{i=1}^{\kappa} \sum_{x_j \in S_i} ||x_j - \mu_i||^2 \right) + \lambda\kappa. \tag{3.8}$$

Much like the *k-means* algorithm addresses the optimization in (3.5), the *DP-means* algorithm was developed to address the optimization posed in (3.8) [52]. *DP-means* addressed the major issues with the standard *k-means* approach by (1) assuming no *a priori* knowledge of the number of clusters and (2) determining the number of clusters incrementally, resulting in less computation.

In execution, *DP-means* is extremely similar to *k-means.* Both algorithms are divided into an assignment step and an update step. The key insight in *DP-means* is that during the assignment step, if an observation is further than $\lambda$ from the nearest cluster center, a new cluster is added, with the center defined as the observation which created it.

Draper Laboratories has released 194 altitude dependent wind profiles [46] collected using the sensor configuration and estimation procedure outlined in Ref. [2]. Because the wind is not directly measured on-board, it must be estimated, which is done using a standard Kalman filtering approach. Applying the DP-means algorithm to this wind profile set, resulted in classification into 3 distinct classes. The three classifications represent successively more conservative models for the evolution of the wind distribution. Ultimately, due to the limited parameters of the model, the only dimension along which two classes can be compare is which represents a more conservative approach. Class 1 represents the most optimistic representation, assuming little to no unknown variation in wind, scaling up to class 3 representing a significant uncertainty in the effect of wind on the vehicle.

### 3.2.3 Wind Model Components

**Persistent Wind Estimate**

This thesis presents a finite impulse response filter approach to estimating the persistent wind component of the model, $\overline{w}$. The finite impulse response filter is a simple, well understood, and reliable method for smoothing data. Particularly, it is used in practice to highlight long-term trends. Filters of this type mitigate the fluctuation in the long-term estimate caused by noise in the short term measurement. Such filters are discussed extensively in Ref. [53]. Such a filter is expressed as a discrete time convolution,

$$y[n] = \sum_{i=0}^{m-1} \frac{b_i}{a_0} x[n-i]. \tag{3.9}$$

This formulation has 3 parameters which must be identified. The weighting parameters $b_i$ and the scaling parameter $a_0$ are set to 1 and $m$, respectively. This reflects the belief that each observed sample (in this case, wind measurement) contains the same amount of information about the long-term trend (prevailing wind). The final parameter to determine is $m$, the width of the filter window. In this case, this parameter has been chosen to provide the planner with the most useful information, regardless of the wind profile being planned against, i.e. the value of $m$ which produces the most accurate predicted landing location.

In order to determine the effect of a wind profile on the parafoil, we consider the wind profile's effect on the landing location. In order to perform this analysis, and to ensure that the measured effect is due to the wind, we make the following assumptions:

1. Zero heading rate command, $\dot{\psi}_d = 0$
2. Zero initialized lag dynamics, $\mathbf{r}(0) = \mathbf{0}$
3. Flat terrain scenario, $T(x, y) = 0$

Based on these assumptions, we can ignore the lag dynamics, and the parafoil heading remains fixed. We can, therefore, compute the landing location $(x(t_f), y(t_f))$ of the parafoil by propagating the location dynamics (1.10) for

$$t_f = \left\{ t \;\middle|\; 0 = z_0 + \int_{t_0}^{t} \dot{z}(z, w_z(z))\mathrm{d}\tau = z_0 + \int_{t_0}^{t} \left( \frac{-v(z)}{L_D} + w_z(z) \right) \mathrm{d}\tau \right\},$$

under the influence of the true wind profile $w_x(z) = w_x^*(z),\ w_y(z) = w_y^*(z),\ w_z(z) = w_z^*(z)$

$$\begin{aligned} x(t_f) &= x_0 + \int_{t_0}^{t_f} \left( v(z)\cos\psi + w_x(z) \right) \mathrm{d}\tau, \\ y(t_f) &= y_0 + \int_{t_0}^{t_f} \left( v(z)\sin\psi + w_y(z) \right) \mathrm{d}\tau. \end{aligned} \tag{3.10}$$

Additionally, propagating these dynamics requires a propagation of the $z$ position (not listed) in order to evaluate $v(z)$, $w_x(z)$ and $w_y(z)$. For each wind profile, we determine the effect of the wind profile by computing the change in landing location between the parafoil under the influence of wind, $(x(t_f), y(t_f))$, and the parafoil under the influence of no wind, $\big(x'(t'_f), y'(t'_f)\big)$, i.e. $w_x(z) = 0,\ w_y(z) = 0,\ w_z(z) = 0$. This change is computed as

$$\Delta_w = \sqrt{\big(x(t_f) - x'(t'_f)\big)^2 + \big(y(t_f) - y'(t'_f)\big)^2}. \tag{3.11}$$

For each filter width $m$, we can characterize the effect of the mean wind on landing location prediction for each profile by computing a quantity, $\Delta_{w,m}$. Just as $\Delta_w$ represents the error between the true landing location and the zero-wind predicted location, $\Delta_{w,m}$ represents the error between the true landing location and the location predicted by the mean wind model with a filter length of $m$. Similar to (3.11), the error between true landing location and the mean wind predicted location is computed as

$$\Delta_{w,m} = \sqrt{\big(x(t_f) - x'_m(t'_f)\big)^2 + \big(y(t_f) - y'_m(t'_f)\big)^2}, \tag{3.12}$$

where $\big(x'_m(t'_f), y'_m(t'_f)\big)$ are computed using (3.10), assuming that $w_x(z) = \overline{w}_x,\ w_y(z) = \overline{w}_y,\ w_z(z) = \overline{w}_z$. This produces a quantity $\Delta_{w,m}$ for each wind profile $w$ and filter width. We characterize the effect that utilizing the mean wind estimate has on the predicted miss distance by comparing the effect of mean wind on landing location with the effect of the true wind profile using,

$$\delta d_{w,m} = \Delta_w - \Delta_{w,m}. \tag{3.13}$$

This comparison indicates that for some profiles and some filter widths, introducing a mean wind improves predictive ability by decreasing the predictive error, $\delta d_{w,m} > 0$, and for others the including a mean wind increases the predictive error, $\delta d_{w,m} < 0$. In order to choose the appropriate filter width, we wish to minimize the negative impact on predictive ability. Therefore, for each filter width we consider only the set of profiles which are adversely impacted by the incorporation of a mean wind prediction, $D_m = \{\delta d_{w,m} \mid \delta d_{w,m} < 0\}$.

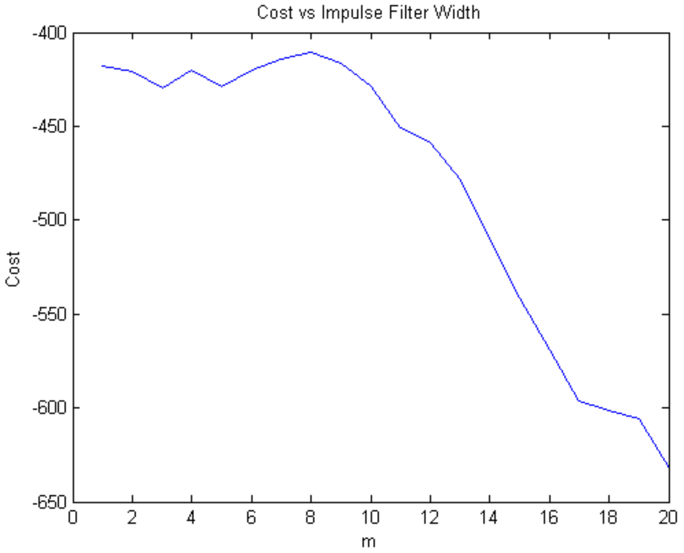


Figure 3-1: Regularized Cost for Filter Width Optimization. High values of the cost represent *limited* negative impact, and are therefore desire-able.

We define the cost related to each filter width as,

$$c_m = \min\left(D_m\right) + \beta\overline{D}_m - \lambda m, \tag{3.14}$$

where $\min\left(D_m\right)$ represents the minimum element of the set of adversely impacted profile predictions, i.e. the prediction where the accuracy decreases by the largest amount, $\overline{D}_m$ represents the average decrease in accuracy, $\beta > 0$ is a weighting term, and $\lambda > 0$ is the regularization term. Maximizing this cost results in finding the filter width with the least negative impact on predictive ability. Figure 3-1 shows the regularized cost $c_m$ versus the filter width $m$ using the Draper wind profiles to compute the wind effects. Considering the construction of the set $D_m$, desireable values of the cost, $c_m$, are high values, i.e. the optimal vlaue of $m$ satisfies $m = \underset{m}{\text{argmax}}\ c_m$. The general trend of this figure is a that of a plateau of relatively similar values, followed by a steep drop-off. Given that the parameters $\beta$ and $\lambda$ can be chosen, any filter width value between $[0, 10]$ is appropriate. For this work, we have chosen $\beta = 2$ and $\lambda = 1$, along with a filter width of $m = 8$.

### Variational Wind Model

Each of the wind classes determined in Section 3.2.2 requires a separately trained variational wind model, $\delta\mathbf{w}$, i.e. requires separately tuned matrices $A_j$ & $B_j$. Each of these models are trained based on the effect of the variational wind component on the predicted landing location. Each of the variational wind models is constructed using the following process:

**Propagate.** For each of the wind profiles within the cluster of interest, construct the quantity $\Delta_{w,m}$, described above, for the optimal filter width.

**CDF Construction.** From the set of distance measures above, construct a CDF miss distance as described in Section 2.2.2.

**Analytic CDF Construction.** The variational wind is modeled as a multi-modal linear system subject to Gaussian noise. Using a state vector

$$\delta\mathbf{w} = \begin{bmatrix}\delta w_x & \delta w_y & \delta w_z\end{bmatrix}^T, \tag{3.15}$$

where $(\delta w_x, \delta w_y, \delta w_z)$ are variational winds in the $x, y, z$ direction. The system in (3.3) for class $j$, is expressed as a continuous time system by

$$\delta\dot{\mathbf{w}} = A_j\delta\mathbf{w} + B_j\boldsymbol{\nu}. \tag{3.16}$$

We assume that $\delta w_x$ and $\delta w_y$ are independent and symmetric. Additionally, we assume the model to be two-dimensional, i.e. $\delta w_z = 0\ \forall t$. Alternatively, these assumptions can be written as conditions on the matrices $A_j$ & $B_j$ by

$$A_j = \alpha_j\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix},\quad B_j = \beta_j\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}, \tag{3.17}$$

where $\beta_c > 0$ is a positive scalar.

Consider the variation of position states $(x, y, z)$ relative to the expected states $(E[x], E[y], E[z])$. For illustration, the expected state $E[x]$ is developed by

$$\begin{aligned}
x &= x_0 + \int_{t_0}^{t_f} \dot{x}(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau))\mathrm{d}\tau, \\
x &= x_0 + \int_{t_0}^{t_f} v(z)\cos\psi + \overline{w}_x + \delta w_x \mathrm{d}\tau, \\
E[x] &= x_0 + \int_{t_0}^{t_f} v(z)\cos\psi + \overline{w}\mathrm{d}\tau, \\
x - E[x] &= \int_{t_0}^{t_f} \delta w_x \mathrm{d}\tau, \\
\delta\dot{x} &= \delta w_x. \qquad (3.18)
\end{aligned}$$

Collecting the variational states,

$$\delta\mathbf{x} = \begin{bmatrix} x - E[x] \\ y - E[y] \\ z - E[z] \end{bmatrix},$$

which are governed by the dynamics

$$\delta\mathbf{x} = \delta\mathbf{w}.$$

Creating a system of augmented dynamics, using the states $\delta\mathbf{x}$ and $\delta\mathbf{w}$,

$$\begin{bmatrix} \dot{\delta\mathbf{x}} \\ \dot{\delta\mathbf{w}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_3 & I_3 \\ \mathbf{0}_3 & A_j \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{w} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 2} \\ B_j \end{bmatrix}}_{\mathbb{B}} \boldsymbol{\nu}, \qquad (3.19)$$

where $\mathbf{0}_3$ represents a 3 × 3 matrix of 0, $\mathbf{0}_{3\times 2}$ represents a 3 × 2 matrix of 0, and where $I_3$ is the 3 × 3 identity matrix. The covariance of the augmented state, $\Sigma$, can be computed by solving

$$\dot{\Sigma} = \mathbb{A}\Sigma + \Sigma\mathbb{A}^T + \mathbb{B}\mathbb{B}^T, \tag{3.20}$$

with initial covariance

$$\Sigma(t_0) = \mathbf{0}_6,$$

since all of this analysis is performed off-line and there is no error in the variational wind or position estimates.

In order to compare the analytic covariance at impact with the true CDF, the lateral position elements must be isolated from the covariance. This isolation is accomplished by defining the transformation matrix $C_T = [I_2 \; \mathbf{0}_{2\times 4}]$. The covariance of interest, $\Sigma'$, is computed by $\Sigma' = C_T \Sigma C_T^T$. Given the independence and symmetry assumptions of $\delta w_x$ and $\delta w_y$, $\Sigma'$ will result in a diagonal $2 \times 2$ matrix with identical, positive diagonal elements, and can therefore be expressed as $\Sigma' = \sigma^2 I_2$, where $\sigma > 0$ is a scalar. Utilizing the standard deviation, $\sigma$, the PDF of a $\chi$-distribution with degree of freedom of 2 is constructed by

$$\chi(x) = \left(\frac{x}{\sigma\Gamma(1)}\right) \exp\left\{-\left(\frac{x}{\sigma\sqrt{2}}\right)^2\right\}, \tag{3.21}$$

where $\Gamma(z)$ is the Gamma Function. Denote the CDF associated with this distribution by $\chi_C(x)$.

**Tuning.** Tuning of the parameters of the matrices $A_j$ & $B_j$ must be done on a per class basis. The true CDF of each class can be expressed as a series of points $(d_i, n_i)$ where $d_i$ represents the $i^{\text{th}}$ largest miss distance and $n_i$ represents the fraction of profiles resulting in a miss less than or equal to $d_i$. The minimization was implemented in MATLAB using `fminunc` to determine the parameters of $A$ and $B$ by addressing the root mean square of the error between the actual CDF and the analytically derive CDF by

$$J = \sum_{i \in C} (n_i - \chi_C(d_i))^2, \tag{3.22}$$

where $C$ represents the cluster of interests.

Figures 3-2(a), 3-2(b) and 3-2(c) show the true CDF (blue) for the wind class and the tuned CDF (red). Class 1 represents an optimistic view of the future effect of wind, where the wind is believed to have low power and/or variability. Class 2 represents a class of moderate effect. Winds in Class 2 have more of an effect than those in Class 1. Class 3 represents a pessimistic view of the effect of the wind. The winds in Class 3 are believed to have significant power and/or variability.

## 3.3 Classification

In order to utilize the varying levels of uncertainty associated with the $k$ classifications determined by *DP-means* in Section 3.2.2, the planner must have a methodology for using the observed wind estimates to assign the wind that is being experienced by the vehicle to a classification. This is known as statistical classification, and is a common machine learning problem.

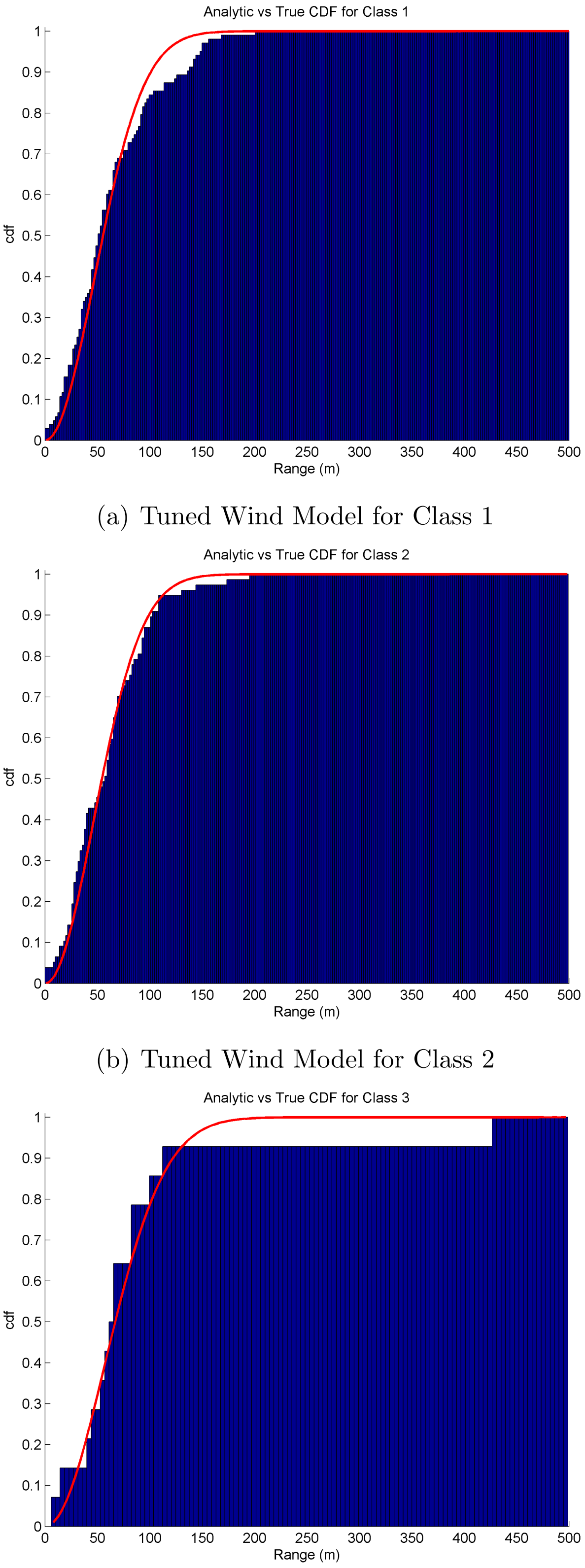
### 3.3.1 Support Vector Machine

The supervised learning problem is the task of inferring a classification function $y = f(x)$ from labeled training data, $P = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{M}$, [47]. Support vector machines (SVM) are a particularly well known method for solving the supervised learning problem. A SVM results from solving the following optimization

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.} \quad & y_i(\mathbf{w} \cdot x_i - b) \geq 1, \end{aligned} \tag{3.23}$$

for the separating hyperplane with normal vector $\mathbf{w}$ and offset $b$. The optimization (3.23) can be expressed with Lagrange multipliers $\boldsymbol{\alpha}$ as

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha} \geq 0} \left\{ \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{M} \alpha_i [y_i(\mathbf{w} \cdot x_i - b) - 1] \right\}. \tag{3.24}$$



(a) Tuned Wind Model for Class 1

(b) Tuned Wind Model for Class 2

(c) Tuned Wind Model for Class 3

Figure 3-2: Tuned Wind Models for Classes 1-3. The blue region corresponds to the data collected from the true wind profiles, while the red lines correspond to the tuned wind models.

The resulting optimization arguments $\mathbf{w}$ and $b$, as well as the Lagrange multipliers $\alpha_i$, fully determine the approximation function $f(x)$.

Relating this scheme with the classes generated in Section 3.2.2, a SVM can be generated for each of the $\kappa$ classes identified by the *DP-means* algorithm, and identify if the wind experienced by the vehicle is a member of a particular class. This process generates $\kappa$ binary inclusion classifiers, indicating whether the observed wind is within the class $\kappa$. It is possible for multiple classifiers to result in affirmative classifications. If this is the case, the algorithm chooses the class which is the most conservative, i.e. has the fastest growing uncertainty.

The decision to choose the most conservative class derived from the need to resolve the problem of multiple affirmative classifications. Since previous analysis has shown there to be only three classes of wind model, a series of one-versus-all classifiers, such as those derived above, is sufficient to accurately and efficiently solve the problem. However, if future data reveals the number of classes to be 5 or greater [54], then it is advised to consider Error-Correcting Output Codes (ECOC) [55]. ECOC are an efficient method for solving small- to medium- sized multi-classification problems, which are proven to have both low bias, as well as low variance in classification [56].

## 3.4 Results and Comparison

### 3.4.1 Miss-Classification Effects

In order to validate the modeling and classification approach described in this chapter, the effect of miss-classification on system performance is considered. The models and classification scheme are highly coupled with the analytic robustness approach presented in Chapter 4, and it is within that framework that the following test was performed.

In order to ascertain the effect of the combined modeling and classification system, a test using forced miss-classification is performed. In this test the algorithm was artificially forced to classify all wind profiles into a single classification. The effect of

Table 3.1: Miss Distance Data for Miss-Classification Test

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| Combined | 500 | 30.8 | 32.8 | 18.7 | 52.5 | 75.8 | 103 | 126 | 218 |
| Class 1 | 500 | 32.0 | 39.7 | 18.8 | 50.4 | 74.8 | 102 | 134 | 349 |
| Class 2 | 500 | 35.4 | 38.6 | 21.2 | 61.1 | 86.2 | 113 | 138 | 284 |
| Class 3 | 500 | 41.2 | 41.2 | 26.1 | 70.6 | 109 | 127 | 157 | 218 |

utilizing each class separately is compared to utilizing them in a combined manner. Figure 3-3 depicts this data graphically using a CDF, while Table 3.1 reproduces this data in tabular form.

Let us consider each class individually. Class 1 represents the optimistic class, where the wind is believed to have low power and/or variability. When all profiles are assumed to be of this class, we would expect that the planning algorithm would take considerable risks, some of which would pay off with low miss distances, whereas others would result in significantly higher worst case scenarios. This phenomenon can be seen by the worst case scenario for the labeled Class 1 demonstrating an increase of nearly 60% in the worst cases, but having a mean which is comparable to the nominal combined case (an improved mean of 29.4m, if one ignores the worst case scenarios). Class 2 is a moderate class in which the wind is believe to have more of an effect relative to Class 1. The understanding of increased wind effect is demonstrated in the data by an increase in the mean miss distance, but a decrease in the worst case scenarios, as the planner takes fewer risks and plans more conservative paths. Class 3 represents the class in which the wind is believed to have the most significant effect on the parafoil. When this class is assumed, the planner is extremely conservative, suffering significantly in average performance, but is able to maintain the low worst case performance. The combined approach, using the classification method described in Section 3.3, utilizes the strengths of each of these classes in order to plan aggressive paths when the wind is believed to have little effect, and maintain a conservative approach when the wind appears to have a greater effect.

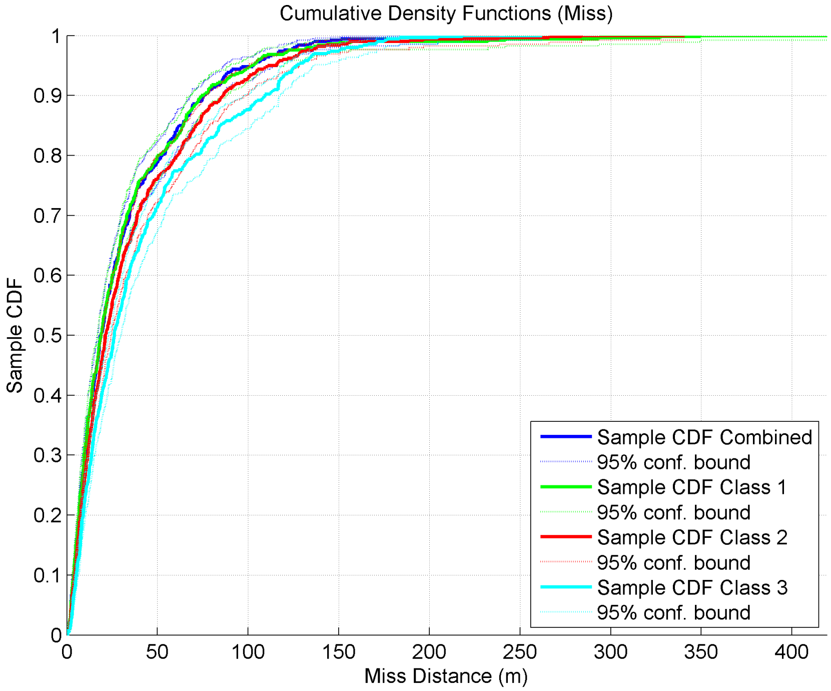


Figure 3-3: Miss Distance CDF for Miss-Classification Test

## 3.5 Summary

This chapter developed a multi-class colored noise wind modelling scheme. The model is broken down into the persistent and the variational components. The persistent component is modelled as a finite impulse response filter, and is assumed to be present throughout the course of the parafoil mission. The variational component of the model incorporates multiple wind classifications determined by the *DP-means* algorithm. The variational wind model is analytically tuned to match parafoil drop data collected by Draper Laboratories Ref. [46]. Additionally, the classification scheme is able to classify the observed wind online and dynamically adjust the level conservatism of the planner (represented by changing the wind class and therefore the variational model for future wind effects). Finally, this chapter considers the effect of miss-classification on the performance of the parafoil planner. This consideration demonstrates the effectiveness of using a multi-class scheme in practice.

# Chapter 4

# Analytic Chance Constraints

## 4.1 Introduction

This chapter will first demonstrate the limitations of replanning as a method for handling environment uncertainty, as well as the limitations of the particle CC-RRT approach to approximating the uncertainty distribution. The core contribution of this chapter is the presentation of the analytic sampling approach. The analytic uncertainty distribution based on the wind model given in Chapter 3 is derived, as well as detail the procedure for generating the analytic samples. Finally, this chapter will demonstrate the computational advantages of the approach. Chapter 5 will demonstrate and discuss the results of this approach in simulation.

### 4.1.1 Algorithm Naming Convention

This thesis compares 3 specific variants of the CC-RRT algorithm. The naming conventions for the variants are listed below:

**Mode A.** Mode A represents a nominal RRT planner, incorporating the effects of mean wind as described in Section 3.2.1. This approach makes no active attempt at robustness against uncertainty, but does utilize replanning at every time step to attempt to counter-act system disturbances.

**Mode AB.** Mode AB is the CC-RRT particle representation proposed in Ref. [57].

This mode incorporates mean wind as in Section 3.2.1, as well as propagates 10 particles (the effective computational limit), sampled from the uncertainty model developed in Chapter 3, in order to approximate the uncertainty distribution to attempt to incorporate robustness to wind effects.

**Analytic.** CC-RRT with Analytic Sampling is the full CC-RRT approach developed in this thesis. It utilizes the wind model from Chapter 3 to inform the choice of analytic covariance samples as discussed in this chapter in order to incorporate robustness against wind uncertainty and terrain collision.

## 4.2 Motivation

### 4.2.1 Results of RRT with Replan vs BLG

Algorithms such as those proposed in Refs. [2, 20] utilize path replanning at every control cycle coupled with prevailing (mean) wind prediction. It is natural to consider the algorithm as proposed thus-far, RRT with replanning, to account for prevailing wind (RRT Mode A). Figure 4-1 and Table 4.1 compare the RRT Mode A algorithm with the state-of-the-art BLG approach (Section 1.3.6), executing on the valley terrain shown in Section 1.3.5. The CDF of the RRT Mode A algorithm demonstrates significant improvement over the state of the art BLG. Based on the criteria set forth in Section 2.2.2, CC-RRT Mode A outperforms BLG in mean, all percentile levels as well as in CDF curve shape. This improvement indicates that under nominal conditions, CC-RRT Mode A represents an improved approach to mitigating and reacting to the effects of the uncertain wind environment.

However, both BLG as well as CC-RRT Mode A exhibit off-nominal cases with unacceptable worst case performance. Such situations are the product of an interaction between the uncertain wind and the difficult terrain encountered by the parafoil. Figures 4-2(a) and 4-2(b) demonstrate step 1 of the vehicle trajectory (blue) as well as the planned path (green) on the Valley Terrain (Section 1.3.5), viewed from the side and skewed top, respectively. From these figures, we can see that the CC-RRT

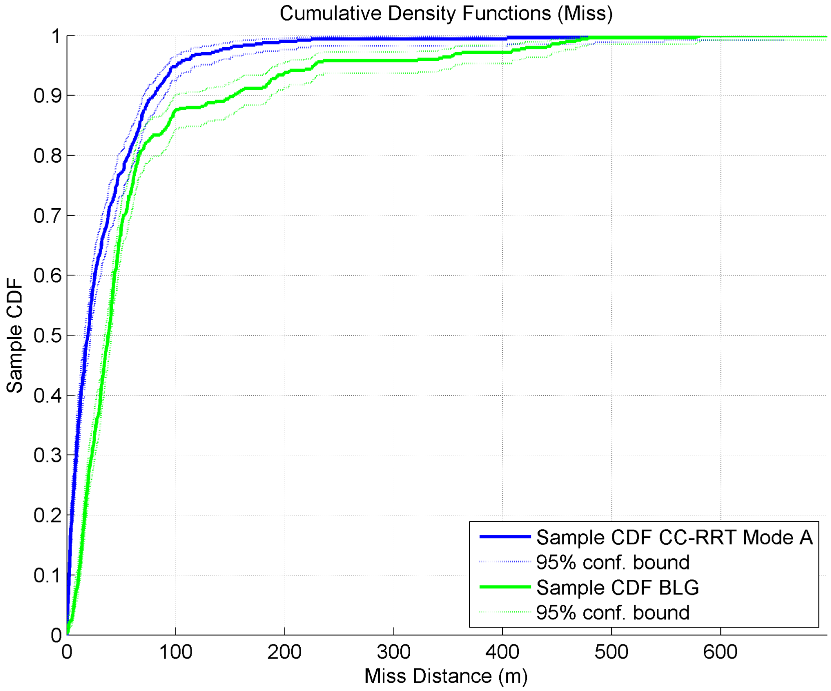
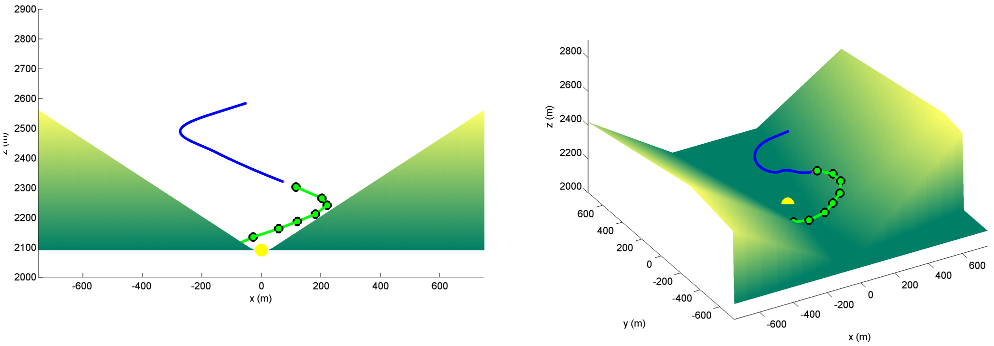


Figure 4-1: Miss Distance CDF: RRT with Replanning vs BLG

Table 4.1: Miss Distance Table: RRT with Replanning vs BLG

| **Name** | **N** | **Mean** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| CC-RRT Mode A | 500 | 33.9 | 49.3 | 19.3 | 56.6 | 79.7 | 101 | 151 | 548 |
| BLG | 500 | 63.5 | 89.0 | 37.9 | 66.1 | 153 | 227 | 431 | 581 |

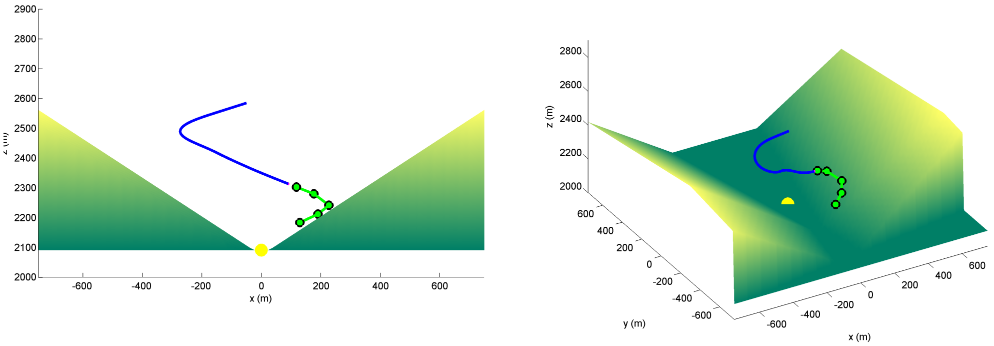
Mode A planner has selected a path which, nominally, produces an accurate landing. It is important to note, however, that the path planned is very close to the terrain. In step 2, Figures 4-3(a) and 4-3(b), we see that the wind has shifted, causing the previously planned path, and all other options as well, to collide with the terrain. It is this behavior which the addition of robustness must address.



(a) Side View

(b) Skewed Top View

Figure 4-2: CC-RRT Mode A Worst Case Step 1. The nominal CC-RRT Mode A planner generated path, showing an accurate simulated landing; the predicted path is in green, while the goal is the yellow circle.



(a) Side View

(b) Skewed Top View

Figure 4-3: CC-RRT Mode A Worst Case Step 2. The nominal CC-RRT Mode A planner generated path from step 1 acted on by unexpected wind, showing a terrain collision; the predicted path is in green, while the goal is the yellow circle.

### 4.2.2 Deficiency of CC-RRT with Particles

In selecting an RRT approach to solving the parfoil terminal guidance problem, there are two important considerations: (1) nonlinear vehicle dynamics and (2) potentially nonlinear/non-Gaussian wind model propagated through the nonlinear vehicle dynamics. Ref. [57] proposes a particle based CC-RRT approach for a problem with a nonlinear vehicle subject to nonlinear/non-Gaussian disturbances. The central idea to the algorithm is the use of particles, randomly sampled instances of the disturbances propagated through the dynamics, to generate a statistical representation of the uncertainty; a representation which approaches truth and allows for guarantees on probabilistic feasibility as the number of particles goes to infinity [57, 58].

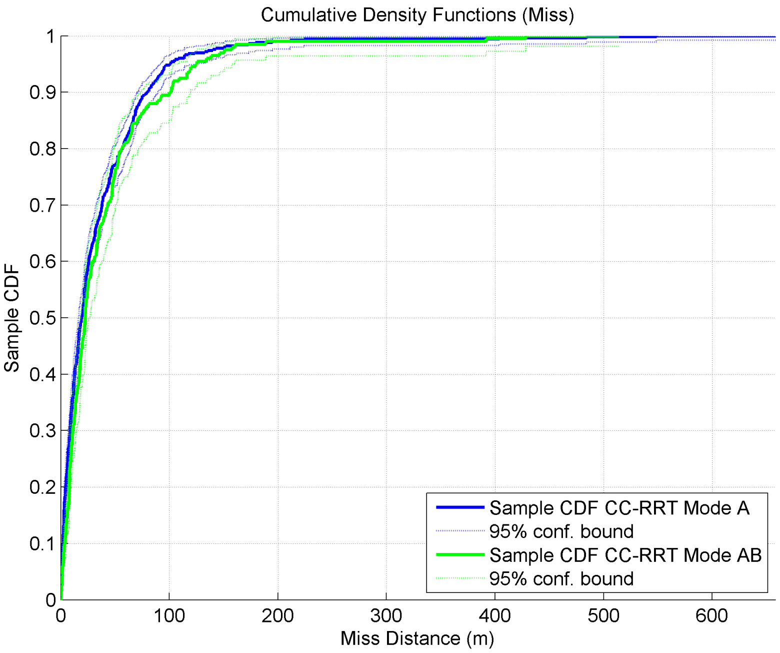


Figure 4-4: Miss Distance CDF of Mode A vs Mode AB

Ref. [59] applies the particle CC-RRT (CC-RRT Mode AB) approach to the parafoil problem, and considers the trade-off between the number of simulated particles and the size of the tree. Figure 4-4 shows no significant change in the performance of the RRT algorithm when incorporating the particle formulation. Moreover, Figure 4-5 shows no significant change in the worst case scenario. The CC-RRT Mode AB algorithm applied to the parafoil terminal guidance problem suffers from 2 significant drawbacks. (1) The particle representation used in the CC-RRT Mode AB algorithm cannot guarantee the detection of a terrain collision. Table 4.2 shows the worst case miss distance for CC-RRT Mode AB exceeds 425m. A miss distance of this magnitude comes from unintended terrain collisions, implying that the particles failed to appropriately cover the uncertainty distribution. (2) Particles incur a extremely high computational cost, resulting in sparse trees and fewer planning options [59].

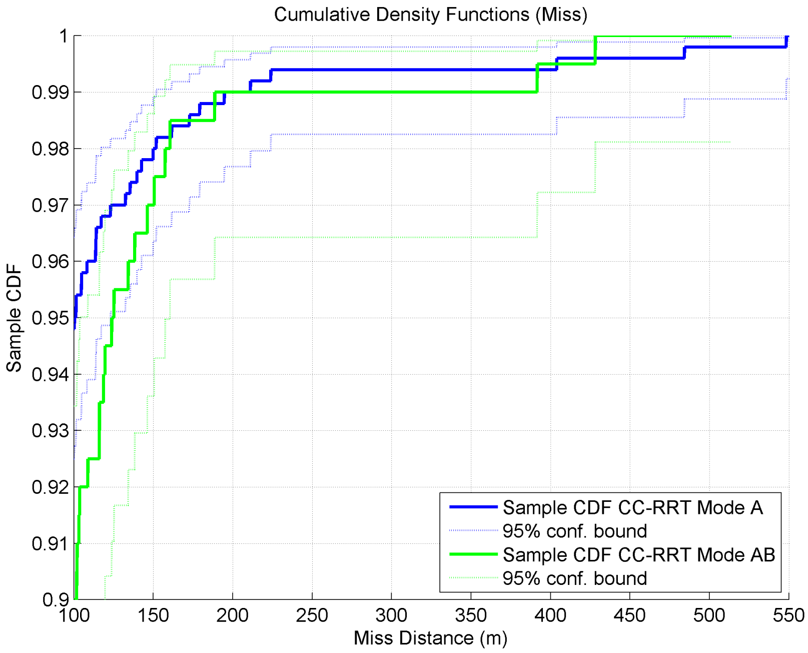


Figure 4-5: Zoomed Miss Distance CDF of Mode A vs Mode AB

Table 4.2: Miss Distance Data of Mode A vs Mode AB

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| CC-RRT Mode A | 500 | 33.9 | 49.3 | 19.3 | 56.6 | 79.7 | 101 | 151 | 548 |
| CC-RRT Mode AB | 200 | 39.2 | 52.6 | 22.3 | 56.4 | 101 | 125 | 159 | 428 |

## 4.3 Analytic Covariance Sampling

Due to the high computation time and poor coverage of the possible state distribution provided by the particle CC-RRT approach, we require a method which is able to cover the state distribution in a computationally efficient manner.

Ref. [1] leverages the ability to form analytic representations of the uncertainty distribution in the initial formulation of CC-RRT. However, this initial formulation considers only polyhedral constraints, and does not address the arbitrary terrain, $h = T(x, y)$, faced in the parafoil problem. In order to efficiently check the terrain constraint, the likelihood of collision with terrain is approximated by generating equispaced samples at specified levels of the uncertainty distribution at each prospective trajectory node. We utilize this method, a user-defined safety threshold, $p_{\text{safe}}$, as well as a user defined discretization level $M$, which allows for tunable levels of robustness. This is accomplished by utilizing the wind model developed in Chapter 3 by,

1. Demonstrating that the wind model results in a linear effect on the state propagation,
2. Deriving the analytic uncertainty distribution caused by the linear wind model,
3. Directly sampling the uncertainty distribution to generate the distribution approximation.

Sampling the distribution in this way allows for coverage of the uncertainty space with relatively few samples, as well removing the need to dynamically propagate each sample, significantly reducing computation time, and retain the computation benefits associated with terrain checking.

### 4.3.1 Deriving the Distribution

**Linear Wind Effect**

Consider the $x$ and $y$ states of the vehicle model (1.11), incorporating the wind model described in Chapter 3,

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \left( v(z_k) \cos \psi_k + \overline{w}_{x,k} + \delta w_{x,k} \right), \\ y_{k+1} &= y_k + \Delta t \left( v(z_k) \sin \psi_k + \overline{w}_{y,k} + \delta w_{y,k} \right). \end{aligned} \tag{4.1}$$

The variation $\delta x = x - E[x]$ and $\delta y = y - E[y]$ of the states (4.1) about the mean can be expressed as,

$$\begin{aligned} \delta x_{k+1} &= x_{k+1} - E[x_{k+1}] = \delta x_k + \Delta t \left( \delta w_{x,k} \right), \\ \delta y_{k+1} &= y_{k+1} - E[y_{k+1}] = \delta y_k + \Delta t \left( \delta w_{y,k} \right). \end{aligned} \tag{4.2}$$

This system is linear in the variational wind components. Moreover, if augmented with the wind model for the variational components (3.3), the resultant system

$$\begin{aligned}
\delta x_{k+1} &= \delta x_k + \Delta t \left(\delta w_{x,k}\right), \\
\delta w_{x,k+1} &= \delta w_{x,k} + v_0 \Delta t \left(\alpha_x \delta w_{x,k} + \beta_x \nu_{x,k}\right), \\
\delta y_{k+1} &= \delta y_k + \Delta t \left(\delta w_{y,k}\right), \\
\delta w_{y,k+1} &= \delta w_{y,k} + v_0 \Delta t \left(\alpha_y \delta w_{y,k} + \beta_y \nu_{y,k}\right),
\end{aligned} \tag{4.3}$$

remains linear. Condensing the linear system (4.3) using the state vector $\delta \mathbf{x}_k = [\delta x_k \ \delta w_{x,k} \ \delta y_k \ \delta w_{y,k}]^T$, and input vector $\boldsymbol{\nu}_k = [\nu_{x,k} \ \nu_{y,k}]^T$ yields,

$$\delta \mathbf{x}_k = \mathbb{A} \mathbf{x}_{k-1} + \mathbb{B} \boldsymbol{\nu}_k \tag{4.4}$$

$$= \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 + v_0 \Delta t \alpha_x & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 + v_0 \Delta t \alpha_y \end{bmatrix} \delta \mathbf{x}_{k-1} + \begin{bmatrix} 0 & 0 \\ v_0 \Delta t \beta_x & 0 \\ 0 & 0 \\ 0 & v_0 \Delta t \beta_y \end{bmatrix} \boldsymbol{\nu}_k. \tag{4.5}$$

### Analytic Uncertainty Distribution

With the ultimate goal of developing the uncertainty distribution, we note that given the linear system (4.4) driven by the Gaussian noise $\boldsymbol{\nu}_k$, all future state distributions of $\delta \mathbf{x}_k$ remain Gaussian [60]. Based on the procedure in Ref. [60], the covariance matrix $P_k = E[\delta \mathbf{x}_k \delta \mathbf{x}_k^T]$ at an arbitrary time step $k$ is computed by the recursion,

$$P_k = \mathbb{A} P_{k-1} \mathbb{A}^T + \mathbb{B}\mathbb{B}^T, \tag{4.6}$$

or explicitly by,

$$P_k = \mathbb{A}^k P_0 (\mathbb{A}^T)^k + \sum_{t=0}^{k-1} \mathbb{A}^{k-t-1} \mathbb{B}\mathbb{B}^T (\mathbb{A}^T)^{k-t-1}. \tag{4.7}$$

As only the effect of the disturbances on the $x$ and $y$ states is considered, we define a transformation matrix $C_T$ to extract the relevant states from the state vector $\delta\mathbf{x}_k$ by (4.8),

$$\begin{aligned} \delta\mathbf{x}'_k &= \begin{bmatrix} \delta x_k & \delta y_k \end{bmatrix}^T = C_T \delta\mathbf{x}_k, && (4.8) \\ C_T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. && (4.9) \end{aligned}$$

Therefore, the covariance of the positions states, $Q_k$, can be expressed by,

$$Q_k = C_T P_k C_T^T. \tag{4.10}$$

### Generating Sample Locations

The problem of determining a series of equi-spaced samples can be considered as the geometric problem of finding a series of points along the uncertainty ellipse. The solution to this problem is broken down into two steps: (1) determine the location of of each point in the principle axis coordinate system, (2) transform the points into the nominal path relative frame.

The covariance matrix $Q_k$ describes a contour of equal probability of points $\mathbf{\Delta x}_k = [\Delta x_k \;\; \Delta y_k]^T$ relative to the nominally propagated trajectory by the conic relation $\mathbf{\Delta x}_k^T Q_k^{-1} \mathbf{\Delta x}_k = 1$. Additionally, denote the elements of $Q_k$ as

$$Q_k = \begin{bmatrix} \sigma_{x,k}^2 & \sigma_{xy,k} \\ \sigma_{xy,k} & \sigma_{y,k}^2 \end{bmatrix}. \tag{4.11}$$

Let $\sigma_a^2$ and $\sigma_b^2$ be the eigenvalues of $Q_k$, with $\sigma_a > \sigma_b$. These eigenvalues represent the semi-major and semi-minor axes of the uncertainty ellipse, which are oriented along the principle axes of the ellipse. The angle $\theta' = \dfrac{1}{2}\tan^{-1}\left(\dfrac{2\sigma_{xy,k}}{\sigma_{x,k}^2 - \sigma_{y,k}^2}\right)$ determines the rotation from the $x/y$ plane of the vehicle to the principle axis system of the ellipse, $x'/y'$. The $M$ equi-spaced samples are indexed by an angle with respect to the $x-$axis, so that the feasibility of each sample can be tracked from one time step to the next; denote this angle $\theta_j = \dfrac{2\pi}{M-1} j$.

In the $x'/y'$ system, the position of the $j^{\text{th}}$ sample point is determined by

$$\begin{aligned} R &= \frac{\sigma_a \sigma_b}{\sqrt{(\sigma_b \cos(\theta_j - \theta'))^2 + (\sigma_a \sin(\theta_j - \theta'))^2}}, & (4.12) \\ \Delta x'_{j,k} &= R \cos(\theta_j - \theta'), & (4.13) \\ \Delta y'_{j,k} &= R \sin(\theta_j - \theta'). & (4.14) \end{aligned}$$

The $j^{\text{th}}$ sample point $\mathbf{\Delta x}_{j,k}$ can be determined by a planar rotation,

$$\mathbf{\Delta x}_{j,k} = \sigma \begin{bmatrix} \cos\theta' & \sin\theta' \\ -\sin\theta' & \cos\theta' \end{bmatrix} \begin{bmatrix} \Delta x'_{j,k} \\ \Delta y'_{j,k} \end{bmatrix}, \tag{4.15}$$

where $\sigma$ denotes the covariance scale factor.

### Feasibility Determination

Given a set of uncertainty samples $\mathbf{\Delta x}_{j,k}$, derived above, feasibility is determined by determining the probability of terrain collision,

$$p_{\text{collide}} = \frac{1}{M} \sum_{j=0}^{M-1} \mathbb{I}\left(\overline{x}_k + \Delta x_{j,k}, \overline{y}_k + \Delta y_{j,k}, \overline{z}_k, T(x,y)\right), \tag{4.16}$$

where $\overline{x}_k$, $\overline{y}_k$, and $\overline{z}_k$ are the nominal trajectory points, $T(x,y)$ is the terrain map, and the function $\mathbb{I}(x,y,z,T)$ takes a trajectory point $(x,y,z)$ and a terrain map $T$ and returns 1 if the trajectory point intersects the terrain and 0 otherwise. If the probability of collision, $p_{\text{collide}}$, exceeds the user specified probability of safety $1-p_{\text{safe}}$, then the trajectory is considered to have landed.

In addition to the uncertainty based feasibility check, if the nominal trajectory point lands, the trajectory is considered landed. That is, if $\mathbb{I}(\overline{x}, \overline{y}, \overline{z}, T) = 1$, then the entire trajectory is considered landed. This landing assignment supersedes the probabilistic assignment discussed above, considering the trajectory as landed even in the case where $p_{\text{collide}} > 1 - p_{\text{safe}}$.

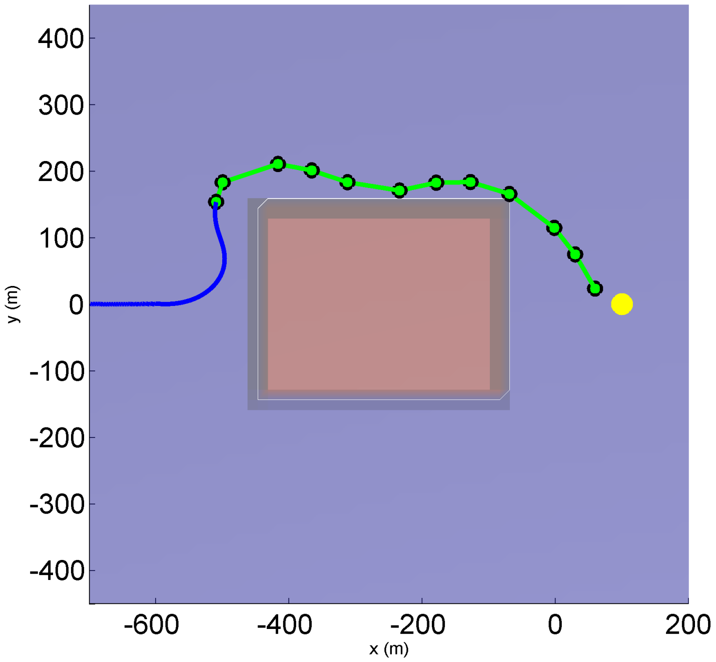


Figure 4-6: Nominal CC-RRT Mode A Plan Unacceptably Close to Obstacle Terrain

### 4.3.2 Parameters: Covariance Buffer

In order to determine the outermost level used for constraint checking, the covariance scale factor introduced in (4.15) must be determined. In order to determine this parameter, a test utilizing the Obstacle Terrain and scenario depicted in Section 1.3.5 is constructed. This test utilizes the near pathological nature of the scenario to incentivize paths planned arbitrarily close to the Obstacle Terrain, as shown in Figure 4-6. This test considered 100 trials of scale factors $\sigma \in [0, 2.5]$ for $p_{\text{safe}} \in \{0.6, 0.7, 0.8, 0.9\}$.

This test considered two variables, the user-defined probability of safety, $p_{\text{safe}}$, as well as the covariance scale factor $\sigma$. Shown in Figure 4-7 is the crash percentage (% of runs resulting in a collision with the obstacle) as a function of the pre-determined covariance scale factor, evaluated for a variety of $p_{\text{safe}}$ (probability of safety, or $1-$ crash percentage) selections. The horizontal lines represent the expected crash percentage for each of the $p_{\text{safe}}$ choices. The point at which the horizontal lines intersect with the appropriate crash percentage vs $\sigma$ curve represents the value of the covariance scale

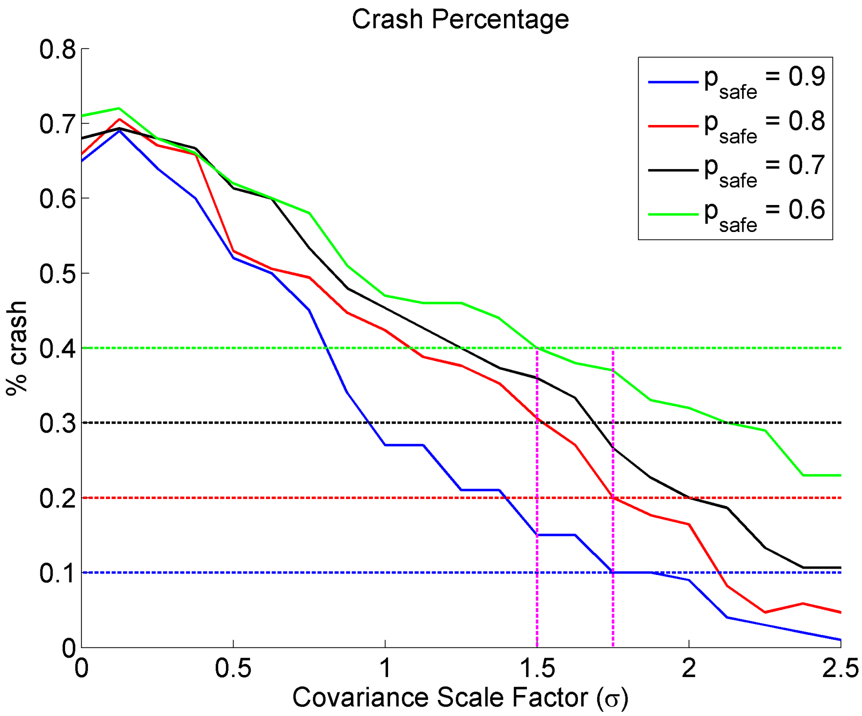


Figure 4-7: Crash Percentage at Standard Deviation Levels

factor which, in practice, results in satisfaction of the user-defined $p_{\text{safe}}$ constraint. The vertical magenta lines bracket the range of $\sigma = [1.5, 1.75]$ which satisfy the $p_{\text{safe}}$ constraint. Any value of $\sigma$ from this range is an appropriate choice for this work.

For this work a value of $\sigma = 1.75$ coupled with $p_{\text{safe}} = 0.9$ has been chosen. This value was chosen so that, in practice, the parafoil achieves a $p_{\text{safe}} = 0.9$. It is important to note that this choice of $\sigma$ does not guarantee $p_{\text{safe}} = 0.9$. Since the analytic covariance samples only cover a finite portion of the uncertainty space, they cannot be used to make robustness claims addressing regions of the uncertainty space beyond the outermost layer sampled.

## 4.4 Computational Comparison of CC-RRT with Analytic Sampling to CC-RRT Mode AB

Table 4.3 presents the trade-off between rate of tree node generation and discretization size of the uncertainty distribution. The discretization size of the uncertainty

Table 4.3: CC-RRT Mode AB and Analytic Sampling Node Generation Times

| | Average ms/Nodes | |
|---|---|---|
| Disc. Size | CC-RRT Mode AB | Analytic Sampling |
| 5 | 51.29 | 17.00 |
| 10 | 94.12 | 26.24 |
| 15 | 163.77 | 33.19 |
| 25 | 290.07 | 46.00 |
| 35 | 342.41 | 83.53 |
| 50 | 427.60 | 83.25 |

distribution is measured by the number of particles in the CC-RRT Mode AB formulation by the number of particles. The discretization size for CC-RRT with Analytic sampling is measured by the number of samples used. This table shows the average number of milliseconds required to generate a new tree node in the Java code for each specified level of discretization.

Table 4.3 clearly demonstrates the significant decrease in node generation time over all discretization sizes. This improvement implies that during real-time operation either a larger discretization of the space can be used, more nodes will be added to the tree during each growth cycle. This result is significant, particularly when put in the context that CC-RRT Mode A requires, on average, 10.4 ms to create a node. This implies that CC-RRT with Analytic Sampling can utilize $M = 10$ samples, the level of discretization used in practice, wile only decreasing tree size, at most, by a factor of 2.5, as opposed to a factor of 9.5 as demonstrated using CC-RRT Mode AB.

## 4.5 Summary

This chapter presented an analytic method for determining the state uncertainty distribution. This method uses uniformly spaced samples of the analytic uncertainty distribution, resulting in an efficient method for constraint checking. Moreover, this method has been shown to significantly decrease the computational impact of representing the uncertainty space.

The second major claim of the approach is an improved coverage of the uncertainty space for the parafoil problem relative to the particle chance constrained approach. Chapter 5 will demonstrate the results of the CC-RRT with Analytic Sampling algorithm in simulation relative CC-RRT Mode AB. Moreover, Chapter 5 will compare CC-RRT with Analytic Sampling to the state of the art BLG algorithm. The comparison of CC-RRT with Analytic Sampling to CC-RRT Mode AB is left to Chapter 5 so as to avoid redundancy of results.

# Chapter 5

# Simulation Results

This chapter presents the simulation results demonstrating the effectiveness of the CC-RRT with Analytic Sampling algorithm. This chapter considers the CC-RRT with Analytic Sampling algorithm, along with 3 algorithms for comparison. The algorithms considered in this chapter are the same as those outlined in Section 4.1.1: **Mode A**, **Mode AB**, **Analytic**, as well as the state-of-the-art **BLG**.

This chapter demonstrates the effectiveness of the CC-RRT with Analytic Sampling algorithm on the Valley Terrain. Furthermore, this chapter extends this claim to show that the CC-RRT with Analytic Sampling algorithm remains invariant to increasingly difficult terrain and responds nearly identically to the Valley Terrain as the algorithm does on a flat terrain scenario. BLG, on the other hand, is unable to demonstrate the same resilience to terrain changes. Moreover, this chapter demonstrates that the CC-RRT with Analytic Sampling planning algorithm is capable of handling increased drop altitude without a significant drop in performance, whereas BLG is unsuitable for use at such altitudes.

## 5.1 Simulation Configuration

### 5.1.1 Wind Profiles

Twenty five wind profiles were used in generating the results demonstrated here. These profiles are drawn from the 194 wind profiles released by Draper Laboratories [46]. Of the 25 used profiles, 7 were artificially generated, while 18 are the result of collected drop data. 6 of the artificially generated profiles represent constant winds of varying intensity applied in cardinal (North, South, East, West) directions. These profiles range in intensity from no wind to 12.9m/s (over 70% of the parafoil airspeed). The $7^{\text{th}}$ artificially generated profile represents an exponentially decreasing wind, with an average wind speed change of $0.0025\frac{\text{m/s}}{\text{m}}$ and a maximum wind speed change of $0.05\frac{\text{m/s}}{\text{m}}$.

The actual drop wind profiles are significantly more aggressive. These profiles have an average overall intensity of 6.7m/s and gust up to 17.1m/s (over 95% of the parafoil airspeed). Additionally, the actual profiles have an overall average intensity change of $0.025\frac{\text{m/s}}{\text{m}}$, and a maximum of $2.4\frac{\text{m/s}}{\text{m}}$. Actual wind profiles are also subject to rapid directional changes, in excess of 2rad/m (115°/m).

### 5.1.2 CC-RRT Simulations

The implementations of all of the CC-RRT algorithms have 2 modes of operation, a real-time running mode and a fixed sample mode. In the real-time running mode, the tree is grown according to the procedure outlined in Section 1.3.4 for 60% of the 1 Hz growth cycle. In the fixed sample mode, a pre-specified number of samples are generated, according to the sampling strategy outlined in Section 1.3.4, and the computation time requires is ignored. For the simulation results presented in this thesis 165 samples are generated at each planning cycle. This number was chosen as it represents the number of samples generated in a 1 Hz planning cycle by the nominal RRT algorithm.

For each CC-RRT algorithm, 500 trials are run in order to gain a complete statistical representation of the landing distribution. With 500 trials, the mean performance has converged, but more importantly, outlier cases are likely to have presented. CC-RRT Mode AB, however, only runs 200 trials. In practice, 200 is enough for an accurate statistical picture, as well as for the mean to converge, and, as will be shown later, outlier cases for CC-RRT Mode AB present after 200 trials.

### 5.1.3 BLG Simulation

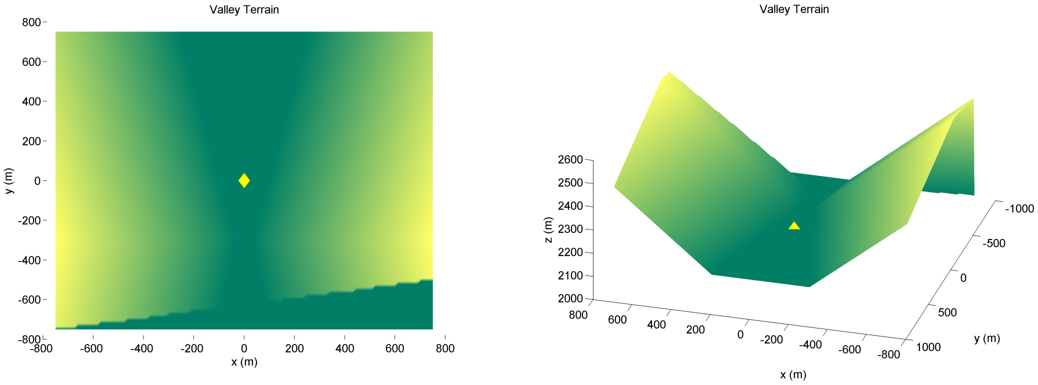
Similar to the CC-RRT algorithms, the BLG optimization engine requires a stopping criteria. Ideally, one would use a tolerance, however, to keep the computation allowed for each of the algorithms comparable, BLG is permitted to simulate the parafoil to the ground 75 times (approximately the equivalent propagated time for 165 CC-RRT samples).

Since BLG does not require lag states, initial conditions can be randomly generated. For this work, 500 initial conditions were generated by randomly sampling $r \sim [100, 400]$, $\theta \sim [0, 2\pi]$, and $\psi_0 \sim [0, 2\pi]$ from uniform distributions. $x_0$ and $y_0$ are then initialized by $x_0 = r\cos\theta$ and $y_0 = r\sin\theta$, and $z_0$ is set to 500m.

## 5.2 Valley Terrain

The major terrain feature of the Valley Terrain scenario is the terrain slope is steeper than the glide-slope of the parafoil. This implies that trajectories cannot approach the goal perpendicular to the terrain valley. Reproduced for convenience from Section 1.3.5, the Valley Terrain scenario is shown in Figures 5-1(a) and 5-1(b).

First consider CC-RRT Mode AB and CC-RRT with Analytic Sampling. Figures 5-2 to 5-3 and Table 5.1 demonstrate the superior performance of CC-RRT with Analytic Sampling over CC-RRT Mode AB for the parafoil guidance problem on difficult terrain. Nominal operating cases show CC-RRT with Analytic Sampling performing approximately 25% better than CC-RRT Mode AB, and worst case scenarios demonstrating a nearly 50% improvement. With CC-RRT algorithms, worst case planning scenarios are caused by unanticipated terrain collisions; terrain collisions that the ap-



(a) Valley Terrain Top View

(b) Valley Terrain Skew-Side View

Figure 5-1: Valley Terrain Images

proximate state probability distribution failed to capture. However, when the worst case scenarios are represented within the probability distribution, the planner is able to compensate and select plans which are appropriately robust so as to avoid worst case scenarios. The improvement demonstrated by CC-RRT with Analytic Sampling substantiates the claim in Chapter 4 that CC-RRT with Analytic Sampling generates a representation of the uncertainty distribution which is more useful in practice than the distribution created by the CC-RRT Mode AB approach. Specifically, the CC-RRT with Analytic Sampling approach demonstrates a superior ability to detect terrain collisions.

Figures 5-2 and 5-3 show considerable similarity in the nominal results of CC-RRT Mode A and CC-RRT with Analytic Sampling, similarity which continues to the 95th percentile. This similarity is highlighted in Table 5.1, where the mean, 50th, 80th, 90th, and 95th percentiles for CC-RRT Mode A and CC-RRT with Analytic Sampling are within 10%. However, there is significant improvement in the worst case scenario for CC-RRT with Analytic Sampling. While CC-RRT Mode A demonstrates worst cases exceeding $500m$, while the worst case scenario for CC-RRT with Analytic Sampling is $218m$.

Finally, consider CC-RRT with Analytic Sampling relative to the BLG algorithm. CC-RRT with Analytic Sampling demonstrates significant improvement over the BLG algorithm, both in nominal performance as shown in Figure 5-2, with mean, 80th,

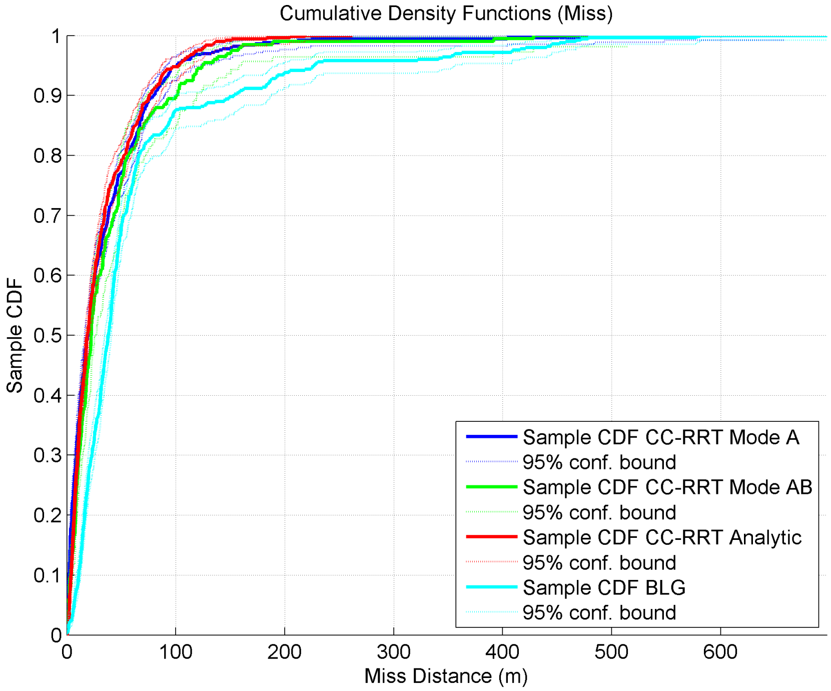


Figure 5-2: Miss Distance CDF on Valley Terrain

90th, and 95th percentiles performing 50% better against the Valley Terrain, as well as worst case, where the largest miss distance recorded for the CC-RRT with Analytic Sampling algorithm is 40% of the largest miss distance recorded by BLG (Table 5.1). The direct optimization technique of BLG does not consider off-nominal future terrain interactions (e.g. terrain interactions caused by changing wind conditions), ultimately resulting in potentially poor terrain interactions (worst cases).

To understand the performance of the BLG algorithm, consider the following example. Shown in Figure 5-4(a) is the overlay of two trajectories, the trajectory planned by the BLG algorithm (shown in red), and the trajectory executed by the parafoil system (blue). The initial position and heading of the parafoil is denoted by the green triangle, while the planner goal is denoted in yellow. Figure 5-4(b) shows the point of deviation between the planned and the executed trajectory. The terrain collision, denoted by $*$ for executed trajectory, is the result of a deviation of less than 1m from the planned trajectory, as illustrated by the yellow line connecting the terminating point of the executed trajectory with the planned trajectory. This

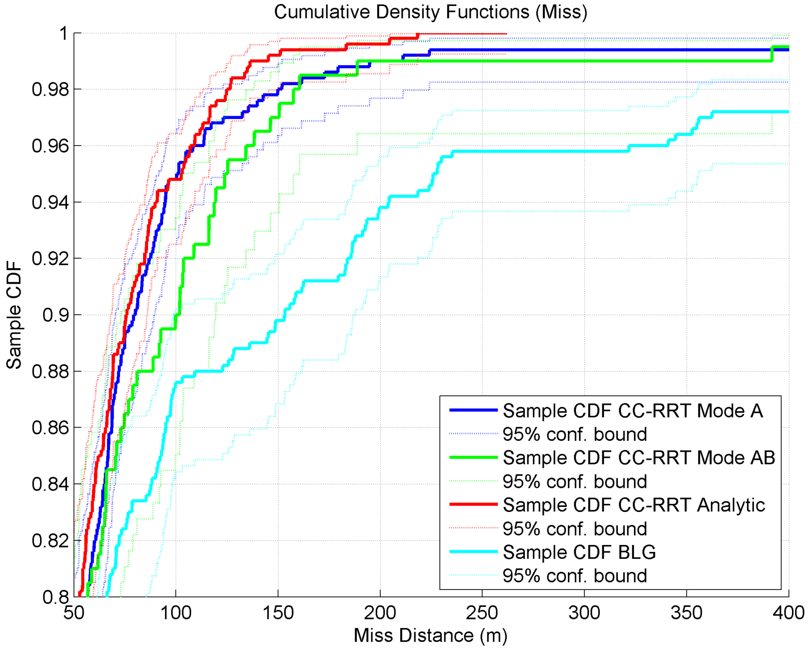


Figure 5-3: Miss Distance CDF on Valley Terrain Zoom to 50m−400m

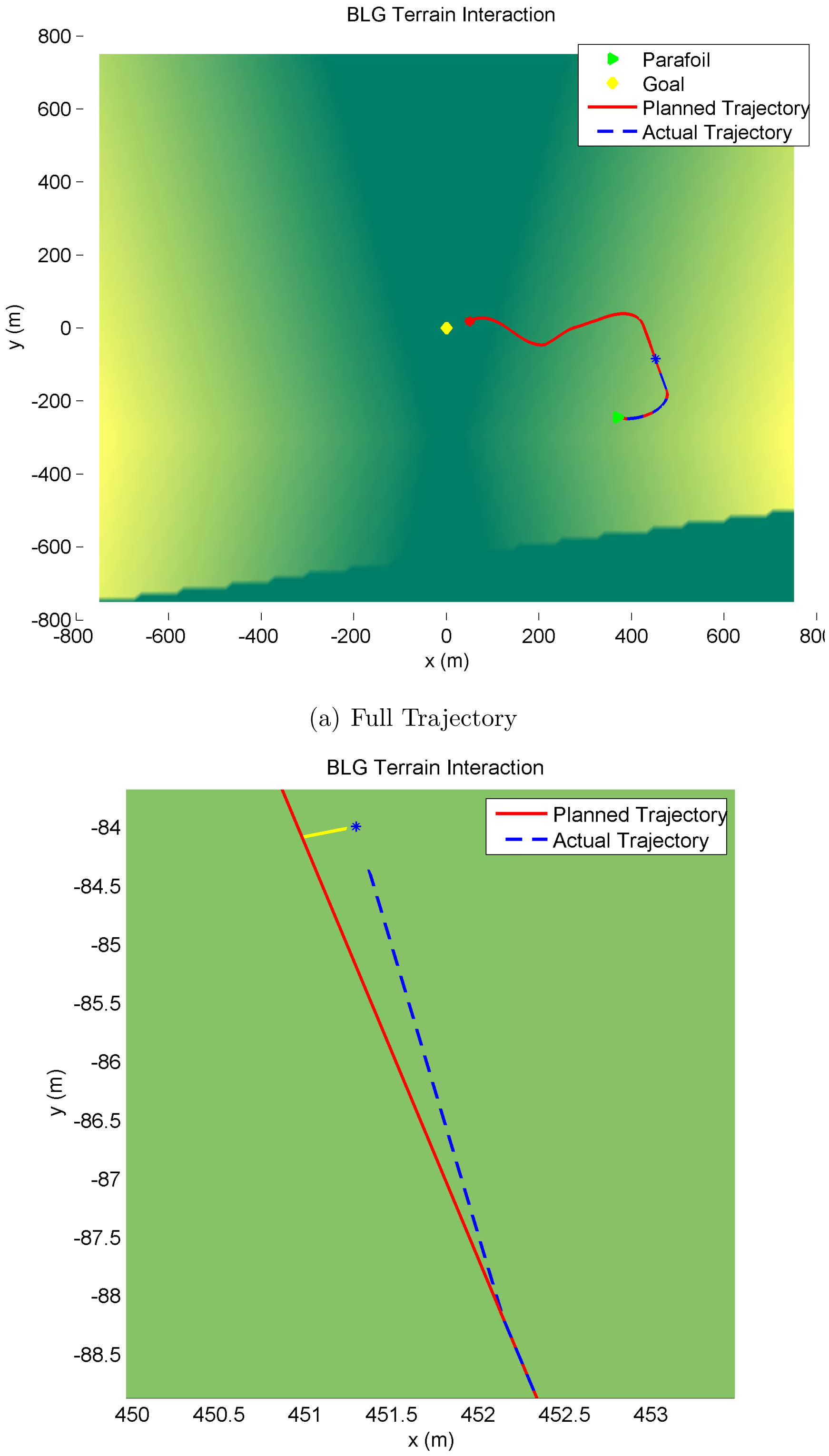
Table 5.1: Miss Distance Data for Valley Terrain Comparison

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| CC-RRT Mode A | 500 | 33.9 | 49.3 | 19.3 | 56.6 | 79.7 | 101 | 151 | 548 |
| CC-RRT Mode AB | 200 | 39.2 | 52.6 | 22.3 | 56.4 | 101 | 125 | 159 | 428 |
| CC-RRT Analytic | 500 | 30.8 | 32.8 | 18.7 | 52.5 | 75.8 | 103 | 126 | 218 |
| BLG | 500 | 63.5 | 89.0 | 37.9 | 66.1 | 153.2 | 226.9 | 430.5 | 581 |

deviation is due to an unexpected wind change, which caused a terrain interaction. Such a scenario is not considered by the BLG optimization process, and therefore such adverse terrain interactions are possible in off-nominal cases.

## 5.3 Flat Terrain

Planning against a scenario with flat terrain is a natural test for any parafoil planner. However, it remains a difficult planning problem as it retains 3 of the 5 challenges presented in the Problem Overview (Section 1.1). The parafoil remains a highly



(a) Full Trajectory

(b) Zoom to Plan Divergence

Figure 5-4: BLG Planned Trajectory vs Executed Trajectory. The planned BLG trajectory (red) is an example of an extremely accurate planned trajectory; the executed trajectory (blue) slightly deviates from the planned trajectory, yet that deviation (yellow) results in a terrain collision.

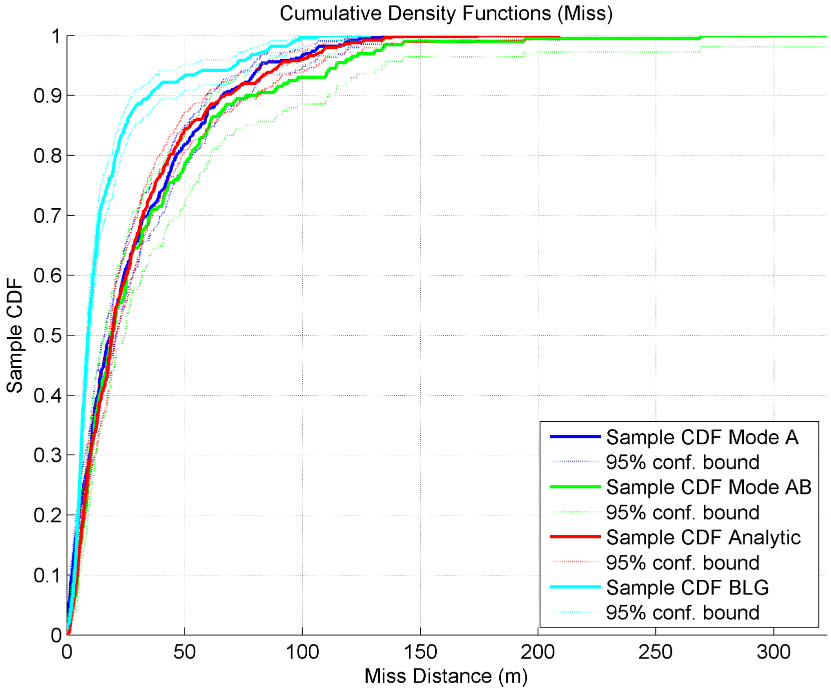


Figure 5-5: Miss Distance CDF on Flat Terrain

non-linear dynamic system (challenge # 1); the parafoil is under-actuated system (challenge # 2), leading to both large turning circles as well as uncontrollable and uncertain landing times; finally, the system environment contains uncertain and variable winds (challenge # 4).

Against flat terrain, CC-RRT Mode A and CC-RRT with Analytic Sampling result in the same algorithm, in terms of planning decisions. In practice, the difference between the two are the additional terrain checks performed by the CC-RRT with Analytic Sampling. Because the CC-RRT with Analytic Sampling generates covariance samples in the $x/y$ plane 4.3.1, against a flat terrain scenario, which does not contain significant terrain obstacles, such additional checks provide no new information to the planner and thus have no effect on the choices made by the planner. This effect can be seen in Figure 5-5, as well as in Table 5.2, showing mean performance differing by 0.6m, and worst case performance differing by 25m (15%).

Consider CC-RRT with Analytic Sampling relative to CC-RRT Mode AB. As shown in Figure 5-5, and also in Table 5.2, in nominal cases, CC-RRT with Analytic

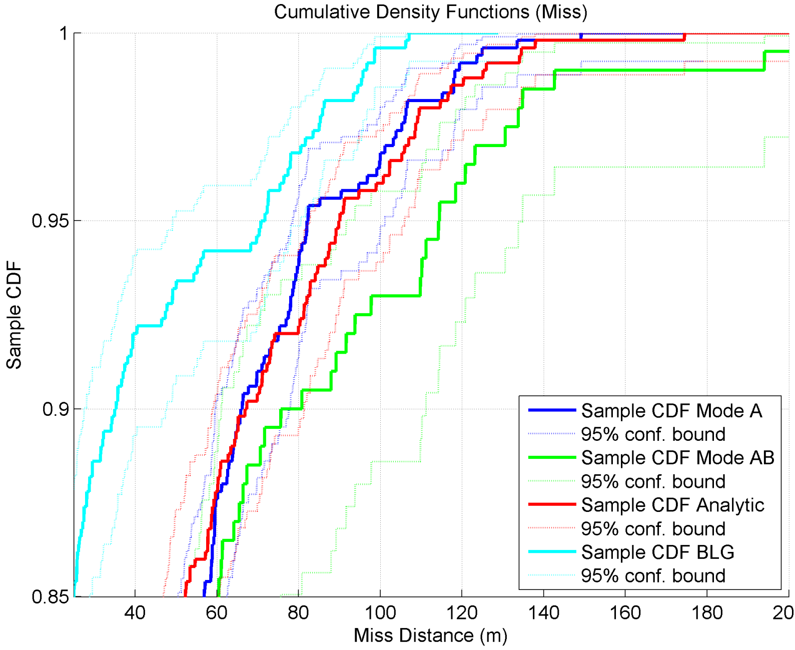


Figure 5-6: Miss Distance CDF on Flat Terrain zoom to 25m−200m

Table 5.2: Miss Distance Data for Flat Terrain Comparison

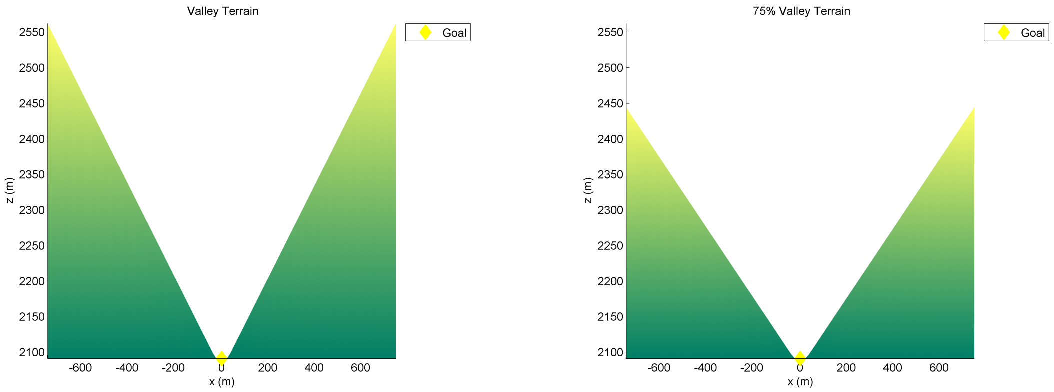
| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| Mode A | 500 | 27.8 | 27.7 | 18.7 | 46.5 | 66.1 | 82.2 | 107 | 149 |
| Mode AB | 200 | 32.8 | 37.5 | 19.0 | 52.5 | 78.3 | 114 | 134 | 269 |
| Analytic | 500 | 28.4 | 28.1 | 19.3 | 43.6 | 67.1 | 90.0 | 112 | 174 |
| BLG | 500 | 15.9 | 19.7 | 8.9 | 20.7 | 35.2 | 71.3 | 86.1 | 107 |

Sampling and CC-RRT Mode AB perform comparably, with the mean, 50th, and 80th percentile performance falling within 10%. However, as in the case of the Valley Terrain, there is a demonstrable difference in off-nominal performance, with CC-RRT with Analytic Sampling outperforming CC-RRT Mode AB by nearly 100m (37%). While part of this difference is due to random variation in the planner, the representation of the mean of the uncertainty distribution is a significant contributing factor. CC-RRT with Analytic Sampling utilizes a direct measure of the mean of the uncertainty distribution, as one of the samples is (by construction) guaranteed to be the predicted mean, while CC-RRT Mode AB relies on the assumption that the mean of the propagated particles accurately accounts for the mean of the uncertainty distribution. An off-nominal case for CC-RRT Mode AB is one in which the mean of the uncertainty distribution is not captured by the mean of the particles. This type of a case is plausable, as only 10 particles are used for CC-RRT Mode AB. Failing to consider mean trajectory position can lead to poor performance in the off-nominal cases.

Finally, consider the BLG algorithm. As is evident from the nominal performance shown in Table 5.2, as well as the worst case performance shown in Figure 5-6, BLG significantly out-performs the CC-RRT with Analytic Sampling algorithm, with mean, 50th, 80th, and 90th percentile BLG miss distances landing 50% closer than CC-RRT with Analytic Sampling, and worst case miss distance landing 40% closer. The CC-RRT with Analytic Sampling algorithm finds feasible solutions to the planning problem, while BLG plans an optimal path. In general, the CC-RRT approach as presented in this thesis is not well suited to addressing this problem. Since the flat terrain scenario lacks significant terrain obstacles, finding feasible solutions (the strength of the RRT based algorithms) is a relatively simple task (while still non-trivial), while optimizing the planned trajectory (the strength of the BLG algorithm) is the most efficient use of the available computational resources. As mentioned previously, the terrain checks performed by the CC-RRT with Analytic Sampling algorithm provide no new information to the planner and thus have no effect on the choices made by the planner. Since the samples provide no new information, creating them and determining feasibility for each was is wasted computational effort.

## 5.4 CC-RRT Invariance to Terrain

The following results will utilize a terrain scenario named "75% Valley Terrain". The naming convention for this terrain references the slope of the valley walls. The "75% Valley Terrain" has 75% the slope of the Valley Terrain. This difference is illustrated in the side views comparing the Valley Terrain in Figure 5-7(a) and the 75% Valley Terrain in Figure 5-7(b). This terrain scenario represents an intermediate step in difficulty between flat terrain and the full Valley Terrain. Testing against this terrain



(a) Side View Valley Terrain (b) Side View %75 Valley Terrain

Figure 5-7: Terrain Comparisons

allows for understanding how simulation results evolve from flat terrain to the full Valley Terrain scenario.

Figure 5-8 shows the response of the BLG algorithm to terrains of increasing difficulty, and Table 5.3 represents select data points in tabular form. As the terrain becomes more complex (represented by a steeper slope), feasible paths become more difficult to find. In the flat terrain case, as mentioned above, a feasible solution is less difficult to find (relative to more difficult terrain cases), and therefore can be optimized to improve performance. On the 75% Valley Terrain, we see that there exists a regime of nominal performance up to 45m miss distance in which BLG performs incredibly well, with 88% of cases falling below this mark. It is in these cases where finding a feasible solution is relatively straightforward and BLG is able to optimize the solution. Above 45m miss distance, interaction with terrain becomes an issue and finding a feasible solution becomes difficult, and the BLG algorithm has a significant increase in miss distance due to terrain collisions. On the full Valley Terrain, feasible solutions are more difficult to find in general, and therefore attempting to optimize a poor feasible solution leads to poor overall algorithm performance, as discussed earlier.

Next, consider the response of the CC-RRT with Analytic Sampling algorithm against the same terrains, as shown in Figure 5-9 and with selected points reproduced in tabular form in Table 5.4. When considering the data, there are 3 important items

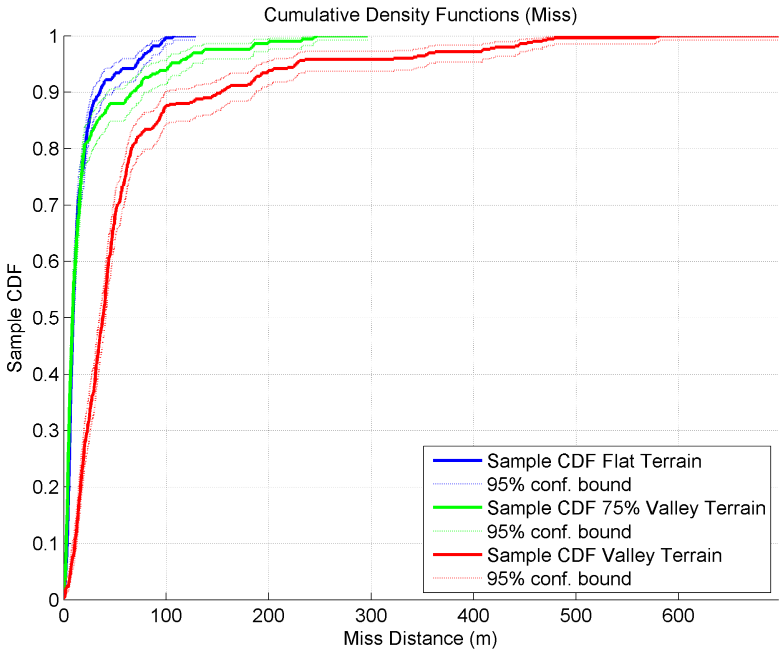


Figure 5-8: Miss Distance CDF for BLG Against Various Terrain Scenarios

Table 5.3: Miss Distance Data for BLG Terrain Comparison

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| Flat Terrain | 500 | 15.9 | 19.7 | 8.9 | 20.7 | 35.2 | 71.3 | 86.1 | 107 |
| 75% Valley Terrain | 500 | 22.5 | 39.5 | 8.5 | 20.1 | 67.1 | 105 | 184 | 247 |
| Valley Terrain | 500 | 63.5 | 89.0 | 37.9 | 66.1 | 153 | 227 | 431 | 581 |

to note. First, the mean miss distance of the CC-RRT with Analytic Sampling algorithm changes by less than 4m, an increase of less than 15%, while the mean miss distance of the BLG algorithm increases by more than a factor of 4. Second, the worst case miss distance for the CC-RRT with Analytic Sampling algorithm increases by approximately 25%, while the worst case miss distance for BLG increases by more than a factor of 5. Third, upon inspecting the CDF curves in Figure 5-9, there is little discernible difference between the shape of the curves, a contention that is further substantiated by the data in Table 5.4, where the largest difference between 2 corresponding table elements is an increase of 40m from flat terrain to the full Valley Terrain in the worst case. The near-identical CDF curves suggest that the CC-

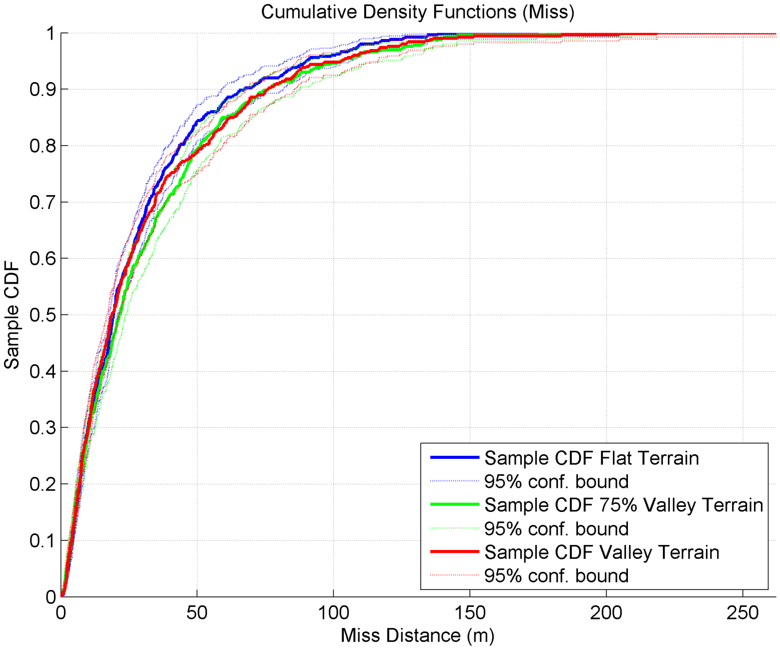


Figure 5-9: Miss Distance CDF for CC-RRT with Analytic Sampling Against Various Terrain Scenarios

Table 5.4: Miss Distance Data for CC-RRT with Analytic Sampling Terrain Comparison

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| Flat Terrain | 500 | 28.4 | 28.1 | 19.3 | 43.6 | 67.1 | 90.0 | 112 | 174 |
| 75% Valley Terrain | 500 | 32.0 | 32.2 | 21.6 | 51.2 | 76.0 | 103 | 133 | 191 |
| Valley Terrain | 500 | 30.8 | 32.8 | 18.7 | 52.5 | 75.8 | 103 | 126 | 218 |

RRT with Analytic Sampling algorithm is able to maintain consistent performance regardless of the difficulty of the terrain scenario.

## 5.5 High Altitude

One of the advantages of the CC-RRT with Analytic Sampling algorithm, and of RRT algorithms in general, which is regularly discusses in this thesis, is the capability to vary the initial planner altitude. Figure 5-10 and Table 5.5 present simulation results considering starting altitudes from 500m to 2000m. When considering the data, there are 3 important items to note. First, the mean miss distance of the CC-RRT with Analytic Sampling algorithm changes by 1m, an increase of less than 4%. Second, the worst case miss distance for the CC-RRT with Analytic Sampling algorithm increases by 27m, approximately 12%. Third, upon inspecting the CDF curves in Figure 5-10, there is little discernible difference between the shape of the curves, a contention that is further substantiated by the data in Table 5.4, where the largest difference between 2 corresponding table elements is an increase of 27m. This data suggests that CC-RRT with Analytic Sampling is capable of operating at high altitudes without a deterioration in performance. Other approaches in the literature, such as those described in Refs. [2, 14, 16–18, 20], require an upper limit on the altitude for initiation of terminal guidance in order to remain computationally tractable.

## 5.6 Summary

This chapter demonstrates that CC-RRT with Analytic Sampling is an effective approach to the parafoil terminal guidance problem under significant wind uncertainty/terrain interaction. The greatest strength of the algorithm is the invariance to increasingly difficult terrain conditions relative to the state-of-the-art BLG algorithm. The analytic sampling, presented in Chapter 4, captures a representation of the state distribution, allowing for a measure of the robustness of a path. Considering the robustness of a candidate path allows the CC-RRT with Analytic Sampling algorithm to plan against worst case scenario outcomes. The wind model and classification scheme presented in Chapter 3 informs the analytic sampling approach, ensuring conservative approaches when the wind appears to have a large effect on the system, and permitting more aggressive paths when the wind is believed to have little effect. Moreover, unlike other approaches which incorporate wind into the planning framework ([14, 61]), this approach is capable of handling arbitrary wind profile shapes.

In addition to demonstrating a robustness to aggressive terrain scenarios, we have shown that the CC-RRT with Analytic Sampling algorithm capable of handling initial

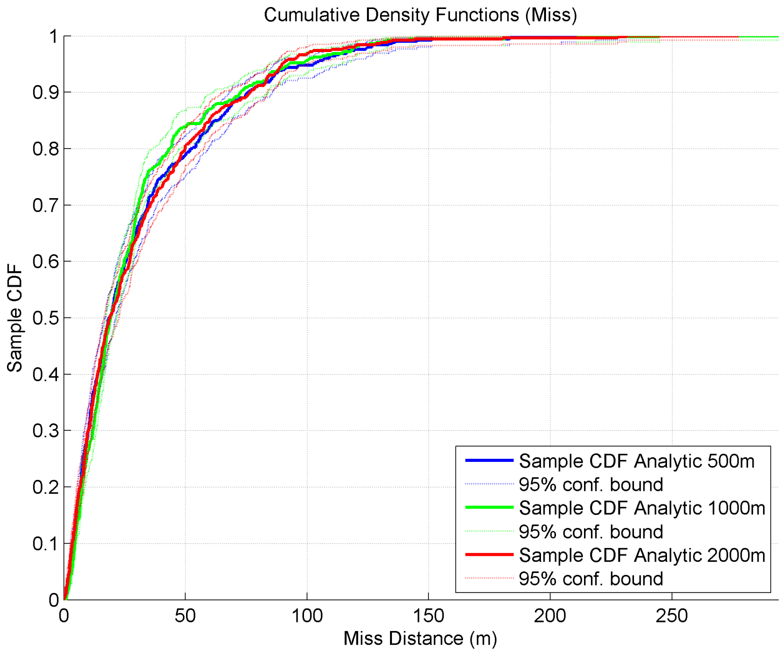


Figure 5-10: Miss Distance CDF for CC-RRT with Analytic Sampling with Various Starting Altitudes

Table 5.5: Miss Distance Data for CC-RRT with Analytic Sampling with Various Starting Altitudes

| **Name** | **N** | **Mean (m)** | **SD ($\sigma$)** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| $z_0 = 500$m | 500 | 30.8 | 32.8 | 18.7 | 52.5 | 75.8 | 103 | 126 | 218 |
| $z_0 = 1000$m | 500 | 29.8 | 32.9 | 19.2 | 43.5 | 72.7 | 92.6 | 120 | 245 |
| $z_0 = 2000$m | 500 | 30.7 | 33.4 | 19.0 | 49.8 | 76.7 | 89.8 | 118 | 231 |

altitude conditions from 500m to 2000m. The cost-to-go function developed in Chapter 2 biases the planner toward choosing high-margin paths at high altitudes, while guiding the parafoil toward the goal during the ending stages of terminal guidance.

The major drawback of the CC-RRT with Analytic Sampling approach stems from the additional terrain collision checks that are required for the analytic sampling, particularly in the case when no terrain obstacles exist. When no, or few, terrain obstacles exist, significant computational effort is spent finding a feasible solution (RRT tree growth), and ensuring that the solution is robust to uncertain wind-obstacle interactions (analytic sampling). Direct optimization techniques, such as BLG, utilize the computational resources to optimize a feasible trajectory.

# Chapter 6

# Conclusion and Future Work

## 6.1 Conclusion

In this work a novel approach to on-line trajectory planning and robust obstacle avoidance for a large autonomous parafoil is developed. This planning strategy, CC-RRT with Analytic Sampling, robustly executes collision avoidance with arbitrary, non-convex, mapped terrain. The underlying rapidly exploring random tree satisfies the hard vehicle constraints by construction, while the analytic sampling maintains probabilistic path feasibility by considering the wind variability about the prevailing mean. The CC-RRT with Analytic Sampling algorithm has been verified in simulation against the state-of-the-art BLG algorithm developed by Draper Laboratories in Ref. [2].

Each chapter has explored a component of the CC-RRT with Analytic Sampling planner which is necessary for applying it to the parafoil terminal guidance problem. Chapter 2 develops a cost-to-go function used by the underlying RRT algorithm to consider the viability of partially planned paths. The cost-to-go function presented provides a balance between margin above the parafoil glide-slope to counter-act future disturbances and utilization of a glide-slope trajectory to minimize range to target at impact.

Chapter 3 develops a variational wind disturbance model. The *DP-means* clustering algorithm is used to automatically generate the appropriate number of clusters for the wind profiles. Next the width of the impulse response filter for prevailing wind estimate is determined. Finally, direct optimization is used to tune colored noise models to fit the landing variation. The effectiveness of the multi-class classification scheme is demonstrated through forced single classification, demonstrating the appropriate use of each of the potential wind classifications in order to plan aggressive paths when the wind is believed to have little effect, and maintain a conservative approach when the wind appears to have a greater effect.

Chapter 4 introduces analytic covariance sampling as a method for probabilistic constraint checking. This approach leverages the structure of the colored noise model to derive an analytic expression for the system covariance as a function of altitude. The approach further samples this covairance to allow for computationally efficient probabilistic constraint checking.

The success of this approach is demonstrated by verifying the success criteria set forth in Section 1.3.1. As previously mentioned, the RRT algorithm satisfies the hard vehicle constraints by construction (# 2). The analytic sampling explicitly considers variations in the wind during future time steps (# 3) to evaluate probabilistic path feasibility (# 4). The use of a cost-to-go function in the RRT framework allows the planner to operate given an arbitrary starting altitude (# 5), although further work is necessary to tune algorithm parameters to these conditions. Section 5 demonstrates the improvement of the CC-RRT with Analytic Sampling over the state of the art BLG alorithm (# 1).

## 6.2 Future Work

### 6.2.1 Analytic Sampling Extension

While this work presents and evaluates an effective, novel approach to parafoil uncertainty, it only considers the uncertainty in the $x/y$ plane. While this has proven effective in avoiding unwanted terrain collisions, updrafts and downdrafts remain as a source of unmodeled uncertainty in the planning problem. Extending the analytic sampling approach into 3 dimensions will add a level of robustness to a downdraft causing a terrain collision. If one assumes that the uncertainty in the vertical direction is uncorrelated with the uncertainty in $x/y$, then augmentation can be broken down into 3 tasks:

1. Re-tune the colored noise wind model parameters from Section 3.2.3. This must be done in two steps, first, re-tune the model for $x/y$ assuming no vertical disturbance. Second, tune a vertical model by assuming zero $x/y$ disturbance.
2. Replicate the approach in Sections 4.3 and 4.4 for the vertical disturbance.
3. These covariances must be used in conjunction with a methodology to produce a set of equi-spaced covariance samples around the 3 dimensional covariance ellipsoid. Such an approach is outlined in Ref. [62] Section 3.5.4.

In addition to performing the above augmentation to the model and the uncertainty characterization, incorporating the vertical dimension requires addressing the distinction between a landing and a collision. Fundamentally, the planner desires a landing to be a high probability event, so the overall cost of the trajectory can be accurately computed. A collision, on the other hand, is, in many cases, a low probability event. This dichotomy of conditions that the planner must operate under implies that the samples cannot be handled uniformly in all cases (as is done in the approach discussed in Section 4.3.1).

### 6.2.2 Cost Function Augmentations

#### Cost Function Distribution

Ref. [14] proposes directly incorporating stochastic effects into the terminal cost function. Extension of the cost-to-go framework demonstrated in this thesis to incorporate stochastic effects would provide additional robustness to the approach. The robustness addressed in this thesis focused on maintaining dynamically feasible solutions. Adding stochastic consideration to the cost-to-go and terminal cost functions would provide robustness with respect to actual landing state, resulting in more consistent miss distances.

### Additional Cost Function Considerations

Ref. [18] proposes the notion of biasing plans in order to minimize the system sensitivity to imperfectly controlled heading rate. The notion proposed in Ref. [18] claims that system impact of heading rate uncertainty is exacerbated by high parafoil ground speeds, and decreasing the ground speeds will reduce the uncertainty in system execution. Future work on this project should consider this claim and how it may be incorporated into the cost-to-go proposed in Chapter 2.

# Bibliography

[1] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, August 2010.

[2] D. Carter, L. Singh, L. Wholey, M. McConley, S. Tavan, B. Bagdonovich, T. Barrows, C. Gibson, S. George, and S. Rasmussen, "Band-limited guidance and control of large parafoils," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2009.

[3] *Popular Mechanics*. No. v. 52, Hearst Magazines, 1929.

[4] J. Oliva, "Jetpix." http://www.jetpix.com/webfiles/440AW/parapanarama1500.jpg, May 2013.

[5] P. Hattis, S. Tavan, and A. Soldier, "Precision airdrop," *Aerospace America*, pp. 38–42, 2007.

[6] "Computed air release point procedures," *Air Force Instruction 11-231*, 2005.

[7] P. Hattis, T. Fill, D. Rubenstein, R. Wright, and R. Benney, "An advanced on-board airdrop planner to facilitate precision payload delivery," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, August 2000.

[8] R. Wright, R. Benney, and J. McHugh, "An on-board 4D atmospheric modeling system to support precision airdrop," in *Infotech@Aerospace Conferences*, American Institute of Aeronautics and Astronautics, September 2005.

[9] R. Wright, R. Benney, and J. McHugh, "Precision airdrop system," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2005.

[10] D. Jalbert, "Multi-cell wing type aerial device," 11 1966.

[11] G. Menard, J. Nicolaides, and R. Speelman, III, "A review of para-foil applications," *Journal of Aircraft*, vol. 7, pp. 423–431, September 1970.

[12] "Joint drop," May-June 2003.

[13] "Draper laboratory - 2010: Transitioning technology," 2010.

[14] J. Rogers and N. Slegers, "Terminal guidance for complex drop zones using massively parallel processing," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2013.

[15] N. Slegers and M. Costello, "Model predictive control of a parafoil and payload system," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 816–821, July 2005.

[16] A. Calise and D. Preston, "Swarming/flocking and collision avoidance for mass airdrop of autonomous guided parafoils," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, August 2005.

[17] N. Slegers, E. Beyer, and M. Costello, "Use of variable incidence angle for glide slope control of autonomous parafoils," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 585–596, May 2008.

[18] S. Bergeron, K. Tavan and A. Fejzic, "Accuglide: Precision airdrop guidance and control via glide slope control," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2011.

[19] T. Gimadieva, "Optimal control of a gliding parachute system," *Journal of Mathematical Sciences*, vol. 103, no. 1, pp. 54–60, 2001.

[20] O. A. Yakimenko and N. Slegers, "Using direct methods for terminal guidance of autonomous aerial delivery systems," tech. rep., DTIC Document, 2009.

[21] B. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, (Toronto, Canada), August 2010. (AIAA-2010-8160).

[22] B. Luders and J. P. How, "Probabilistic feasibility for nonlinear systems with non-Gaussian uncertainty using RRT," in *AIAA Infotech@Aerospace Conference*, (St. Louis, MO), (AIAA-2011-1589), March 2011.

[23] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006. Available at http://planning.cs.uiuc.edu/.

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[25] N. Roy, G. Gordon, and S. Thrun, "Planning under uncertainty for reliable health care robotics," in *Springer Tracts in Advanced Robotics* (S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, eds.), vol. 24, pp. 417–426, Springer Berlin Heidelberg, 2006.

[26] A. Foka and P. Trahanias, "Real-time hierarchical POMDPs for autonomous robot navigation," in *IJCAI Workshop Reasoning with Uncertainty in Robotics*, 2007.

[27] M. Greytak and F. Hover, "Motion planning with an analytic risk cost for holonomic vehicles," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 5655–5660, 2009.

[28] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3313–3318, 2007.

[29] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1261–1267, 2006.

[30] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," tech. rep., Iowa State University, October 1998.

[31] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, pp. 1105–1118, September 2009.

[32] B. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *American Control Conference (ACC)*, (Piscataway, NJ), pp. 5406–5412, IEEE, June/July 2010.

[33] B. Rademacher, *In-flight trajectory planning and guidance for autonomous parafoils.* PhD thesis, Iowa State University, January 2009.

[34] D. Jacob, *Introduction to Atmospheric Chemistry.* Princeton University Press, 1999.

[35] D. Carter, S. George, P. Hattis, L. Singh, and S. Tavan, "Autonomous guidance, navigation and control of large parafoils," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2005.

[36] S. Carter, D and. George, P. Hattis, M. McConley, S. Rasmussen, L. Singh, and S. Tavan, "Autonomous large parafoil guidance, navigation, and control system design status," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2007.

[37] "fmincon." http://www.mathworks.com/help/optim/ug/fmincon.html, June 2013. MATLAB version 7.12.0. Natick, Massachusetts: The MathWorks Inc., 2011.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization.* New York: Springer, 2nd ed., 2006.

[39] T. Schouwenaars, B. DeMoor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *European Control Conference 2001*, pp. 2603–2608, 2001.

[40] A. Bellingham, J. Richards and J. How, "Receding horizon control of autonomous aerial vehicles," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 5, pp. 3741–3746 vol.5, 2002.

[41] Y. Kuwata, *Trajectory Planning for Unmanned Vehicles using Robust Receding Horizon Control.* PhD thesis, Massachusetts Institute of Technology, 2003.

[42] D. Delahaye and S. Puechmorel, "Aircraft local wind estimation from radar tracker data," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pp. 1033–1038, 2008.

[43] J. Petrich and K. Subbarao, "On-board wind speed estimation for uavs," in *Guidance, Navigation, and Control and Co-located Conferences*, American Institute of Aeronautics and Astronautics, August 2011.

[44] K. Hunt and G. P. Nason, "Wind speed modelling and short-term prediction using wavelets," 2001.

[45] X. Jiang, B. Dong, L. Xie, and L. Sweeney, "Adaptive gaussian process for short-term wind speed forecasting," in *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, (Amsterdam, The Netherlands, The Netherlands), pp. 661–666, IOS Press, 2010.

[46] J. How, L. Breger, A. Kellas, and C. Dever, "Personal correspondence." E-mails dated October 11th, 2011 and March 29th, 2012.

[47] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Pearson Education, 2 ed., 2003.

[48] S. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[49] E. Forgy, "Cluster analysis of multivariate data: efficiency vs interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[50] J. Hartigan and M. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[51] C. Sugar and G. James, "Finding the number of clusters in a data set: An information theoretic approach," *Journal of the American Statistical Association*, vol. 98, pp. 750–763, 2003.

[52] B. Kulis and M. Jordan, "Revisiting k-means: New algorithms via bayesian nonparametrics," *CoRR*, vol. abs/1111.0352, 2011.

[53] A. V. Oppenheim, A. Willsky, and S. Nawab, *Signals & systems (2nd ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[54] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[55] T. Sejnowski and C. Rosenberg, "Parallel networks that learn to pronounce english text," 1987.

[56] E. Kong and T. Dietterich, "Error-correcting output coding corrects bias and variance," in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 313–321, Morgan Kaufmann, 1995.

[57] B. Luders and J. P. How, "Probabilistic feasibility for nonlinear systems with non-Gaussian uncertainty using RRT," in *AIAA Infotech@Aerospace Conference*, (St. Louis, MO), (AIAA-2011-1589), March 2011.

[58] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *In Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[59] S. Klerman, "Path planning for autonomous parafoils using particle chance constrained rapidly-exploring random trees in a computationally constrained environment," Master's thesis, Massachusetts Institute of Technology, 2012.

[60] P. S. Maybeck, *Stochastic models, estimation and control. Volume I.* 1979.

[61] L. Fowler and J. Rogers, "Bezier curve path planning for parafoil terminal guidance," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2013.

[62] B. Luders, "Robust trajectory planning for unmanned aerial vehicles in uncertain environments," Master's thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 2008.