# Analytic Chance Constraints for the Robust Guidance of Autonomous Parafoils

by

Aaron Cole Ellertson
B.S., Aerospace Engineering
Georgia Institute of Technology (2013)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2015

Signature redacted

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Aeronautics and Astronautics
May 21, 2015

Signature redacted

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

Signature redacted

Certified by. . . . . . . . . . . . . . . . . . . . . . . . . .
Louis S. Breger
Member of the Technical Staff, Draper Laboratory
Thesis Supervisor

Signature redacted

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . .
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

# Analytic Chance Constraints for the Robust Guidance of Autonomous Parafoils

by

Aaron Cole Ellertson

Submitted to the Department of Aeronautics and Astronautics
on May 21, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

Autonomously guided parafoil systems can deliver supplies and aid to remote, geographically diverse locations, while providing important safety and logistical advantages over ground-based transportation methods. A key challenge facing modern airborne delivery systems, such as parafoils, is the ability to accurately and consistently deliver supplies into difficult, complex terrain. Parafoil guidance algorithms must be able to generate feasible trajectory solutions to the target location within highly constrained terrain environments and from a wide range of initial conditions. Robustness is critical for successful payload delivery in the presence of uncertain atmospheric wind disturbances.

This thesis presents two online trajectory planning algorithms for autonomous parafoil guidance in complex terrain and wind environments. These algorithms are capable of operating from arbitrary initial conditions, including altitude, and are robust to wind disturbances that may be highly dynamic throughout terminal descent. The first algorithm, known as Analytic CC-RRT, builds upon the framework of chance-constrained rapidly-exploring random trees (CC-RRT). This planner enables fast incremental trajectory construction in cluttered, non-convex environments, while using chance constraints to ensure probabilistic feasibility. The designed cost-to-go function prioritizes target accuracy and upwind landings through the selection of partial paths that intelligently consider current and reachable future states. A trained multi-class wind uncertainty model is introduced to classify and anticipate the effect of future wind disturbances online. Utilizing this model, robustness to wind variations is achieved via a novel analytic uncertainty sampling technique, allowing the probability of constraint violation to be efficiently evaluated against arbitrary and aggressive terrain.

The second algorithm, known as CC-BLG, incorporates the Analytic CC-RRT proactive wind model and uncertainty sampling technique into the optimized Band-Limited Guidance (BLG) framework. Through the design of a novel risk-based objective function, CC-BLG trajectories efficiently balance the parafoil performance metrics of landing accuracy and landing speed with the risk of off-nominal terrain collisions caused by future wind disturbances. Proposed extensions to the analytic uncertainty sampling technique are shown to yield enhanced planning robustness by refining the estimation of trajectory risk. Multi-phase CC-BLG path planning enables initialization of parafoil terminal guidance from potentially high altitudes, while discrete reachability set approximation is used to maintain robust obstacle avoidance over disjoint planning horizons.

Extensive Monte Carlo simulation analysis demonstrates that the Analytic CC-RRT and CC-BLG algorithms achieve significant improvements in mean and worst-case landing accuracy within complex terrain scenarios relative to the state-of-the-art Band-Limited Guidance (BLG) algorithm. Flight test experiments conducted with a full-scale *UltraFly* parafoil system confirm that the more computationally efficient CC-BLG algorithm is capable of robust parafoil guidance and precision landings subject to real-world testing conditions, hardware limitations, and challenging terrain environments.

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

Thesis Supervisor: Louis S. Breger
Title: Member of the Technical Staff, Draper Laboratory

# Acknowledgments

This thesis is the product of hard work, commitment, and the contributions of friends, family, and colleagues who have helped me to achieve my dreams.

First, I would like to thank my advisor Professor Jonathan How for his guidance and support throughout my time at MIT. Working with Professor How has been an invaluable experience that has enabled me to mature as both a researcher and an engineer with the confidence to seek out problems, and develop solutions. I am grateful for Professor How's dedication to the success of all his students, and for the pride he shares in our accomplishments.

I would also like to express my gratitude to Draper Laboratory for funding this research, and for providing the tools, resources, and assistance that have helped me to be successful throughout my graduate career. In particular, I would like to thank my advisor Dr. Louis Breger for overseeing my thesis progress at Draper Laboratory, and for his direction and encouragement over the last two years. I would also like to thank members of the airdrop team Brett Streetman, Matthew Neave, and Akil Middleton for offering their time to further my research.

Additionally, I would like to thank Dr. Brandon Luders and Ian Sugel whose research and collaboration have helped to lay the foundation for this thesis work. My sincerest thanks to Brandon for acting as my mentor, and for his patience and guidance throughout our work together. I would also like to thank the members of the Aerospace Controls Laboratory (ACL) at MIT who have given me valuable advice and feedback along the way.

To my parents and to my sister, thank you for believing in me, for encouraging me, and for teaching me the values and the lessons that have made me the person I am today. I am here because of you, and because of how you have inspired me. Finally, to Cynthia Gonzalez-Espinos, thank you for your tireless love, support, and devotion—without which, none of this would have been possible. This is for you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Modern airborne delivery systems must be able to accurately and consistently deliver supplies into difficult, complex terrain. For delivery systems such as parafoils, the terminal guidance problem – guiding the parafoil from a potentially high initial altitude to land precisely with a desired position and heading – presents significant technical challenges, particularly for large, heavy parafoils such as those considered in this work. Parafoil dynamics are highly nonlinear and underactuated, with large turning radii and severely limited or no vertical control, resulting in a descent rate which is influenced primarily by atmospheric conditions and wind disturbances [1]. Parafoil drop locations are often difficult to reach and may be surrounded by arbitrary, nonconvex terrain that can pose a significant problem for constraint satisfaction, even if mapped in advance [2, 3]. Parafoils are also subject to uncertain and variable wind environments, which, if uncompensated, can result in large deviations between predicted and actual trajectories, and undesirable landing errors [4]. Finally, many airborne delivery applications often have tight landing restrictions. Missing the target location, even by a small distance, can lead to unintended collisions with natural or man-made hazards, or even theft of cargo [5]. Precise delivery is essential in order to avoid loss of supplies or unacceptably dangerous recovery efforts.

The parafoil was invented by Domina Jalbert and patented in 1966 as an inflatable, nonrigid canopy wing composed of several contiguous fabric cells [6]. Each cell is shaped like an airfoil section, closed at the trailing edge and open at the leading

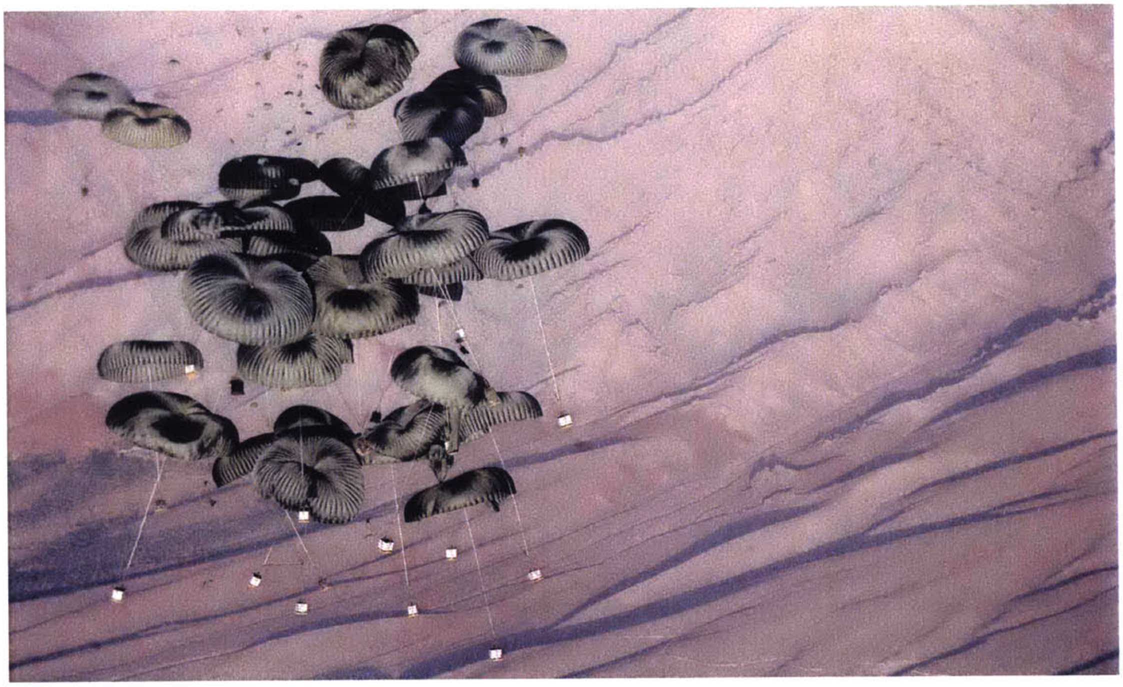


Figure 1-1: Round unguided G-12 parachutes used for cargo delivery [7]

edge, such that inflation of the canopy is produced via incoming ram air pressure [8]. Since the late 1960s, autonomously guided parafoils have been considered a viable alternative to round, unguided parachute systems, shown in Figure 1-1, which were first introduced as a means of airborne cargo delivery during World War II [9]. Ram-air parafoils offer several advantages over round parachutes for precision airdrop, where perhaps the most important advantage is increased vehicle maneuverability [6, 9]. This is achieved via the parafoil control lines used to generate asymmetric deflection of the canopy trailing edge. In addition, parafoils are designed to fly at higher airspeeds compared to conventional round canopies, allowing them to penetrate significant atmospheric winds in order to reach the intended target location [10]. While unguided airdrop systems have traditionally been constrained to low release altitudes to maintain acceptable landing accuracy, parafoils can be released from much higher altitudes and larger aircraft standoff distances with the potential for significantly improved accuracy and precision [11]. This result is particularly beneficial for both military resupply applications and airdrop scenarios involving challenging environmental terrain, for which low release altitudes may be dangerous or inaccessible to

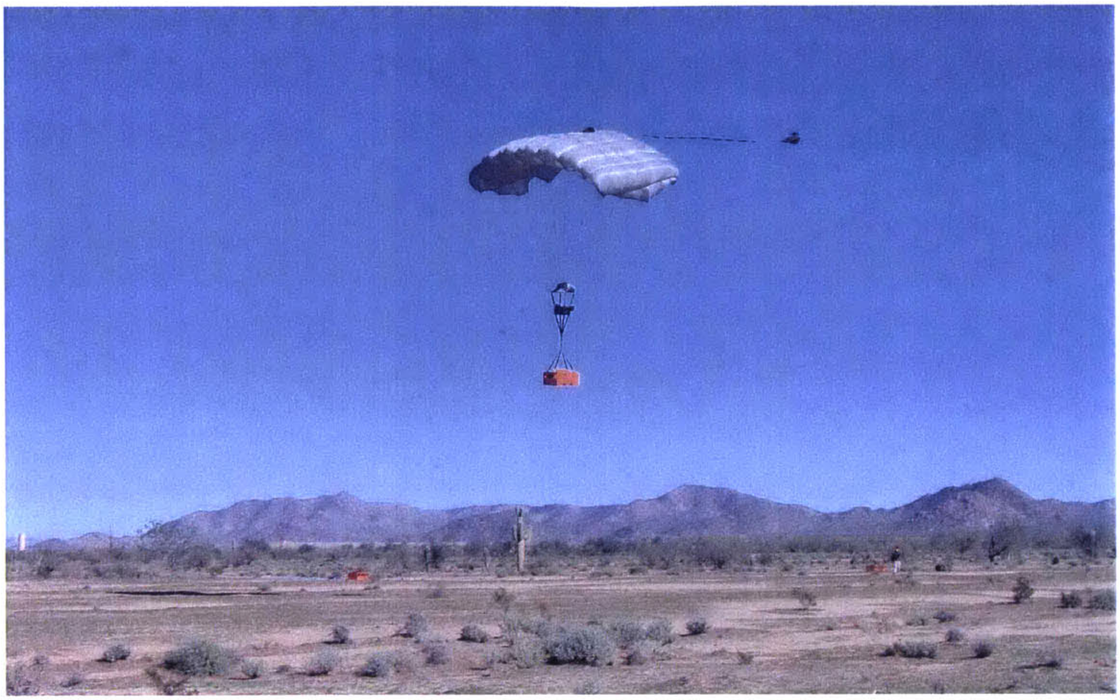


Figure 1-2: Autonomously guided *UltraFly* parafoil system (author photo)

cargo aircraft [2]. Although advancements in the calculation of the Computed Air Release Point (CARP) have since enabled unguided/ballistic airdrop from altitudes on the order of 18,000–25,000 feet, the average landing accuracy of these systems can be more than 850 feet, and may not be acceptable for certain delivery situations [12]. Today, guided parafoil applications include disaster relief and humanitarian aid, sensor delivery, battlefield support, resupply for remote, difficult to reach locations, and spacecraft landing operations, among other uses [13].

Given the additional control authority of parafoil systems, significant research effort has been spent in the development of advanced guidance algorithms for precision airdrop using onboard GPS receivers [14]. Two government funded programs sponsored by NASA and the U.S. Army began investigating guidance, navigation and control (GN&C) architectures for parafoil systems in the 1990s [11, 15]. The application of parafoils to military resupply helped to accelerate the design process, leading to improvements in atmospheric wind prediction, high-fidelity parafoil simulation models, and extensive flight test experiments [10, 16, 17]. Many of these developments took place as part of a combined initiative between the U.S. Army and the Air Force known as the Joint Precision Airdrop System (JPADS) [2, 18], along with predecessor programs such as New World Vistas (NWV) and the Precision Guided Airdrop System (PGAS) [16, 19]. The U.S. Army Natick Soldier Research, Development and Engineering Center (NSRDEC) continues to oversee JPADS research to enhance the capability of guided airdrop systems ranging from 10–42,000 pounds [20]. An example of an autonomously guided *UltraFly* [21] parafoil is shown in Figure 1-2 from the Ultra-Light Weight (JPADS-ULW) class of airdrop systems between 250–700 pounds.

The research conducted during these government programs has led to the design of a three phase flight profile for parafoil guidance that has since been adopted throughout the airdrop community [22–24]. This profile consists of a homing phase intended to steer the parafoil toward the target location, an energy management phase designed to descend over the target region, and a terminal guidance phase for performing the final approach maneuvers required for landing. Over the last decade, work in the field of autonomous parafoil guidance has primarily focused on improved strategies for the terminal phase, which is considered to be the most crucial for determining final landing accuracy [22, 25]. This thesis presents the development of robust, flexible planning algorithms for parafoil terminal guidance intended to extend the performance envelope of current airdrop systems.

## 1.1 Literature Review

Research on terminal guidance for autonomous resupply can be largely subdivided into two categories. The first category, glide-slope-based planning, utilizes the concept of the glide-slope surface or cone: the set of all position and heading states which, assuming constant velocity and disturbances, would guide the parafoil to the target location. Calise and Preston [26] utilize a series of scripted maneuvers online to estimate glide-slope parameters, then execute turning maneuvers to drive the parafoil to the glide-slope. This provides a useful approach trajectory, but the framework heavily constrains the solution space and requires long-term glide-slope tracking beginning from a large initial distance. Additionally, the presence of terrain obstacles is not considered during the path planning process. This makes the approach sensitive to uncertainty in both the vehicle dynamics and environment, especially given that the glide-slope surface shifts as a function of current wind conditions. Slegers et al. [27] track the glide-slope using nonlinear model predictive control (MPC), improving rejection of small-scale disturbances, but also requiring long-term glide-slope tracking. Bergeron et al. [28] use feedback control, known as Glide-slope Surface Guidance (GSG), to drive the approach to the goal based on the estimated glide-slope and wind conditions. This minimizes the effect of coupled system uncertainty and ensures a maximum heading deviation from the estimated wind direction. Recent efforts by Ward and Costello [14, 29] have also demonstrated the potential for improved glide-slope tracking through both online system identification, and by exploiting the longitudinal control coupling between incidence angle and symmetric brake deflection. To summarize, while the above approaches [14, 27–29] take some measures to account for the effect of wind uncertainty on the parafoil landing position, they offer no robustness to interaction with terrain obstacles, and are subject to the fundamental constraining of the solution space imposed by the glide-slope approach paradigm.

Trajectory-based approaches, on the other hand, generate arbitrary reference trajectories online to optimize a pre-specified cost function, utilizing various control strategies to track these trajectories. Gimadieva [30] formulates parafoil terminal guidance as an optimal control problem and establishes the necessary conditions for optimality, but the resulting approach lacks the computational efficiency needed for real-time implementation, thus making it unable to adjust for varying wind conditions and model uncertainties during flight. Cleminson [31] introduces a Dynamic Programming formulation for parafoil guidance by developing a lattice matrix of possible position and heading states. Paths are designed recursively in 2D through a sequence of state transitions between grid points using a finite set of available parafoil heading commands. However, this formulation faces challenges in scalability due to the multiplicative number of time steps, model dimensions, decision variables, and lattice points required for planning. The complexity associated with time-varying winds and arbitrary 3-Dimensional terrain must also be addressed. As a result, this approach cannot yet provide the computational efficiency required for online operation [31].

Rademacher et al. [13] presents a hybrid optimal control strategy involving both modified Dubins and minimum-control-energy trajectories. Under the assumption of deterministic wind, these planned trajectories are optimized to reach the target location at the final time, and with the desired terminal heading, while a replanning strategy is used to compensate for the effects of the wind uncertainty. Although this approach is capable of operating online, terrain obstacles are not considered during the trajectory optimization. For this reason, a feasible path to the target location cannot be guaranteed without the addition of more complicated environmental constraints.

The Band-Limited Guidance (BLG) algorithm [32] uses direct optimization via Nelder-Mead simplex search to minimize a cost function based on the predicted terminal vehicle state. BLG guarantees that control bandwidth constraints are satisfied to ensure accurate trajectory following, and its computational efficiency enables the use of online replanning, making it effective for many nominal wind and terrain conditions. However, BLG is limited in its starting altitude due to high dimensionality and optimization scalability, thus constraining mission flexibility. BLG incorporates no notion of wind variation in its planner, instead relying on reactive replanning to address unexpected wind effects. Additionally, its direct optimization technique does not consider the possibility of off-nominal, adverse terrain interactions caused by changing wind conditions, particularly on complex terrain maps.

The Inverse Dynamics in the Virtual Domain (IDVD) algorithm developed by Yakimenko and Slegers [4] utilizes inverse dynamics to connect the initial vehicle state to the target terminal state, while guaranteeing the terminal conditions of the nonlinear boundary value problem (BVP) are satisfied. While computationally efficient, this approach cannot guarantee satisfaction of control bandwidth constraints, requiring iteration in order to ensure the planned trajectory can be tracked by the controller. This method also relies on rapid, reactive replanning to offset uncertainties during execution, but assumes a constant wind during planning. Recent extensions to the IDVD algorithm [33] consider altitude-varying and/or higher-dimensional wind profiles during terminal descent, accounting for both cross-track winds and updrafts/downdrafts. For each of these formulations, however, the BVP assumes a known, deterministic wind profile and robustness to future wind variations is captured only through replanning, rather than explicit modeling. Additionally, the effect of terrain geometry is not considered during the design of feasible descent trajectories.

Subsequent work by Rogers and Slegers considers robustness to wind variations by utilizing graphics processing units (GPUs) to parallelize a Monte Carlo simulation of possible future winds, and the resulting parafoil trajectories, based on available measurements [3, 34]. However, significant computational effort is required to run these Monte Carlo simulations online, leading to issues of scalability for high-dimensional systems [35, 36]. Within each set of simulations, the solution space is restricted to a limited number of candidate solutions of the original BVP, where each candidate assumes a constant-rate turn and terminal heading constraint, and is simulated over a set of constant wind profiles. Based on these assumptions, online replanning is used to correct for the effect of future wind disturbances. While such an approach effectively incorporates the overall, trajectory-wide wind effect, each simulation assumes a deterministic wind. As a result, this method does not model the possibility of dynamic wind changes during the planning process, potentially making it overly optimistic. Although the work in [3, 34] considers the presence of environmental obstacles in the vicinity of the target, terminal guidance is also assumed to begin in relative proximity to the target location due to the selected parameterization of candidate trajectories (i.e., using a single constant-rate turn and straight line segment). This approach may therefore prove difficult to implement in constrained terrain geometries, such as canyons and valleys, where robust planning and obstacle avoidance must begin from high initial altitudes, and greater path flexibility is required.

Recent work by Fowler and Rogers considers the use of Bezier curves to perform optimized path planning for a small parafoil in three-dimensional obstacle fields [37]. This method offers geometric flexibility to online trajectory planning by adjoining multiple cubic Bezier curves, which are used to navigate constrained terrain environments. However, the computation of the proposed optimization is shown to scale poorly with the number of degrees of freedom (i.e., the control points of the adjoined Bezier curves) and convergence is sensitive to the initial guess solution. These factors may limit the effectiveness of this approach for environments of increased complexity and/or initial altitude. In addition, path feasibility and terrain collisions are only assessed for the nominal planned trajectory under the assumption of mean wind, with reactive replanning used to mitigate the effects of future wind disturbances. Because some replanning iterations may require significant computation time in order to converge, this approach can render the parafoil vulnerable to terrain collisions due to uncompensated wind effects. In these situations, safety in complex terrain environments cannot be guaranteed.

Due to the challenges induced by time-varying and uncertain wind conditions, ongoing research efforts have focused on a variety of strategies to improve parafoil guidance performance subject to dynamic wind environments. An overview of several approaches is presented in the recent work by Ward et al. [38]. One particularly effective technique is the implementation of ground-based wind measurements, which can be recorded using LIDAR or an anemometer, and communicated in real-time to the descending parafoil system [38–40]. The addition of wind knowledge in the vicinity of the drop zone has been shown to produce enhanced landing accuracy and parafoil survivability by improving the estimate of the future wind profile used during planning [39].

Work by Chiel and Dever [41] explore methods for high wind parafoil guidance beginning from high initial altitudes and large target offsets. This work results in two homing style algorithms designed to maintain target attainability and landing accuracy in both strong tailwinds and shifting wind conditions. Additionally, several recent parafoil guidance strategies consider planning in a wind fixed reference frame [13, 20, 24, 31, 38, 41] so as to decouple the trajectory design from the anticipated drift due to atmospheric winds. The wind fixed coordinate system provides a convenient planning framework for the most commonly considered airdrop scenario involving flat terrain, but may not be as effective for parafoil guidance in the presence of environmental obstacles. For complicated terrain scenarios, constraint checks are most efficiently performed with respect to the inertial reference frame [3, 32, 37].

Lastly, advances in parafoil longitudinal control have recently been achieved using a variety of techniques such as symmetric brake deflection [28], incidence angle control [27, 42], payload weight shifting [43], and bleed-air spoilers [44]. This additional degree of control authority can enable the parafoil to adjust both glide-slope and airspeed so as to compensate for possible wind disturbances and modeling errors during flight. While all of these developments in the field of parafoil terminal guidance [27, 28, 38–44] have the potential to create valuable performance improvements, none of these research topics have addressed the problem of wind uncertainty modeling or robust parafoil guidance in complex terrain scenarios.

In summary, the general body of parafoil terminal guidance algorithms is subject to some or all of the following limitations:

1. An artificially-constrained solution space, often based on preconceived notions of the solution form;

2. Implicit or explicit constraints on the initial altitude, which require a prior descent phase to bring the parafoil to initial conditions suitable for successful terminal guidance;

3. An assumed obstacle free/unconstrained terrain environment; and/or

4. A reactive approach to handling the effect of wind uncertainty.

The purpose of this thesis is to address these limitations, leading to the development of two robust parafoil terminal guidance strategies that directly consider the effects of dynamic wind disturbances during the trajectory planning process. Robustness is achieved through explicit modeling of the future wind uncertainty, while particular emphasis is placed on computational efficiency for online trajectory planning using currently available flight hardware. The proposed terminal guidance algorithms will be shown to operate successfully from high initial altitudes and large target offsets, and to scale favorably under such conditions. Finally, these algorithms provide for the design of flexible trajectory shapes of near arbitrary complexity, enabling successful parafoil guidance in challenging, highly constrained terrain environments.

## 1.2 Overview of the BLG Algorithm

Throughout this thesis, the Band-Limited Guidance (BLG) algorithm [32] is selected as a baseline for comparison, representing one of several state-of-the-art approaches for parafoil terminal guidance. This section provides a brief overview of the key BLG algorithm components.

During parafoil terminal guidance, BLG determines an optimized control input by choosing coefficients $\psi_k$ for the heading rate profile,

$$\psi'(z) = \sum_{k=0}^{N} \psi_k \frac{\sin(\pi(p_z - k\Delta h)/\Delta h)}{\pi(p_z - k\Delta h)/\Delta h}, \tag{1.1}$$

based on simulating forward the simplified parafoil kinematics,

$$\begin{aligned} p_x' &= -L_D \cos(\psi) + w_x/\dot{p_z}, \quad \text{and} \quad p_y' = -L_D \sin(\psi) + w_y/\dot{p_z}, \\ (\cos(\psi))' &= -\psi(p_z)' \sin(\psi), \quad \text{and} \quad (\sin(\psi))' = \psi(p_z)' \cos(\psi), \end{aligned} \tag{1.2}$$

where $(\cdot)'$ denotes a derivative with respect to altitude $p_z$ [32]. BLG formulates terminal guidance as an unconstrained optimization problem, designed to minimize the cost function

$$J_{BLG} = w_1(\Delta p_x^2 + \Delta p_y^2) + w_2(\sin(\Delta\psi/2))^2 \tag{1.3}$$

via the propagation of (1.1)–(1.2), where $\Delta p_x^2$ and $\Delta p_y^2$ are squared miss distances, and $\Delta\psi$ is the difference between the final heading and desired heading at the terminal trajectory state [4, 32]. The terms $w_1$ and $w_2$ in (1.3) denote user specified weights selected to penalize the landing error for position and heading, respectively. By penalizing the miss distance of the terminal trajectory state, BLG implicitly considers the presence of terrain obstacles via the cost associated with prematurely terminating the trajectory before reaching the target location.

The BLG optimization is solved repeatedly online using the Nelder-Mead simplex algorithm, while the integration of the kinematics is performed using fixed point arithmetic for computational efficiency [32]. In addition, BLG periodically compares the current optimization cost against a set of randomly generated trajectory solutions to prevent possible convergence to local minima [20]. Lastly, through the selection of appropriate values for $N$ and $\Delta h$ in (1.1), the BLG algorithm ensures accurate trajectory tracking by considering only those heading rate profiles with frequencies sufficiently less than the control bandwidth constraints. These parameters serve to enforce the "Band-Limited" quality of the trajectory design so as to avoid excitation of payload and canopy modes [32].

## 1.3 Contributions and Content Overview

This thesis presents contributions towards the design, development, and analysis of two online trajectory planning algorithms for autonomous parafoil guidance in complex terrain and wind environments. These contributions are outlined below.

- **Chapter 2:** This chapter presents the parafoil terminal guidance algorithm known as Analytic CC-RRT, which builds upon the framework of chance-constrained rapidly-exploring random trees (CC-RRT) [45]. By extending prior work [46, 47], a novel multi-class wind uncertainty model is presented using observed wind data to classify and anticipate the future wind environment online. From this wind model, the analytic *a priori* uncertainty distribution is derived over future parafoil trajectories. A method of analytic uncertainty sampling is then introduced to efficiently evaluate the probability of constraint violation against mapped terrain. Additionally, prior work in [46] is extended to consider parafoil landing speed penalties within the Analytic CC-RRT cost-to-go function used for trajectory selection. Extensive analysis demonstrates the effectiveness of Analytic CC-RRT relative to the state-of-the-art BLG algorithm over a series of Monte Carlo simulation experiments in challenging wind and terrain scenarios.

- **Chapter 3:** The design of a new parafoil terminal guidance algorithm is presented known as Chance-Constrained Band-Limited Guidance (CC-BLG). This algorithm incorporates the wind uncertainty model and analytic sampling technique outlined in Chapter 2 into the optimized BLG trajectory planning framework. Motivations are introduced for this alternative guidance strategy, followed by a description of the three core algorithm components. First, a method of weighted analytic uncertainty sampling is proposed to refine the estimated probability of constraint violation using the wind uncertainty model. Next, a novel risk-based objective function is developed for trajectory optimization. This objective function enables the planner to efficiently balance the parafoil performance metrics of landing accuracy and landing speed, with the risk of off-nominal terrain collisions caused by future wind disturbances. Finally, the principles of discrete reachability set approximation presented in Chapter 2 are applied to the CC-BLG algorithm for robust obstacle detection, and avoidance, during fixed-horizon trajectory planning in constrained terrain environments.

- **Chapter 4:** A detailed design analysis is provided for each component of the CC-BLG algorithm developed in Chapter 3. Analysis is conducted through several Monte Carlo simulation experiments using realistic valley terrain, and Draper Laboratory's high fidelity nonlinear parafoil simulator [48]. Specific parameter settings for the CC-BLG objective function and reachability set approximation are evaluated, leading to the selection of a final configuration. The reachability set approximation and cost-to-go function are demonstrated in simulation to maintain robust trajectory feasibility over future planning horizons. This chapter also considers additional algorithm refinements and alternatives for both the analytic chance constraints, and method of uncertainty sampling developed in Chapter 2.

- **Chapter 5:** This chapter presents Monte Carlo simulation results demonstrating the effectiveness of the CC-BLG algorithm developed in Chapter 4 for robust parafoil guidance and trajectory optimization. Simulations are used to compare the performance of CC-BLG alongside the nominal BLG algorithm in three difficult real-world terrain environments. This chapter also evaluates BLG and CC-BLG performance in flat, obstacle free terrain, and demonstrates the ability of the CC-BLG algorithm to satisfy terminal heading constraints for reduced landing speed.

- **Chapter 6:** Flight test results are presented for the CC-BLG algorithm using a full-scale parafoil system. Modifications are first introduced to the CC-BLG algorithm in order to satisfy the computational limitations of available flight hardware. These modifications are shown to retain CC-BLG robustness properties while significantly reducing online computation. Flight test results from 9 parafoil drop experiments demonstrate the successful operation of the CC-BLG algorithm for robust planning in the presence of simulated environmental obstacles. Analysis is presented from recorded CC-BLG test data and compared against Monte Carlo simulations of BLG and CC-BLG performance subject to recreated flight test conditions.

- **Chapter 7:** This chapter provides conclusions and recommendations for future work.

# Chapter 2

# Chance-Constrained Rapidly-exploring Random Trees (CC-RRT)

## 2.1 Introduction

This chapter presents a real-time trajectory planning algorithm known as Analytic CC-RRT which enables a large, autonomous parafoil to robustly execute collision avoidance and precision landing on mapped terrain, even in the presence of significant wind uncertainty. Analytic CC-RRT builds upon the framework of chance-constrained rapidly-exploring random trees (CC-RRT) for robust motion planning in cluttered, non-convex environments [45]. In this way, the proposed parafoil guidance strategy incorporates the benefits of sampling-based planners and particularly rapidly-exploring random trees (RRT) [49] (e.g. incremental construction, trajectory-wise constraint checking, rapid exploration, dynamically feasible trajectories), while using chance constraints to ensure probabilistic feasibility within guaranteed, user-specified bounds. Through trajectory-wise constraint checking, Analytic CC-RRT can efficiently evaluate the risk of constraint violation online due to multiple sources of both internal and external uncertainty.

In collaboration with Luders and Sugel [46, 47], three primary contributions of the Analytic CC-RRT algorithm are introduced for robust parafoil guidance in complex terrain and wind environments. First, a novel wind uncertainty model is presented, using real-time observed wind data. The resulting model is shown to accurately represent true wind behavior, while adjusting the conservatism of the guidance algorithm to reflect prevailing conditions. Second, this multi-class wind model is utilized to derive the analytic *a priori* uncertainty distribution over future parafoil trajectories. This is leveraged through a novel variation of the CC-RRT path planner, which performs analytic sampling of the uncertainty distributions to ensure robust avoidance of undesirable collisions with arbitrary, potentially aggressive terrain maps. Finally, the relative value of paths is assessed via a novel terminal cost-to-go function, which utilizes a fixed-horizon discrete approximation of the parafoil reachability set. This enables selection of partial paths from any altitude that intelligently trade off between current and reachable future states.

Simulation results in Section 2.7 reveal the effectiveness of each of these components, and demonstrate that the full parafoil Analytic CC-RRT algorithm can achieve superior landing accuracy in both average-case and worst-case performance relative to state-of-the-art algorithms such as Band-Limited Guidance (BLG) [32]. In particular, this chapter demonstrates that the analytic-sampling approach achieves higher robustness to wind uncertainty than replanning and/or mean-wind estimation alone. The Analytic CC-RRT planner is also shown to be largely invariant to changes in both altitude and terrain. Building on previous work [46], results in Section 2.7.2 indicate that Analytic CC-RRT is capable of operating in real-time while preserving these robustness properties. Lastly, this work extends previous developments [46] to consider terminal heading constraints, by incorporating landing speed penalties into the cost-to-go function (Section 2.6) to encourage upwind landings.

## 2.2 Problem Overview

The terminal guidance paradigm typically utilizes a combination of a homing phase, designed to steer the parafoil directly toward the target, and an energy management phase, designed to descend the parafoil above the target, until an appropriate altitude is reached for terminal guidance [4, 32, 50]. Such algorithms generally assume that terminal guidance will begin in relative proximity to the target location in both lateral distance and altitude. Although the approach presented in this work will often operate under similar conditions, such assumptions are not necessary.

The vehicle state is represented as $\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \psi & \mathbf{s}^T \end{bmatrix}^T$, where $\mathbf{p} = (p_x, p_y, p_z)$ is the position in the inertial reference frame, $\psi$ is the heading, and $\mathbf{s}$ represents a vector of any additional states needed to characterize the parafoil's motion—in this case, the lag dynamics. In the terminal guidance problem, the objective is to guide the parafoil from some initial position $\mathbf{p}_I$ and heading $\psi_I$ (full state $\mathbf{x}_I$) to some target location $\mathbf{p}_G$ (full state $\mathbf{x}_G$). The parafoil dynamics are represented as the nonlinear state-space system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w}), \quad \mathbf{x}(t_I) = \mathbf{x}_I, \tag{2.1}$$

where $t_I$ is the initial time, $\mathbf{u}$ are the control inputs and $\mathbf{w} = (w_x, w_y, w_z)$ are the wind disturbances.

The wind disturbances are unknown at current and future times; denote the most recent wind observation as $\mathbf{w}_I$ (if none have been taken, a prior value from forecasting data may be applied, or simply assume $\mathbf{w}_I = 0$). In this work, we choose to represent the wind disturbances using the generalized model

$$\dot{\mathbf{w}} = f_w(\mathbf{w}, \overline{\mathbf{w}}, \mathbf{v}), \quad \mathbf{w}(t_I) = \mathbf{w}_I, \tag{2.2}$$

where $\overline{\mathbf{w}}$ is an estimate of the mean wind, assumed to be available to the planner, and $\mathbf{v}$ is unknown model noise. The wind model developed for this work is derived in Section 2.3.

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

$$\begin{aligned}
\min_{\mathbf{u}} \quad & \phi_f(\widehat{\mathbf{x}}(t_f), \mathbf{x}_G) + \int_{t_I}^{t_F} \phi(\widehat{\mathbf{x}}, \mathbf{x}_G) && (2.3)\\
\text{s.t.} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w}), \;\; \mathbf{x}(t_I) = \mathbf{x}_I, && (2.4)\\
& \dot{\widehat{\mathbf{x}}} = f(\widehat{\mathbf{x}}, \mathbf{u}, \widehat{\mathbf{w}}), \;\; \widehat{\mathbf{x}}(t_I) = \mathbf{x}_I, && (2.5)\\
& \dot{\mathbf{w}} = f_w(\mathbf{w}, \overline{\mathbf{w}}, \mathbf{v}), \;\; \mathbf{w}(t_I) = \mathbf{w}_I, && (2.6)\\
& \dot{\widehat{\mathbf{w}}} = f_w(\widehat{\mathbf{w}}, \overline{\mathbf{w}}, 0), \;\; \widehat{\mathbf{w}}(t_I) = \mathbf{w}_I, && (2.7)\\
& \mathbf{u} \in \mathcal{U} \;\; \forall t, && (2.8)\\
& \mathbb{P}_{\mathbf{v}}\left(\mathbf{x} \in \mathcal{X}\right) \geq p_{\text{safe}}. && (2.9)
\end{aligned}$$

The parafoil state $\widehat{\mathbf{x}}$ and wind state $\widehat{\mathbf{w}}$ evolve according to (2.1) and (2.2) respectively assuming $\mathbf{v} \equiv 0$, representing deterministic, nominal propagation of the dynamics under the assumption of constant wind $\overline{\mathbf{w}}$. This is utilized simply to ensure that the objective (2.3) being optimized is deterministic, though stochastic forms may be used. In practice, the optimization (2.3) is solved repeatedly during the descent, with $\mathbf{x}_I$ and $\mathbf{w}_I$ being set to the most recent state and wind measurements, respectively, at current time $t_I$.

The sets $\mathcal{U}$ and $\mathcal{X}$ represent constraints on the input and state, respectively. The state constraints $\mathcal{X}$ must be *probabilistically* satisfied, i.e. satisfied with probability of at least $p_{\text{safe}}$ over all possible $\mathbf{v}$, as represented in (2.9) by $\mathbb{P}_{\mathbf{v}}$. These constraints include the terrain map $T(p_x, p_y)$, which is assumed to be perfectly known; the terminal time $t_F$ is the time at which $p_z \leq T(p_x, p_y)$. Additional state constraints may be included, such as internal state bounds or no-fly zones, though this is not explored further in this work. The stochastic elements of this optimization manifest themselves only in the final chance constraint (2.9); Section 2.4 details the implementation of this chance constraint, yielding a deterministic optimization [51].

### 2.2.1 Parafoil Model

The parafoil is modeled as a Dubins vehicle [52] descending at a rate governed by atmospheric conditions subject to updrafts/downdrafts, with the input-to-heading-rate mapping governed by complex lag dynamics. The lack of altitude control, coupled with a large minimum turning radius and slow turning rate, necessitates significant advance planning for precision guidance and landing. This is exacerbated by the presence of heavy winds, which can lead to loss of goal reachability and/or premature terrain collisions if not properly anticipated.

The parafoil velocity $v(p_z)$ is assumed to be a function of the vehicle altitude $p_z$, via [13, 53]

$$v(p_z) = v_0 e^{p_z/2\tau_z}, \tag{2.10}$$

where $\tau_z = 10^4$ m, and $v_0$ is the nominal vehicle velocity at sea level. In this chapter, we adopt the 10,000-pound Dragonfly parafoil used by Carter et al. [22], with $v_0 =$ 17.8 m/s and lift-to-drag ratio $L_D = 2.8$.

The heading rate of the parafoil is modeled as a second-order approximation of the canopy Dutch roll lateral mode; our specific model selects time constant $\tau = 11.5$ s and damping ratio $\zeta = 0.5$ as suggested by Carter et al. [22]. A first-order lag is used to model the differential toggle control input mechanism with $\tau = 5$ s [48], while the controller is a PID with feedforward gains tuned to achieve comparable performance [22]. In total, this yields a 5th order state $\mathbf{s}$ and dynamics $(A, B, C, D)$, augmented to the state vector $\mathbf{x}$ and dynamics (2.1), respectively. The control input is a scalar, $\mathbf{u} \equiv u \equiv \dot{\psi}_d$, representing the desired heading rate, subject to the symmetric input bounds $\mathcal{U} = \{u \mid |u| \leq \omega_{\max}\}$. The overall parafoil dynamics (2.1) thus take the form

$$\dot{p}_x = v(p_z)\cos\psi + w_x, \tag{2.11}$$

$$\dot{p}_y = v(p_z)\sin\psi + w_y, \tag{2.12}$$

$$\dot{p}_z = \frac{-v(p_z)}{L_D} + w_z, \tag{2.13}$$

$$\dot{\mathbf{s}} = A\mathbf{s} + Bu, \tag{2.14}$$

$$\dot{\psi} = \text{sat}(C\mathbf{s} + Du, -\omega_{\max}, \omega_{\max}), \tag{2.15}$$

where the saturation function $\text{sat}(a, b, c)$ bounds $a$ between $b$ and $c$, and $\omega_{\max} = \pi/15 = 0.2094$ rad/s [48] (such that the vehicle's minimum turning radius $R_{\min}$ equals $v_0/\omega_{\max} = 85$ m). In this formulation, only the position states $\mathbf{p}$ are affected by the wind disturbance uncertainty $\mathbf{w}$, including possible effects on altitude $p_z$ by updrafts and downdrafts via $w_z$. Within the planning framework, which operates in discrete time, the optimization (2.3) is discretized with time step $dt = 0.1$ s.

## 2.3 Real-Time Wind Modeling

The wind model detailed in this section is utilized by the planner to improve prediction accuracy and robustness for the parafoil terminal guidance problem. The development of this wind model is based on satisfying three main objectives. First, the wind model should improve predictability of future wind effects. Improved predictability, especially in scenarios where there is significant prevailing wind, can mitigate the amount of replanning needed and improve the quality of solutions provided by the proposed algorithm. Second, the wind model should capture the uncertainty of future wind effects, giving the planner knowledge of a distribution over possible outcomes of a planned trajectory. Characterizing and utilizing such an uncertainty distribution in a probabilistic framework (Section 2.4) strengthens planner robustness to terrain obstacles. Finally, the wind model should be kept simple, to maintain real-time planner operation and discourage data overfitting.

Given the importance of wind modeling in many engineering applications, there has been considerable work on developing wind prediction and estimation models, including the case of online estimation [54–57]. However, as described in [46], none of these modeling approaches [54–57] address the wind prediction problem over the short timescales and limited datasets inherent in parafoil precision guidance. This section fits an uncertainty model to the wind which can be incorporated into the planner to enforce robustness. This approach includes online learning to determine, in real time, the class of wind scenario being experienced by the parafoil and the corresponding parameters of the variational estimate associated with each class. Each model is tuned to capture the amount of uncertainty typical to wind profiles within its corresponding class. In this manner, the level of conservatism in the planner can be adjusted online to reflect the wind conditions being observed. Draper Laboratory has released 194 altitude-dependent wind profiles from parafoil drops [46], collected using the sensor configuration and estimation procedure outlined in work by Carter et al. [32]. These wind profiles are used as training data during the development of the wind model presented in this work.

### 2.3.1 Model Form

The wind model is assumed to take the form (2.2), written in discrete time as

$$\mathbf{w}_{t+1} = f_w(\mathbf{w}_t, \overline{\mathbf{w}}, \mathbf{v}_t), \quad \mathbf{w}_0 = \mathbf{w}_I, \tag{2.16}$$

where time step 0 occurs at system time $t_I$. The 3-D wind estimate at time step $t$, $\mathbf{w}_t$, is assumed to take the form

$$\mathbf{w}_t = \overline{\mathbf{w}} + \delta\mathbf{w}_t, \tag{2.17}$$

comprising the sum of a 3-D persistent estimate $\overline{\mathbf{w}}$ and a 2-D variational estimate $\delta\mathbf{w}_t$.

The persistent estimate $\overline{\mathbf{w}}$ reflects the notion that there typically exists a prevailing wind which acts on the parafoil throughout the entire mission, and must be accounted for during the state prediction. It is represented using a finite impulse response filter,

$$\overline{\mathbf{w}} = \frac{1}{m} \sum_{i=t_I-m-1}^{t_I} \mathbf{w}_i, \tag{2.18}$$

where $m$ is the filter window width.

The filter width $m$ is chosen by optimizing a metric representing the filter predictive accuracy [47]. Consider propagating the parafoil dynamics from some initial state $\mathbf{p}_I$ to the ground, assuming zero input ($u \equiv 0$), no lag dynamics ($\mathbf{s} \equiv 0$), and flat terrain ($T(p_x, p_y) \equiv 0$). Additionally, assume that previous observations of the wind profile have been observed prior to the parafoil reaching $\mathbf{p}_I$, such that the filter (2.18) can be applied in full. For each available 3D wind profile, the dynamics are propagated from the same initial state and observations. For the $w$th wind profile, three possible landing positions are of interest:

- The landing position under the true wind, $\mathbf{p}_T^{(w)}$;
- The landing position under zero wind, $\mathbf{p}_0^{(w)}$; and
- The landing position under constant wind using (2.18) with width $m$, $\mathbf{p}_m^{(w)}$.

Define the quantity

$$\delta d_m^{(w)} = \left\| \mathbf{p}_0^{(w)} - \mathbf{p}_T^{(w)} \right\| - \left\| \mathbf{p}_m^{(w)} - \mathbf{p}_T^{(w)} \right\|, \tag{2.19}$$

which takes the difference in accuracy between the zero-wind model and the impulse-filter model in predicting the true landing position. For those wind profiles in which prediction accuracy degrades with the impulse-filter model, $\delta d_m^{(w)} < 0$, denote $D_m = \{\delta d_m^{(w)} \mid \delta d_m^{(w)} < 0\}$. The filter width is then chosen as

$$m = \underset{m>0}{\operatorname{argmax}} \left\{ \min(D_m) + \beta \operatorname{mean}(D_m) - \lambda m \right\}, \tag{2.20}$$

where $\beta, \lambda > 0$; in this work, $\beta = 2$ and $\lambda = 1$. This cost function includes terms for the worst-case and average-case accuracy in $D_m$, as well as a regularization term [47].

The variational estimate $\delta \mathbf{w}_t$ is represented as multi-modal linear dynamics subject to Gaussian noise,

$$\delta \mathbf{w}_{t+1} = (I + dt A_c)\delta \mathbf{w}_t + dt B_c \mathbf{v}_t, \quad c \in \{1, \ldots, N_C\}, \tag{2.21}$$

where $N_C$ is the number of modes/classifications used, $A_c$ and $B_c$ are the tuned matrices used for the $c$th wind classification (Section 2.3.2), and $\mathbf{v}_t \in \mathcal{N}(0,1)$, i.e. zero-mean, unit-variance Gaussian noise. This colored noise process reflects the idea that, while wind at lower altitudes is correlated with the wind measured at the current altitude, this correlation tends to degrade with increasing separation.

By substituting (2.17) into (2.21), the wind model function (2.16) can be written as

$$f_w(\mathbf{w}_t, \overline{\mathbf{w}}, \mathbf{v}_t) = \overline{\mathbf{w}} + (I + dtA_c)(\mathbf{w}_t - \overline{\mathbf{w}}) + dtB_c\mathbf{v}_t, \quad c \in \{1, \ldots, N_C\}. \tag{2.22}$$

The remaining questions, then, are (i) how to identify an appropriate number of classifications $N_C$ and the corresponding wind model dynamics $(A_c, B_c)$, $c \in \{1, \ldots, N_C\}$ for each, and (ii) how to select the appropriate classification online. These topics are discussed next.

### 2.3.2 Wind Model Training

The wind profiles used for training parameterize each component of the measured wind velocity vector as a function of altitude, i.e. $\{w_x(p_z), w_y(p_z), w_z(p_z)\}$, over a set of altitude data points. This can pose problems for clustering and classification algorithms, which are typically designed to operate on observations, rather than functions. By using feature selection, the dimension of the system model can be reduced, allowing for the use of efficient clustering and classification schemes [58]. For each data point, denote $\rho = \sqrt{w_x^2 + w_y^2 + w_z^2}$ and $\theta = \text{atan2}(w_y, w_x)$; for this work, we use the feature vector

$$\Phi = \begin{bmatrix} \text{mean}(\rho) & \max(\rho) & \text{mean}\left(\frac{d\rho}{dp_z}\right) & \max\left(\frac{d\rho}{dp_z}\right) & \text{mean}\left(\frac{d\theta}{dp_z}\right) & \max\left(\frac{d\theta}{dp_z}\right) \end{bmatrix} \tag{2.23}$$

For each wind profile, this feature vector takes the mean and maximum value over all data points of three quantities: the wind magnitude, the rate of change of wind magnitude, and the rate of change of wind direction. Collectively, these features were chosen to represent the power and variability inherent in each profile.

The objective is then to use the feature-based representation of each wind profile (2.23), denoted for the $w$th wind profile as $x_w$ below, to classify the $N_W$ wind profiles into $N_C$ classes. We represent each possible disjoint partition of these profiles as $\mathbf{S} = \{S_1, S_2, \ldots, S_{N_C}\}$. The partition is chosen so as to minimize the sum of squared distances to the mean within each cluster, $\mu_i$, such that

$$\mathbf{S}^\star = \underset{\mathbf{S}}{\operatorname{argmin}} \left( \sum_{i=1}^{N_C} \sum_{x_w \in S_i} \|x_w - \mu_i\|^2 \right) + \lambda k, \tag{2.24}$$

with the last term $\lambda k$ being included for regularization. This optimization is solved using the DP-means algorithm [59], which extends $k$-means clustering [60] such that the appropriate number of clusters $N_C$ can be incrementally identified, rather than assumed *a priori*. During the DP-means assignment step, if an observation is further than $\lambda$ from the nearest cluster center, a new cluster is added with its center defined as the observation $x_w$ which created it. Using the aforementioned Draper wind profiles [46], three distinct classes were identified.

For each classification, the variational wind model dynamics $(A_c, B_c)$ are constructed by matching the analytic uncertainty distribution to the empirical distribution identified from the wind profiles. To simplify uncertainty sampling (Section 2.4), the variational wind model is constructed to be two-dimensional, i.e. $\delta w_z = 0$; observed updrafts and downdrafts are still incorporated via the mean wind $\overline{\mathbf{w}}$. We further assume that $\delta w_x$ and $\delta w_y$ are independent and symmetric, such that the variational wind model dynamics $A_c \in \mathbb{R}^{3\times 3}$ and $B_c \in \mathbb{R}^{3\times 2}$ can be written as

$$A_c = \alpha_c \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_c = \beta_c \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \tag{2.25}$$

where $\alpha_c, \beta_c \in \mathbb{R}$.

For each wind profile in the cluster, compute the miss distances $\left\|\mathbf{p}_m^{(w)} - \mathbf{p}_T^{(w)}\right\|$ used in (2.19). Over these wind profiles, let $d_i$ denote the $i$th largest miss distance, and $n_i$ the fraction of profiles with a miss distance less than or equal to $d_i$. These characteristics of the cumulative density function (CDF) will be compared against the analytic wind model for tuning, as described next.

Represent the wind model (2.21) in continuous-time form as

$$\delta\dot{\mathbf{w}} = A_c \delta\mathbf{w} + B_c \mathbf{v}, \tag{2.26}$$

and define the position variation $\delta\mathbf{p} = \begin{bmatrix} p_x - E[p_x] & p_y - E[p_y] & p_z - E[p_z] \end{bmatrix}^T$. We can then construct the augmented dynamics

$$\begin{bmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{w}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0_3 & I_3 \\ \mathbf{0}_3 & A_c \end{bmatrix}}_{A_{\text{aug}}} \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{w} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 2} \\ B_c \end{bmatrix}}_{B_{\text{aug}}} \mathbf{v}. \tag{2.27}$$

The covariance at impact time $t_F$, $\Sigma_F \equiv \Sigma(t_F)$, can be propagated forward using the dynamics

$$\dot{\Sigma} \;=\; A_{\text{aug}}\Sigma + \Sigma A_{\text{aug}}^T + B_{\text{aug}} B_{\text{aug}}^T, \;\; \Sigma(t_I) = 0. \tag{2.28}$$

For comparison with empirical data, the lateral position covariance is isolated via

$$\begin{aligned} \Sigma_F' &= C_T \Sigma_F C_T^T, && (2.29) \\ C_T &= \begin{bmatrix} I_2 & 0_{2\times 4} \end{bmatrix}. && (2.30) \end{aligned}$$

Given the independence and symmetry assumptions on $\delta w_x$ and $\delta w_y$, $\Sigma_F'$ can be written as $\Sigma_F' = \sigma^2 I_2$, where $\sigma > 0$ is a scalar. This represents a chi distribution on landing miss distances with standard deviation $\sigma$; denote its CDF for dynamics $(A_c, B_c)$ as $\chi(x, A_c, B_c)$. This CDF can be matched directly to the empirical wind profile CDF with characteristics $(d_i, n_i)$ as described above. For the $c$th wind classification, the dynamics $(A_c, B_c)$ are identified by minimizing the root mean square error between the two CDFs,

$$(A_c, B_c) = \underset{(A,B)}{\operatorname{argmin}} \sum_{i \in S_c} (n_i - \chi(d_i, A, B))^2 . \tag{2.31}$$

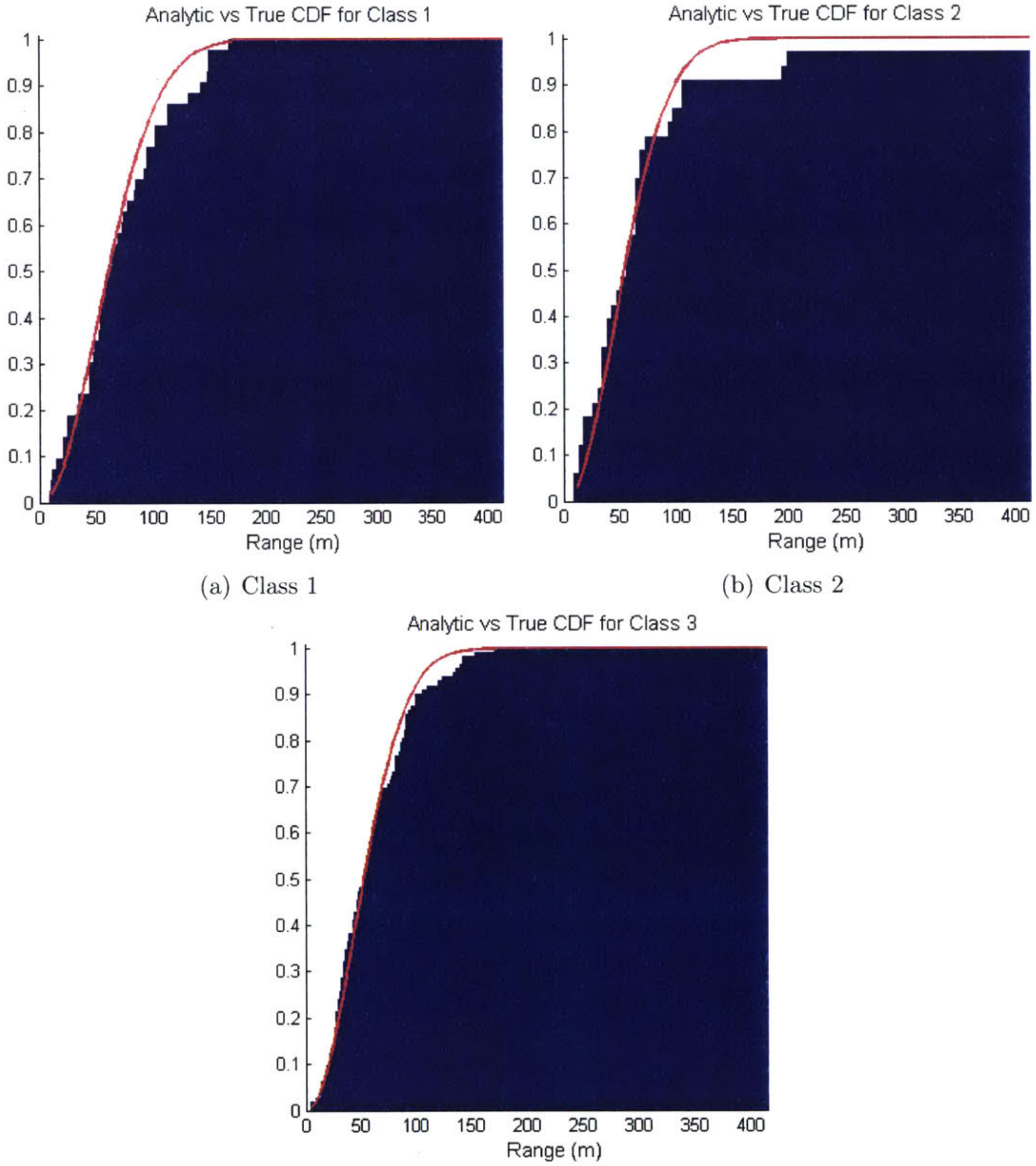
Figure 2-1 compares the analytic and "true" (empirical) CDFs for each of the classes identified from the Draper wind profiles.

### 2.3.3 Online Classification Selection

In order to utilize the varying levels of uncertainty associated with the $N_C$ classifications identified in Section 2.3.2, the planner uses support vector machines (SVM) [61] to classify each wind estimate observed by the vehicle. For each of the $N_C$ classes, the planner generates an SVM binary inclusion classifier, which can be used to identify if the wind estimates being received are a member of a particular class. Online, the trained SVM classifier compares the feature vector produced by the most recent wind observations (2.23) against the $N_C - 1$ hyperplanes separating the members of the $N_C$ wind classes. Each hyperplane determines in sequence whether the vector $\Phi$ does or does not belong to Class X. In the case of Figure 2-1, any feature vector which fails the binary inclusion after comparison against the dividing Class 1 and Class 2 hyperplanes is assigned to the remaining Class 3 cluster.

## 2.4 Analytic Uncertainty Sampling

This section presents a novel framework for modeling future uncertainty in trajectory predictions, based on CC-RRT, such that robustness to possible future variation in disturbances can be achieved. Recall that in the formulation of the parafoil terminal guidance problem (Section 2.2), satisfaction of state constraints is specified via the chance constraint (2.9). This represents a minimum likelihood that all state constraints, here consisting of the terrain surface $p_z \geq T(p_x, p_y)$, be satisfied with a minimum probability of $p_{\text{safe}}$ along each trajectory. In the CC-RRT algorithm, the chance constraint (2.9) must be satisfied at each time step and is converted to a tight-



(a) Class 1

(b) Class 2

(c) Class 3

Figure 2-1: Analytic (red) and empirical (blue) miss distance CDFs for wind profiles in each class

ened, deterministic constraint [45]. Under the assumptions of linear dynamics and Gaussian noise, these tightened constraints are shown to guarantee probabilistic feasibility to polyhedral constraints at each time step. Furthermore, due to CC-RRT's trajectory-wise constraint checking, a risk bound can be explicitly computed at each time step online against each uncertainty source.

It is shown here that, though the parafoil dynamics are nonlinear, the effect of the wind uncertainty is linear-Gaussian. As a result, uncertainty distributions can be derived analytically *a priori* at all future time steps, and theoretical guarantees maintained—but subject to polyhedral state constraints. In subsequent developments, we choose to take equi-spaced samples of the uncertainty distributions, such that they can be checked for collision against arbitrary (i.e., not necessarily polyhedral), and potentially aggressive terrain map functions. Though probabilistic guarantees are approximated statistically, uncertainty samples are obtained without dynamic state propagation. As a result, this variant of CC-RRT is more efficient, and better representative of uncertainty distributions, than previous particle-based formulations [62]. Furthermore, the use of sampling allows for path-wise probabilistic feasibility to be quickly evaluated.

### 2.4.1 Analytic Uncertainty Derivation

Consider the parafoil state dynamics (2.11)-(2.15), written in discrete-time form as

$$
\begin{aligned}
p_{x,t+1} &= p_{x,t} + dt\left[v(p_{z,t})\cos\psi_t + w_{x,t}\right], && (2.32)\\
p_{y,t+1} &= p_{y,t} + dt\left[v(p_{z,t})\sin\psi_t + w_{y,t}\right], && (2.33)\\
p_{z,t+1} &= p_{z,t} + dt\left[v(p_{z,t})\left(-L_D^{-1}\right) + w_{z,t}\right], && (2.34)\\
\mathbf{s}_{t+1} &= \mathbf{s}_t + dt\left[A\mathbf{s}_t + Bu_t\right], && (2.35)\\
\psi_{t+1} &= \psi_t + dt\cdot\text{sat}(C\mathbf{s}_t + Du_t, -\omega_{\max}, \omega_{\max}). && (2.36)
\end{aligned}
$$

The final two equations (2.35)-(2.36) are unaffected by the uncertainty $\mathbf{v}_t$, which manifests itself only through $\mathbf{w}_t$. As in Section 2.3.2, take the variation $\delta\mathbf{p} =$

$\begin{bmatrix} p_x - E[p_x] & p_y - E[p_y] & p_z - E[p_z] \end{bmatrix}^T$. Recalling that $\delta w_z = 0$ (and thus $p_z = E[p_z]$), combining (2.32)-(2.34) with (2.26) in discrete-time form yields

$$\begin{aligned} \delta p_{x,t+1} &= \delta p_{x,t} + dt\,(\delta w_{x,t})\,, && (2.37)\\ \delta p_{y,t+1} &= \delta p_{y,t} + dt\,(\delta w_{y,t})\,, && (2.38)\\ \delta p_{z,t+1} &= \delta p_{z,t}, && (2.39)\\ \delta w_{x,t+1} &= (I + dt\alpha_c)\delta w_{x,t} + dt\beta_c v_{x,t}, && (2.40)\\ \delta w_{y,t+1} &= (I + dt\alpha_c)\delta w_{y,t} + dt\beta_c v_{y,t}; && (2.41) \end{aligned}$$

as a result, (2.39) has decoupled from the other variational dynamics. By defining the 2D variation state vector $\delta\mathbf{x}_t = \begin{bmatrix} \delta p_{x,t} & \delta p_{y,t} & \delta w_{x,t} & \delta w_{y,t} \end{bmatrix}^T$, the variation dynamics (2.37)-(2.38), (2.40)-(2.41) can be written in the linear form

$$\delta\mathbf{x}_{t+1} = \mathbb{A}\delta\mathbf{x}_t + \mathbb{B}\mathbf{v}_t, \qquad (2.42)$$

$$\mathbb{A} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 + dt\alpha_c & 0 \\ 0 & 0 & 0 & 1 + dt\alpha_c \end{bmatrix}, \quad \mathbb{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ dt\beta_c & 0 \\ 0 & dt\beta_c \end{bmatrix}. \qquad (2.43)$$

Because the linear system (2.42) is driven by Gaussian noise, all future state distributions take the form $\mathbf{x}_t \in \mathcal{N}(\widehat{\mathbf{x}}_t, P_t)$, i.e. Gaussian with mean $\widehat{\mathbf{x}}_t$ and covariance $P_t \equiv E[\delta\mathbf{x}_t \delta\mathbf{x}_t^T]$. The mean can be computed using the disturbance-free dynamics (2.50)-(2.51), while the covariance can be represented either implicitly as

$$P_{t+1} = \mathbb{A}P_t\mathbb{A}^T + \mathbb{B}\mathbb{B}^T \qquad (2.44)$$

or explicitly as

$$\begin{aligned} P_t &= \mathbb{A}^t P_0 (\mathbb{A}^T)^t + \sum_{k=0}^{t-1} \mathbb{A}^{t-k-1}\mathbb{B}\mathbb{B}^T(\mathbb{A}^T)^{t-k-1}, && (2.45)\\ P_0 &= \begin{bmatrix} Q_I & 0 \\ 0 & \Sigma_I \end{bmatrix}, && (2.46) \end{aligned}$$

where $Q_I$ and $\Sigma_I$ are the initial covariance for the position and wind, respectively. Either quantity is zero if perfectly known, but may be non-zero if, for example, an estimator is providing data to the system. As in the conventional CC-RRT framework, (2.44)-(2.45) can be computed *a priori*, independently of any individual simulated trajectory [45]. Finally, the covariance of the position states can be isolated via the transformation

$$Q_t = C_T P_t C_T^T, \quad C_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.47}$$

### 2.4.2 Covariance Sampling Generation

In order to efficiently check path-wise feasibility of the terrain constraint $p_z \geq T(p_x, p_y)$, the probability of terrain collision is approximated by generating equi-spaced samples at specified levels of the uncertainty distribution at each prospective trajectory node. Sampling the distribution in this way allows for coverage of the uncertainty space with relatively few samples, as well as removing the need to dynamically propagate each sample, significantly reducing computation. The discretization level of the samples $N_S$, as well as the minimum probability of feasibility $p_{\text{safe}}$, can both be specified by the user to allow for tunable levels of robustness.

The covariance samples are placed at a series of equi-spaced points along uncertainty ellipses. The covariance matrix $Q_t$ describes a contour of equal probability points $\delta \mathbf{p}_t = \begin{bmatrix} \delta p_{x,t} & \delta p_{y,t} \end{bmatrix}^T$, relative to the nominally propagated trajectory $\widehat{\mathbf{x}}_t$, via the conic relaxation $\delta \mathbf{p}_t^T Q_t^{-1} \delta \mathbf{p}_t = 1$. Denote the elements of $Q_t$ as

$$Q_t = \begin{bmatrix} \sigma_{x,t}^2 & \sigma_{xy,t} \\ \sigma_{xy,t} & \sigma_{y,t}^2 \end{bmatrix}, \tag{2.48}$$

with eigenvalues $\sigma_a^2$ and $\sigma_b^2$ ($\sigma_a > \sigma_b$). The principle axis of the uncertainty ellipse has angle $\theta' = \frac{1}{2}\tan^{-1}\left(\frac{2\sigma_{xy.t}}{\sigma_{x,t}^2 - \sigma_{y,t}^2}\right)$. The $N_S$ samples are spaced at equal angular intervals relative to this principle axis; the $j$th sample $\delta \mathbf{p}_t^{(j)}$ has angle $\theta_j = \frac{2\pi}{N_S - 1} j$, relative to

$\theta'$. The samples are thus placed at

$$\begin{aligned} \delta\mathbf{p}_t^{(j)} &= \sigma R_Q \mathbb{R}(-\theta') \begin{bmatrix} \cos(\theta_j - \theta') \\ \sin(\theta_j - \theta') \end{bmatrix}, \\ R_Q &= \frac{\sigma_a \sigma_b}{\sqrt{(\sigma_b \cos(\theta_j - \theta'))^2 + (\sigma_a \sin(\theta_j - \theta'))^2}}, \end{aligned} \tag{2.49}$$

where $\sigma > 0$ is the covariance scale factor and $\mathbb{R}(\alpha)$ is the 2D rotation matrix for a counterclockwise rotation of $\alpha$.

The parameter $\sigma$ represents the number of standard deviations within the uncertainty ellipse; samples may also be distributed across multiple values of $\sigma$. In subsequent results, two rings of covariance samples are used, one at $0.4\sigma$ and another at $1\sigma$ for $\sigma = 1.75$. These values were found to work well empirically.

## 2.5 Parafoil CC-RRT Path Planning

This section presents the parafoil Analytic CC-RRT algorithm, for robust motion planning using the previously-developed wind model (Section 2.3) and covariance sampling technique (Section 2.4). The core algorithm upon which the Analytic CC-RRT framework builds is RRT, which incrementally constructs a tree of dynamically feasible trajectories from the current state [49]. While many algorithms are available for motion planning problems of this nature [63, 64], an RRT-based approach is particularly well-suited to this application. The lack of controllability in the altitude state $p_z$ limits the utility of graph-based approaches. An RRT can quickly identify and refine feasible solutions online within the 9-dimensional configuration space (3 for position, 1 for heading, 5 for lag dynamics) without the need for state space discretization. Additionally, RRT's incremental trajectory construction and constraint checking allow is to scale favorably with both problem complexity and available computational resources [46].

Let the current time step be $t$; the tree is rooted at the current vehicle state, $\mathbf{x}_t$, and the most recent wind measurement is denoted as $\mathbf{w}_t$. Each simulated trajectory within the tree uses the nominal dynamics (2.5) and wind model (2.7), written as

$$\begin{aligned} \widehat{\mathbf{x}}_{t+k+1|t} &= f(\widehat{\mathbf{x}}_{t+k|t}, \mathbf{u}_{t+k|t}, \widehat{\mathbf{w}}_{t+k|t}), \quad \widehat{\mathbf{x}}_{t|t} = \mathbf{x}_t, & (2.50) \\ \widehat{\mathbf{w}}_{t+k+1|t} &= f_w(\widehat{\mathbf{w}}_{t+k|t}, \overline{\mathbf{w}}_t, 0), \quad \widehat{\mathbf{w}}_{t|t} = \mathbf{w}_t. & (2.51) \end{aligned}$$

where subscript $(\cdot)_{\alpha|\beta}$ denotes simulation time step $\alpha$ and execution time step $\beta < \alpha$.

Whereas the nominal RRT algorithm grows a tree of states which are known to be feasible, with any uncertainties assumed to maintain nominal values (here $\mathbf{v} \equiv 0$), CC-RRT grows a tree of state *distributions* which are checked for feasibility by satisfying an upper bound on the probability of collision at each time step [45]. The Analytic CC-RRT algorithm similarly generates a tree of uncertainty distributions around trajectories, but further performs analytic sampling via the approach introduced in Section 2.4.1. This allows path-wise probabilistic feasibility checks to be enforced against arbitrary terrain maps.

As shown in Section 2.4.1, the uncertainty at each prediction time step $t + k$, relative to execution time step $t$, can be represented as a Gaussian state distribution

$$\mathbf{x}_{t+k|t} \sim \mathcal{N}(\widehat{\mathbf{x}}_{t+k|t}, P_{t+k|t}). \qquad (2.52)$$

The mean state $\widehat{\mathbf{x}}_{t+k|t}$, with position $(\widehat{p}_{x,t+k|t}, \widehat{p}_{y,t+k|t}, \widehat{p}_{z,t+k|t})$, can be simulated using the disturbance-free dynamics (2.50)-(2.51), while the covariance $P_{t+k|t}$ can be computed via (2.44). Using (2.49), the covariance samples are placed at offsets $\delta\mathbf{p}^{(j)}_{t+k|t} = \left(\delta p^{(j)}_{x,t+k|t}, \delta p^{(j)}_{y,t+k|t}, 0\right), \ j \in \{1, \ldots, N_S\}$.

Probabilistic feasibility is checked statistically by determining whether the fraction of covariance samples for a given trajectory state $\mathbf{x}_{t+k|t}$ that have intersected the terrain, at this or any previous simulation step, exceeds $1 - p_{\text{safe}}$. Given the terrain map $T(p_x, p_y)$, the probability of terrain collision $p_{\text{collide}}$ at simulation time step $t + k$ is approximated as

$$p_{\text{collide}} = \frac{1}{N_S} \sum_{j=1}^{N_S} \mathbb{I} \left[ \bigwedge_{i=0}^{k} \widehat{p}_{z,t+i|t} \leq T\left( \widehat{p}_{x,t+i|t} + \delta p^{(j)}_{x,t+i|t}, \widehat{p}_{y,t+i|t} + \delta p^{(j)}_{y,t+i|t} \right) \right], \qquad (2.53)$$

**Algorithm 1** Analytic CC-RRT: Tree Growth [46]

```
Take a sample $\mathbf{x}_{\text{samp}}$ from the environment
Identify the $m$ nearest nodes using heuristics
for $m \leq M$ sorted nearest nodes do
    $N_{near} \leftarrow$ current node, $(\widehat{\mathbf{x}}_{t+k|t}, \widehat{\mathbf{w}}_{t+k|t}) \leftarrow$ final vehicle and wind state of $N_{near}$
    $p_{\text{collide}} \leftarrow 0$
    while $p_{\text{collide}} < p_{\text{safe}}$ and (2.54) true and $\widehat{\mathbf{x}}_{t+k|t}$ has not reached $\mathbf{x}_{\text{samp}}$ do
        Select input $\mathbf{u}_{t+k|t} \in \mathcal{U}$
        Simulate $(\widehat{\mathbf{x}}_{t+k+1|t}, \widehat{\mathbf{w}}_{t+k+1|t})$ using (2.50),(2.51)
        Create intermediate nodes as appropriate
        Compute/retrieve $P_{t+k+1|t}$ using (2.44)
        Compute $p_{\text{collide}}$ using (2.49),(2.53)
        $k \leftarrow k+1$
    end while
    for each identified node $N$ do
        Add $N$ to tree
        Try connecting $N$ to $\mathbf{x}_G$
    end for
end for
```

where $\mathbb{I}[\cdot]$ is the indicator function, i.e. 1 if the contained statement is true and 0 otherwise, and $\bigwedge$ represents a conjunction of the indexed constraints. If $p_{\text{collide}} > 1 - p_{\text{safe}}$, then the trajectory is considered to have landed on the terrain.

In addition to the uncertainty-based feasibility check, a trajectory is also considered to have landed if the nominal trajectory intersects the terrain. In other words,

$$\widehat{p}_{z,t+k|t} > T\left(\widehat{p}_{x,t+k|t}, \widehat{p}_{y,t+k|t}\right) \tag{2.54}$$

is added as an additional state constraint.

As with the conventional RRT algorithm, the Analytic CC-RRT algorithm consists of two core components: a "tree growth" step (Algorithm 1) that incrementally constructs the tree, and an "execution" step (Algorithm 2) that selects paths from the tree for parafoil execution with some frequency [46]. Each time Algorithm 1 is called, a state is first sampled from the environment, and the nodes nearest to this sample in terms of some heuristics are identified as candidates for tree expansion. An attempt is made to form a connection from the nearest node(s) to the

**Algorithm 2** Analytic CC-RRT: Execution [46]

```
Input: Initial vehicle state $\mathbf{x}_I$, initial wind measurement $\mathbf{w}_I$, goal state $\mathbf{x}_G$
$t \leftarrow 0$, $\mathbf{x}_t \leftarrow \mathbf{x}_I$, $\mathbf{w}_t \leftarrow \mathbf{w}_I$
Initialize tree with node at $\mathbf{x}_t$
while $\mathbf{x}_t \in \mathcal{X}$ do
    Update current vehicle state $\mathbf{x}_t$, wind measurement $\mathbf{w}_t$, and mean wind estimate $\overline{\mathbf{w}}_t$
    Propagate mean state $\mathbf{x}_t$ by the computation time $\rightarrow \mathbf{x}_{t+\Delta t}$ using (2.5),(2.7)
    Update tree feasibility and costs using (2.55)
    while time remaining for this time step do
        Expand the tree by adding nodes (Algorithm 1)
    end while
    Use cost (2.55) to identify lowest-cost path $\{N_{\text{root}}, \dots, N_{\text{target}}\}$
    if at least one path exists then
        Apply best path
    else
        Apply "safe" action
    end if
    $t \leftarrow t + \Delta t$
end while
Mark vehicle as landed at $\mathbf{x}_t$
```

sample by generating a probabilistically feasible trajectory between them. Let the vehicle state and wind state at the nearest node be denoted by $(\widehat{\mathbf{x}}_{t+k|t}, \widehat{\mathbf{w}}_{t+k|t})$. This trajectory is incrementally simulated by selecting some feasible input $\mathbf{u}_{t+k|t} \in \mathcal{U}$, then simulating the disturbance-free vehicle and wind dynamics via (2.50)-(2.51) to yield $(\widehat{\mathbf{x}}_{t+k+1|t}, \widehat{\mathbf{w}}_{t+k+1|t})$. This input may be selected at the user's discretion, such as through random sampling or a closed-loop controller [65], but should guide the state distribution toward the sample. Intermediate nodes may be occasionally inserted during the trajectory generation process, to encourage future expansion.

To check feasibility, the algorithm computes, or retrieves if previously computed offline, the covariance $P_{t+k+1|t}$ at each simulation step using (2.44). It then computes $p_{\text{collide}}$ (initialized to 0) based on the covariance samples that are feasible up to that simulation step via (2.53), in order to check whether both $p_{\text{collide}} < 1 - p_{\text{safe}}$ and (2.54) are satisfied. Trajectory simulation continues until either constraint is violated or the state has reached the sample. Unlike conventional RRT algorithms, every simulated node is added to the tree, even if it does not reach the intended sample before intersecting the terrain. As each node is added to the tree, an attempt is made to connect it directly to the goal state $\mathbf{x}_G$.

After completing the "tree growth" step of the current iteration, the "execution" step (Algorithm 2) executes some portion of the tree while continuing to grow and update the tree in subsequent iterations. Algorithm 2 updates the current best path to be executed by the system every $\Delta t$ seconds, where in this work $\Delta t = 1$ s.

The execution step first retrieves the current vehicle state, wind measurement, and mean wind estimate, then updates the tree via re-propagation. In this update, all nodes are re-checked for probabilistic feasibility and any nodes which have become infeasible are pruned, along with their descendants. Additionally, costs are updated for each node using the cost function (2.3), written in discrete form as

$$J(N_{\text{target}}) = \phi_F(\mathbf{x}_{t_F|t+\Delta t}, \mathbf{x}_G) + \Delta t \sum_{i=t+\Delta t}^{t_F} \phi(\mathbf{x}_{i|t+\Delta t}, \mathbf{x}_G). \tag{2.55}$$

After these updates, the tree is repeatedly expanded using Algorithm 1 for the duration of the time step. Following this tree growth, (2.55) is used to select the lowest-cost path in the tree for execution. If no path exists in the tree, some "safe" motion primitive can be applied to attempt to keep the vehicle in a safe state.

As described above, the entire tree is re-propagated for both feasibility and cost, rather than simply re-checking the feasibility of the lowest-cost path via "lazy check" [65] without updating costs. In the context of the parafoil terminal guidance problem, where feasibility and cost are inextricably linked and highly dynamic as a function of the uncertainty, it is useful to update all possible trajectories in order to achieve reliable performance. While additional computation is required to perform this update, in practice this computation is balanced over time by the tree size, via the amount of time spent in Algorithm 1. Section 2.7 demonstrates the effectiveness of the Analytic CC-RRT algorithm for improving worst-case performance of parafoil terminal guidance, particularly subject to complex terrain and pathological wind conditions.

Figure 2-2 shows an example of an Analytic CC-RRT tree generated for the valley terrain, described in Section 2.7.1, for $\sigma = 1.5$. The planner constructs a tree of

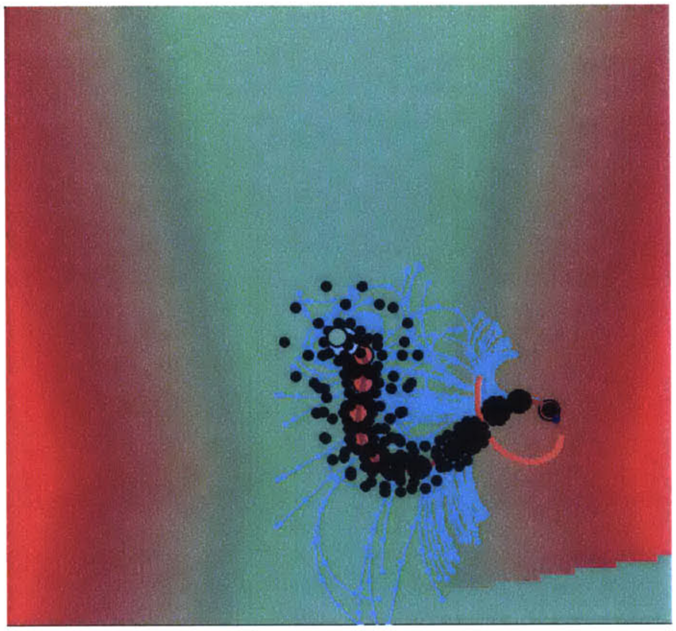


Figure 2-2: Analytic CC-RRT simulation in progress, showing covariance samples (black)

feasible trajectories (teal), which guides the parafoil (blue) to land on the terrain at the goal location (green circle). In Figure 2-2, the terrain background changes from green to red with increasing altitude, and the covariance samples (black) are shown for the path currently selected for execution (orange).

### 2.5.1 Reference Model

A reference model is used in the Analytic CC-RRT algorithm to generate a sequence of inputs $u = \dot{\psi}_d$ for each trajectory. Each trajectory connects a nearest node, with position $\mathbf{p}_n$ and heading $\psi_n$, to a sample with position $\mathbf{p}_s$. Since the altitude state $p_z$ is uncontrollable, the proposed reference model generates the 2D circular arc which connects the nearest node and the sample, and is tangent to the heading at the nearest node, as shown in Figure 2-3. Such a circular arc can be followed by the ideal (i.e., no lag dynamics) parafoil in 2D, by applying a constant input $\overline{u}$ for some time duration $\overline{t}$. Resulting trajectories are thus sequences of piecewise-constant-angular-rate segments. The number of segments and the duration of each segment are not fixed, such that complex trajectories can still be specified within this reference model.

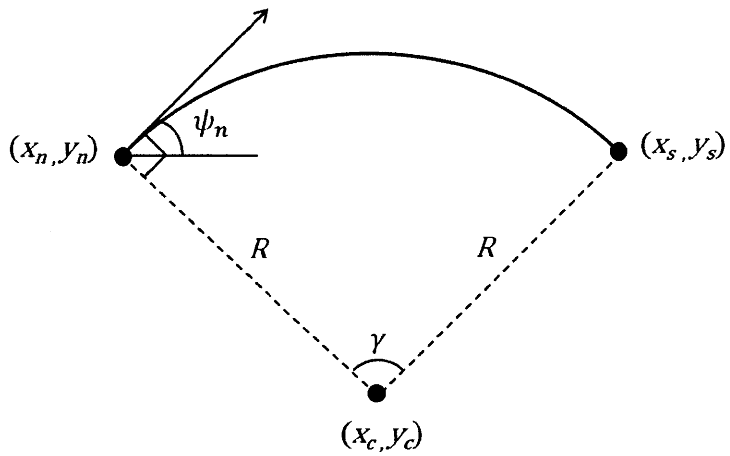


Figure 2-3: Reference generation model using 2D circular arcs

Defining $\delta_y = y_s - y_n$, $\delta_x = x_s - x_n$ and $\delta = \sqrt{\delta_x^2 + \delta_y^2}$, the radius of this circle is

$$R = \frac{\delta^2}{2(\delta_y \cos\psi_n - \delta_x \sin\psi_n)}, \tag{2.56}$$

where the sign of $R$ in (2.56) encodes the turn direction. This is used with the velocity model (2.10) to yield the desired angular rate, $\overline{u} = v(p_{zn})/R$. The duration of the reference command is determined by computing the arc subtended angle $\gamma$:

$$\gamma = 2\sin^{-1}\left(\frac{\delta}{|2R|}\right), \tag{2.57}$$

$$\overline{t} = \gamma/\overline{u}. \tag{2.58}$$

Alternative reference generation models could be applied, such as Bezier curves [50], B-splines, or piecewise linear (rather than piecewise constant) angular rate commands. Closed-loop propagation may also be used to limit uncertainty growth over time.

## 2.6 Reachability-based Cost-to-go

One of the advantages of using RRTs, due to their incremental construction, 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 path during later planning cycles. Critical to this capability is an informative cost-to-go function for assessing path quality. In previous work, Luders et al. [46] proposed a cost-to-go formulation which combines the cost at a point of interest – the end of a trajectory within the tree – with a discrete approximation of the reachability set beyond that point. In this manner, the cost-to-go function weighs the value of the system's present state against possible near-term future states, which are heavily constrained by the present state and particularly the parafoil heading [46]. This work extends the previous cost-to-go formulation with the option to include an additional penalty term on the parafoil ground speed. Because the planner can only affect its velocity by changing its heading relative to the wind direction, such a penalty term encourages the selection of paths with upwind landings to reduce landing speed.

The cost-to-go function is constructed by assigning costs $J_i$ to each of a set of states $\mathbf{x}_i$, $i \in \{0, \ldots, N_u\}$, where $\mathbf{x}_0$ is the vehicle's current state and $\mathbf{x}_i$, $i \in \{1, \ldots, N_u\}$ are drawn from the boundary of a finite-time reachability set for the parafoil [46]. The cost $J_i$ is the Euclidean distance from the point $\mathbf{x}_i$ to the goal $\mathbf{x}_G$ at position $\mathbf{p}_G = (p_{x,G}, p_{y,G}, p_{z,G})$, after accounting for drift due to the persistent wind estimate $\overline{\mathbf{w}} = (\overline{w}_x, \overline{w}_y, \overline{w}_z)$. This takes the form

$$J_i = \sqrt{(p_{xi} - p_{z,G} - t_{zG}\overline{w}_x)^2 + (p_{yi} - p_{y,G} - t_{zG}\overline{w}_y)^2 + (p_{zi} - p_{z,G})^2}, \tag{2.59}$$

where $t_{z,G} = (p_z - p_{z,G})/v_z$ is the time to reach the goal altitude from the current state. The full cost-to-go function takes the form

$$\phi_F(\mathbf{x}_0, \mathbf{x}_G) = C_v \max(J_0, \min(J_1, J_2, \ldots, J_{N_u})) + J_v(\mathbf{x}_0, \overline{\mathbf{w}}), \tag{2.60}$$

where $C_v > 0$. The first term of (2.60) takes the maximum between the cost of the initial point, $J_0$, and the minimum of the cost of points on the reachability set approximation. The $J_0$ portion encourages the planner to situate the vehicle directly above the goal, after correcting for wind, to facilitate disturbance rejection, while the $\min(J_1, J_2, \ldots, J_{N_u})$ portion represents the most favorable state the vehicle can reach within the approximated finite-time reachability set [46], described in Section 2.6.1.

The second term of (2.60) drives the desired heading at landing $\psi_G$ toward the upwind direction by penalizing the node's terminal ground speed. Since the ground speed is minimized when the parafoil heading and wind direction are opposing, and maximized when they align, this induces the desired behavior. This cost term is defined as

$$J_v(\mathbf{x}_0, \overline{\mathbf{w}}) = \sqrt{(v(p_{z0}) \cos \psi_0 + \overline{w}_x)^2 + (v(p_{z0}) \sin \psi_0 + \overline{w}_y)^2}, \qquad (2.61)$$

incorporating both the parafoil air speed and estimated wind effect. The relative importance of landing accuracy vs. landing speed is controlled by the weighting coefficient $C_v$; in this work, a weight of $C_v = 10$ is used. Section 2.7.3 examines the effect of this additional penalty term within the cost-to-go function on the parafoil landing speed during terminal approach.

### 2.6.1 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. For a given dynamical system, the full reachability set can be defined as all possible future states that can be reached from the current state by applying the appropriate sequence of input commands. In the case of the parafoil system, this set can be computed by propagating the dynamics forward in time through all possible input sequences until intersection with the terrain occurs. Because such a set is intractable to construct, previous work in [46] developed a reachability approximation which includes only a finite set of $N_u$ constant control sequences, propagated over the finite time horizon $t_P$. To cover the largest possible region, control sequences $\omega_i$ along the reachability horizon are spaced at equal intervals within the limits of the input $|u| \leq \omega_{max}$ according to

$$\omega_i = -\omega_{\max} + \frac{2\omega_{\max}}{N_u - 1} i, \quad i \in \{0, ..., N_u - 1\}. \qquad (2.62)$$

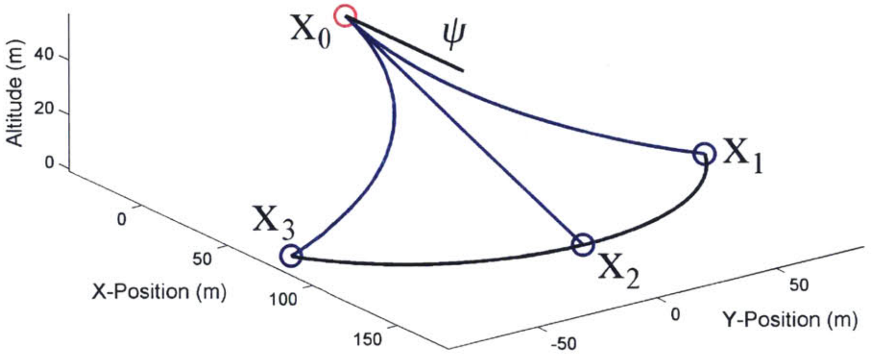


Figure 2-4: Reachability set approximation $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})$ relative to the state $\mathbf{x_0}$

Here only odd values of $N_u$ are considered in order to include the propagation of the dynamics under zero angular rate, ($\omega_i = 0$, for $i = (N_u - 1)/2$). Assuming lag free dynamics ($A = B = C = 0$, $D = 1$, and $\dot{\psi} = u$), the states $\mathbf{x}_i = (p_{xi}, p_{yi}, p_{zi})$ along the reachability frontier can by calculated relative to the current state $\mathbf{x}_0 = (p_{x0}, p_{y0}, p_{z0})$ and heading $\psi_0$ using

$$\begin{aligned} p_{xi} &= p_{x0} + \left|\frac{v(p_{z0})}{\omega_i}\right| \left(\cos(\psi_0 + \mathrm{sign}(\omega_i)\frac{\pi}{2}) + \cos(\psi_0 + (2 + \mathrm{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i\tau)\right), \\ p_{yi} &= p_{y0} + \left|\frac{v(p_{z0})}{\omega_i}\right| \left(\sin(\psi_0 + \mathrm{sign}(\omega_i)\frac{\pi}{2}) + \sin(\psi_0 + (2 + \mathrm{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i\tau)\right), \\ p_{zi} &= p_{z0} - \frac{v(p_{z0})}{L_D}\tau, \end{aligned} \tag{2.63}$$

where the propagation time $\tau \equiv \min\{t_{zG}, t_P\}$ is taken as the minimum between the time to reach the goal altitude $t_{zG}$, and the propagation upper bound $t_P$. Figure 2-4 illustrates the result of the reachability set approximation using $N_u = 3$ and $\tau = t_P = \frac{\pi}{2\omega_{\max}}$, representing a quarter-turn at maximum angular rate.

By formulating the reachability approximation in this way, increasing the value of $N_u$ increases the density of states $\mathbf{x}_i$ along the reachability frontier through the number of control sequences $\omega_i$ considered by the planner. Likewise, extending the upper bound on the propagation horizon $t_P$ allows the planner to look further forward in time while holding each input constant for $u = \omega_i$. From previous analysis conducted in [46, 47], the values of $N_u = 3$ and $t_P = 7.5$ seconds are selected for use throughout the remainder of this chapter.

## 2.7 Simulation Results

This section presents simulation results demonstrating that Analytic CC-RRT achieves superior landing accuracy relative to both nominal RRT and BLG [32] on challenging terrain. In particular, Analytic CC-RRT demonstrates a significant improvement in mean accuracy over BLG, and superior worst-case landing accuracy over both RRT and BLG. Adding a penalty on landing speed to the cost-to-go function (Section 2.6) reduces parafoil landing accuracy slightly, but significantly reduces typical landing speeds by better orienting the parafoil upwind during final approach. Further analysis shows that while BLG performance tends to degrade as the difficulty of the terrain increases, Analytic CC-RRT performance is largely invariant to changes in terrain for both the mean and worst-case. Analytic CC-RRT is also shown to be not only capable of use at higher initial altitudes, but also invariant to initial drop altitude.

### 2.7.1 Implementation

Three algorithms are compared throughout this section:

1) **RRT with mean wind**, which represents a nominal RRT planner using the mean wind estimate $\overline{\mathbf{w}}$, but assumes no future wind variation (i.e. $\delta\mathbf{w} \equiv 0$, $\mathbf{w}_t \equiv \overline{\mathbf{w}}$). This approach makes no active attempt at robustness against uncertainty, but does utilize replanning at every time step to try to counteract system disturbances.

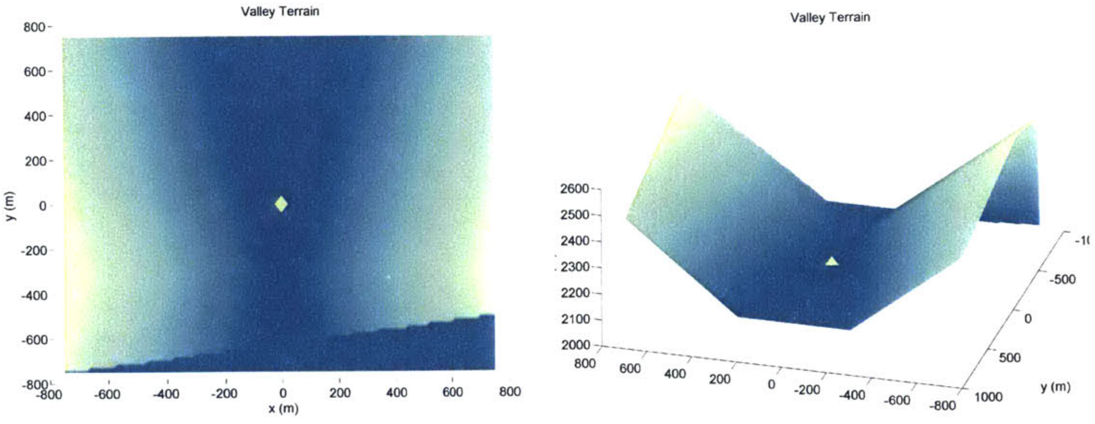
2) **Analytic CC-RRT**, the full Analytic CC-RRT algorithm presented throughout this chapter and specified in Algorithm 1 and Algorithm 2 of Section 2.5.

3) **BLG**, or Band-Limited Guidance, which utilizes band-limited control to ensure accurate tracking and prediction, as well as knowledge of the mean wind estimate $\overline{\mathbf{w}}$, and replanning to account for system disturbances [32]. The implementation of BLG is detailed below.

For each parafoil drop scenario, each algorithm is tested over a large series of simulation trials, which vary in the combination of wind profile and initial conditions used. Of the wind profiles released by Draper Laboratory [46], a set of 25 representative wind profiles are selected. These 25 wind profiles consist of 18 profiles from collected drop data and 7 artificially generated profiles. Of the 7 artificially generated profiles, 6 are constant-wind profiles moving in the cardinal directions, varying in intensity from zero wind to 25 knots (over 70% of the parafoil airspeed). The final artificially generated profile represents an exponentially decaying wind, with average and maximum wind speed changes with respect to altitude of 0.0025 $\frac{\text{m/s}}{\text{m}}$ and 0.05 $\frac{\text{m/s}}{\text{m}}$, respectively. The actual drop wind profiles are significantly more aggressive, with an average overall intensity of 6.7 m/s and gusts up to 17.1 m/s (nearly matching the parafoil airspeed). These profiles are subject to average and maximum wind speed changes with respect to altitude of 0.025 $\frac{\text{m/s}}{\text{m}}$ and 2.4 $\frac{\text{m/s}}{\text{m}}$, respectively. They are also subject to rapid directional changes, potentially as large as 115°/m.

In each trial, the parafoil state is initialized 500 meters above the target, with a random heading and a lateral distance from the target between 100 and 400 meters. In all scenarios, the target/goal is assumed to be located at $\mathbf{p}_G = (0, 0, 0)$. Each algorithm is subject to the same sequence of wind profile and initial condition combinations. A total of 500 trials are performed for each algorithm in each scenario, representing 20 uses of each wind profile for different initial conditions.

The primary terrain used for simulations is the 1.5 km × 1.5 km valley terrain, $T_{\text{valley}}(p_x, p_y)$, pictured in Figure 2-5. The green shades in Figure 2-5 indicate areas of lower altitude, while the goal is located at the yellow diamond. This represents a particularly challenging terrain for the parafoil terminal guidance problem, for several reasons. First, the slope of the valley is greater than the glide-slope of the parafoil, limiting planning options at lower altitudes by making approach from either side impossible. Second, the large low-altitude regions away from the goal (bottom-right and top in Figure 2-5(a)), where terrain collisions can be avoided for longer path durations, are likely to lead to terrain interactions as the parafoil's path crosses in and out of these regions. Finally, placing the goal near a terrain "bottleneck" makes planning near the goal more difficult than planning away from the goal. To test how algorithmic performance varies with terrain "difficulty," this section also considers



(a) top view (b) skew-side view

Figure 2-5: Valley terrain used for parafoil terminal guidance scenarios

scaled-down versions of the valley terrain, i.e. $\alpha T_{\text{valley}}(p_x, p_y)$ for $\alpha \in [0, 1]$. In particular, simulations are performed for $\alpha = 0$, representing completely flat terrain, and $\alpha = 0.75$, representing intermediate conditions.

The Analytic CC-RRT algorithm has been implemented as a single-threaded Java application. To simplify comparisons, a fixed number of samples, or iterations of Algorithm 1, are performed per loop of Algorithm 2. In subsequent results, 165 samples per loop are used, representing the average number of samples generated in a 1 Hz planning cycle with 60% duty cycle by the nominal RRT algorithm. The mean wind impulse filter (Section 2.3) has a width $m = 10$, while two rings of 10 covariance samples each are used with an overall $p_{\text{safe}} = 0.9$ (Section 2.4).

The BLG algorithm, against which Analytic CC-RRT is compared, determines an optimal control by choosing coefficients $\psi_k$ for the heading rate profile (1.1) described in Section 1.2. This vehicle model is fundamentally different from the one used by Analytic CC-RRT (Section 2.3.1) in the way heading rate is handled. Whereas Analytic CC-RRT assumes heading rate is the output of a linear lag-dynamics model, the BLG vehicle model assumes lag-free control over the heading rate, provided that the controls are bounded by (1.1).

The BLG algorithm has been implemented in MATLAB for comparison. In this work, the optimization is performed using iterations of MATLAB's `fmincon` function, instead of Nelder-Mead simplex optimization as in Ref. [32]. The parameter values

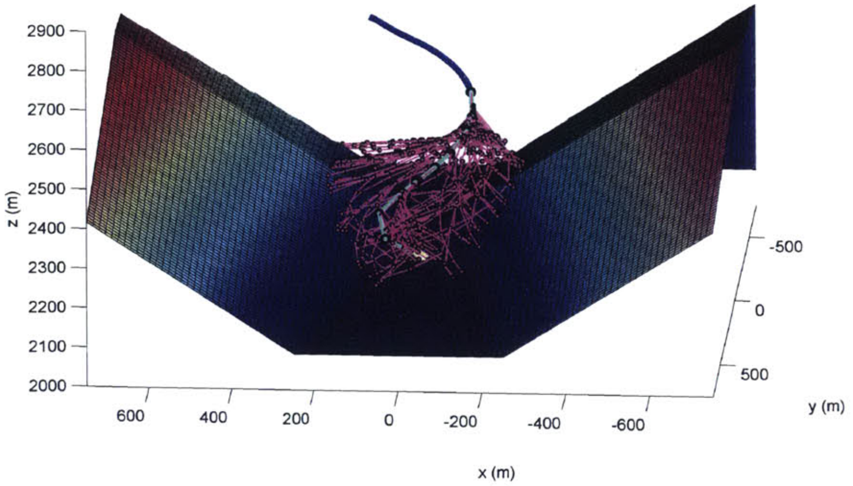


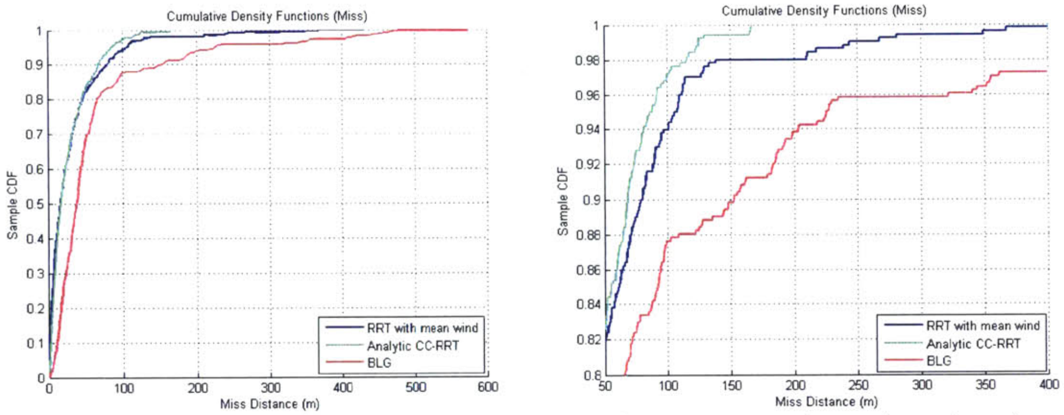
Figure 2-6: Screenshot of Analytic CC-RRT simulation on the valley terrain

$\delta h = 200$ m and $N = 4$ used in Ref. [32] are also selected for this implementation, as well. For comparison with Analytic CC-RRT, rather than using a tolerance-based stopping criterion with a maximum of 100 iterations [32], BLG is permitted to simulate the parafoil to the ground through 75 iterations per planning cycle. This number of iterations requires a comparable amount of computation to the number of RRT samples generated per planning cycle, as described above.

### 2.7.2 Valley Terrain Simulations

First, 500 trials were run for each algorithm on the valley terrain scenario, shown in Figure 2-5. Figure 2-6 provides a simulation example of Analytic CC-RRT's online execution in the valley terrain, where the parafoil trajectory (blue) and tree of feasible trajectories (magenta) are shown, including the current best path (green) to the goal location (yellow circle). Figure 2-7 and Table 2.1 show the CDF and statistics, respectively, over all trials. The results indicate that Analytic CC-RRT demonstrates matching or improved landing accuracy, relative to RRT with mean wind and BLG, at nearly all percentiles.

Both RRT-based algorithms show significant improvement over BLG for all but the worst-case trials. The mean landing accuracy for both algorithms is lower than



(a) full CDF

(b) zoomed in to $(50, 400)\text{m} \times (0.8, 1)$

Figure 2-7: Miss distance CDF for valley terrain comparison, over 500 trials

Table 2.1: Miss distance data for valley terrain comparison, over 500 trials (in meters)

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| RRT with mean wind | 32.3 | 47.0 | 17.0 | 46.9 | 79.6 | 106 | 139 | 431 |
| Analytic CC-RRT | 28.2 | 28.5 | 18.0 | 45.7 | 68.6 | 86.5 | 115 | 167 |
| BLG | 63.5 | 89.0 | 37.9 | 66.1 | 153 | 227 | 431 | 581 |

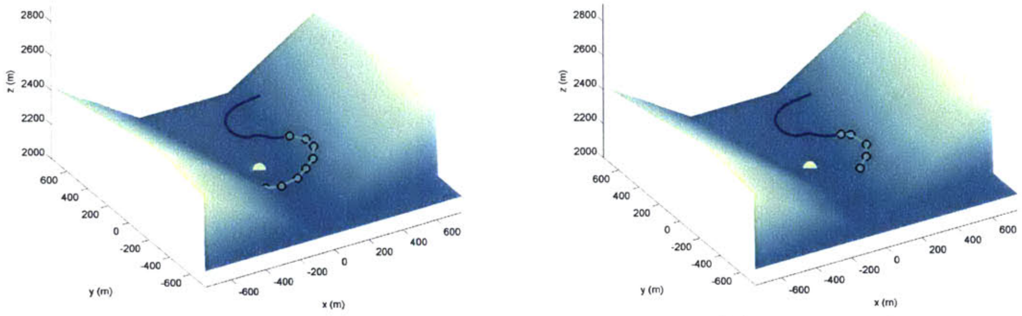
BLG by a factor of 2. This ratio continues approximately up to the 95th percentile, and increases to a factor of 3–4 by the 98th percentile (Table 2.1). In particular, about 12% of BLG trials have a miss distance exceeding 100 m, whereas less than 5% of the Analytic CC-RRT trials, and less than 6% of the RRT trials, have a miss distance exceeding 100 m (Figure 2-7(b)). The BLG algorithm also demonstrates a "long tail": 4% of trials have a miss distance of 300 m or worse, while the worst-case trial misses by 581 m.

Landing accuracy is comparable between RRT with mean wind and Analytic CC-RRT up to the 80th percentile; however, Analytic CC-RRT demonstrates superior performance over both RRT and BLG in the worst 20% of trials. This suggests that for those trials in which terrain interaction is unlikely, the robustness-based enhancements in Analytic CC-RRT do not significantly influence performance relative to RRT with mean wind prediction alone. However, the two CDF curves diverge near the 82nd percentile (Figure 2-7(b)). At the 95th and 98th percentiles, Analytic CC-RRT miss distance is 17-18% lower than RRT. By the worst-case trial (i.e. 99.8%), Analytic CC-RRT miss distance is 61% lower. All trials of Analytic CC-RRT have an accuracy under 170 m, whereas RRT demonstrates some trials exceeding 430 m (Table 2.1). This "shorter tail" for the Analytic CC-RRT distribution, relative to RRT with mean wind (Figure 2-7), demonstrates the robustness of the algorithm to pathological uncertainty conditions, which might otherwise drive the vehicle prematurely into the terrain.

Both RRT with mean wind and BLG encounter trials where landing accuracy exceeds 400 m—with BLG sometimes encountering accuracies approaching 600 m. Such situations are the product of interaction between the uncertain wind and the difficult terrain encountered by the parafoil. Figures 2-8 and 2-9 demonstrate how changing wind conditions can cause selected/executed paths from RRT and BLG, respectively, to become infeasible despite replanning.

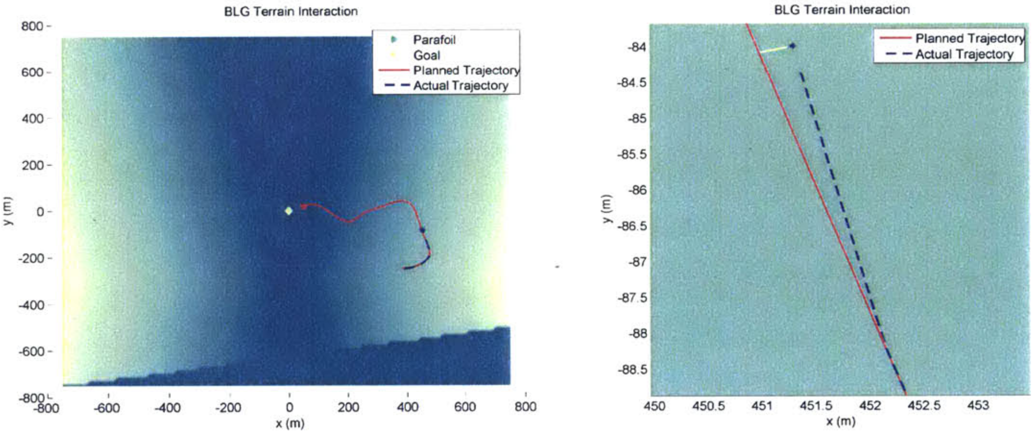
Figure 2-8 shows the planned paths (green) for RRT on successive time steps. On the first time step (Figure 2-8(a)), the RRT planner has identified a semi-circular path which brings the parafoil relatively close to the goal (yellow circle). However, after a new wind measurement, this trajectory is now predicted to collide with the terrain only halfway through this path (Figure 2-8(b)). This causes the second half of the path to be pruned, leaving the parafoil on a trajectory which now has poor terminal accuracy. The issue, in this case, is that several of the intermediate path nodes are very close to the terrain, such that a wind shift causes them to become infeasible.

Figure 2-9 compares a planned trajectory (solid red line) and executed trajectory (dashed blue line) for the BLG planner. The planned, nominal trajectory, based on the mean wind estimate, takes the parafoil very close to the goal, but also comes very close to the terrain surface on the right side of the valley before turning back toward the goal (Figure 2-9(a)). About one-quarter of the way through execution, a small wind shift takes place, resulting in a deviation between prediction and execution (Figure 2-9(b)) that yields a mismatch of less than 1 m (yellow line). Yet this mismatch is sufficient to cause the parafoil to collide with the terrain (blue star), resulting in a miss distance exceeding 450 m. The direct optimization technique of BLG does not consider off-nominal, future terrain interactions caused by changing wind conditions.



(a) Time step 1 (b) Time step 2

Figure 2-8: Trajectory planned using RRT with mean wind during the first two time steps



(a) top view (b) zoomed-in view

Figure 2-9: Trajectory planned using BLG (red) vs. executed trajectory (blue)

As a result, such adverse terrain interactions are possible in the worst case. In both examples (Figure 2-8 and Figure 2-9), such proximity to this sloping terrain would have been captured by the analytic uncertainty samples (Section 2.4) using the CC-RRT formulation, such that the original path with low robustness margin would not have been chosen for execution.

Table 2.2 shows the average computation time per node generated for both RRT with mean wind (i.e. 0 samples) and Analytic CC-RRT, for various numbers of covariance samples. The time per node for Analytic CC-RRT scales favorably both with the number of covariance samples, and relative to RRT alone. Table 2.3 demonstrates that the desirable properties of the RRT-based algorithms are largely preserved when each algorithm is run in real-time, rather than using a fixed number of iterations.

Table 2.2: Average node generation times for nominal RRT and Analytic CC-RRT

| # Covariance samples | Time per Node (ms) |
|---|---|
| 0 (RRT) | 10.4 |
| 10 | 17.00 |
| 20 | 26.24 |
| 30 | 33.19 |
| 50 | 46.00 |

Table 2.3: Miss distance comparison in real time (60% duty cycle), over 500 trials (in meters)

| **Algorithm** | $\Delta t$ (s) | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| RRT with mean wind | 1 | 30.6 | 39.6 | 18.5 | 48.3 | 76.4 | 94.6 | 135 | 353 |
| | 2 | 29.1 | 37.1 | 17.7 | 45.6 | 68.5 | 90.0 | 113 | 417 |
| Analytic CC-RRT | 1 | 42.1 | 38.1 | 30.4 | 65.7 | 91.7 | 115 | 151 | 296 |
| | 2 | 33.1 | 32.7 | 22.4 | 50.5 | 80.9 | 92.0 | 122 | 237 |

In these tests, 500 trials are again performed. However, rather than running a fixed number of iterations, the "execution" step (Algorithm 2) is run on a $\Delta t$ = 1s or $\Delta t$ = 2s update cycle. During each cycle, a 60% duty cycle (i.e. 0.6 s of each 1 s) is available for the planner to update and grow the tree.

Compared to Table 2.1, the performance of RRT with mean wind is similar for both sets of real-time results. Additionally, the extra computation available by doubling the planning time does not appreciably improve performance. Because of the additional computation needed to check covariance samples, the performance of Analytic CC-RRT degrades slightly when subject to real time planning constraints. Regardless, Analytic CC-RRT still yields the best performance in the worst-case relative to RRT with mean wind alone.

### 2.7.3 Landing Speed Penalty

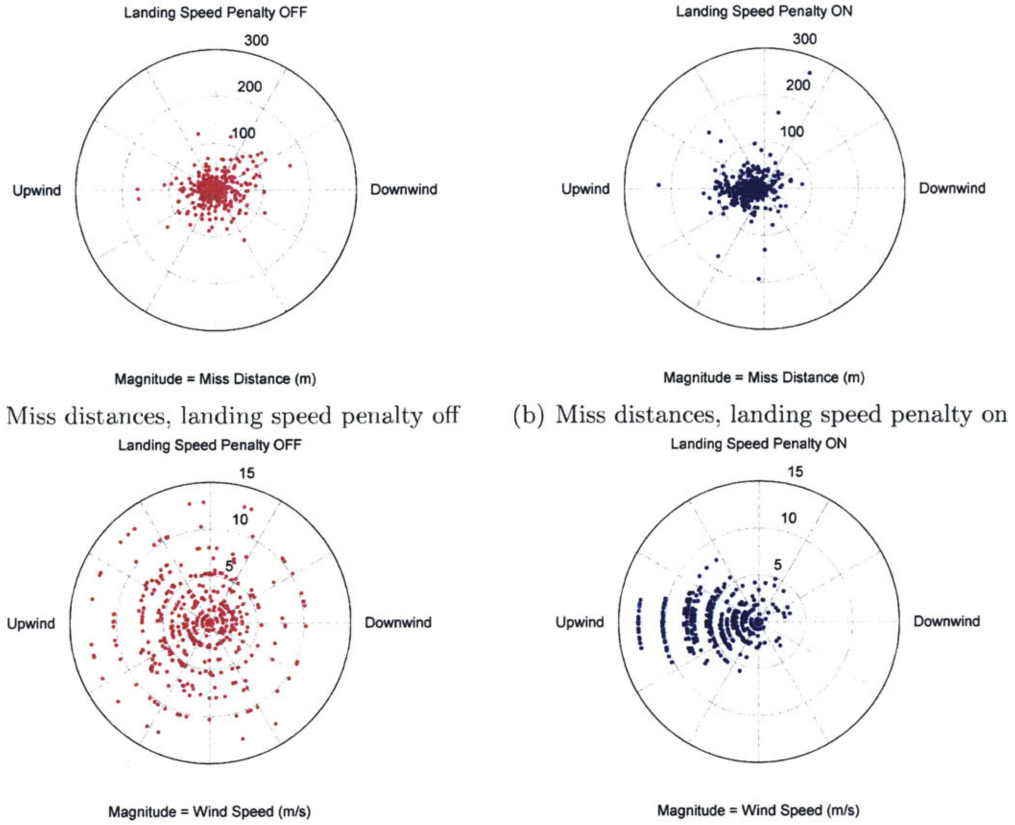
Table 2.4 compares results for Analytic CC-RRT in the valley terrain over a set of 500 trials, in which the landing speed penalty of Section 2.6 is applied ($C_v$ = 10). Results are compared in terms of both the miss distance and the ground speed of the parafoil at landing, noting that the nominal landing speed of the parafoil in the absence of wind is 19.79 m/s.

Table 2.4: Miss distance and landing speed data for Analytic CC-RRT on the valley terrain, over 500 trials

| **Metric** | **Speed penalty?** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| Miss distance (m) | No | **28.2** | 28.5 | 18.0 | 45.7 | 68.6 | 86.5 | 115 | **167** |
| | Yes | **33.6** | 32.2 | 24.9 | 54.1 | 71.9 | 92.2 | 119 | **265** |
| Landing speed (m/s) | No | **19.2** | 3.8 | 19.8 | 21.6 | 23.3 | 25.3 | 28.3 | **30.0** |
| | Yes | **16.0** | 3.7 | 16.6 | 19.8 | 19.8 | 20.1 | 21.8 | **23.1** |

Without the landing speed penalty, more than 50% of trials land at the nominal speed or higher. In the worst case, the ground speed is increased by 36% relative to the air speed, leading to potentially precarious landing conditions. With the landing speed penalty turned on, however, the mean landing speed is reduced by 17%, well below nominal speed. Less than 10% of trials yield a landing speed higher than the nominal speed, implying that more than 90% of trials are oriented within 90 degrees of the upwind direction. In the worst case, the landing speed is only 14% larger than nominal. Additionally, while the miss distances of Analytic CC-RRT are increased due to incorporation of the penalty, the increase is not large. The mean miss distance increases by less than 20%, while the worst-case miss distance remains under 300m.

Figure 2-10 gives polar scatter plots representing the spread of landing orientations over each set of 500 trials in Table 2.4. In these images, the heading of each data point corresponds to the parafoil's heading at landing relative to the wind direction in that trial; the most desirable heading is upwind (left side of each figure), while the least desirable heading is downwind (right side of each figure). With the landing speed penalty off (Figure 2-10(a) and 2-10(c)), the 500 trials are widely distributed across all landing orientations. In particular, for cases where the terminal wind exceeds 10 m/s, many trials can be seen to result in a downwind landing, significantly increasing landing speed (Figure 2-10(c)). In contrast, when the landing speed penalty is enabled, trials are heavily biased toward landing in the upwind direction. Most trials land within 30 degrees of the upwind direction, and no trials with a terminal wind exceeding 5 m/s land downwind (Figure 2-10(d)). However, some outliers are introduced in terms of miss distance (Figure 2-10(b)).



(a) Miss distances, landing speed penalty off

(b) Miss distances, landing speed penalty on

(c) Wind speeds, landing speed penalty off

(d) Wind speeds, landing speed penalty on

Figure 2-10: Radial scatter plots of each trial of Analytic CC-RRT performed on the valley terrain

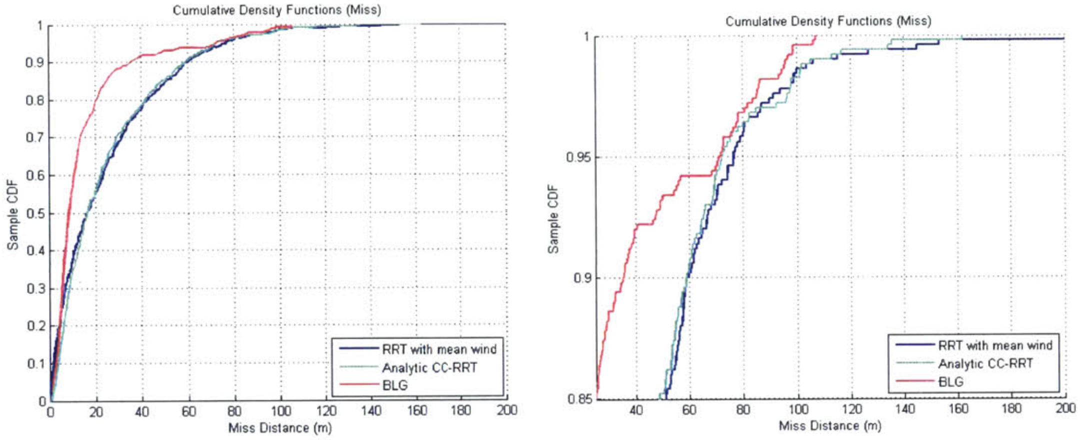
### 2.7.4 CC-RRT Invariance to Terrain

Figure 2-11 and Table 2.5 show the CDF and statistics, respectively, for 500 trials performed on a completely flat terrain, i.e. $T(p_x, p_y) \equiv 0$. Although this terrain proves less complex for the planner, the parafoil's underactuated dynamics still yield a challenging planning problem. In Figure 2-11, the miss distance accuracy of RRT with mean wind and Analytic CC-RRT have converged to approximately the same CDF, with Analytic CC-RRT showing only a slight improvement in miss distance relative to RRT with mean wind at most percentiles, including the worst-case. This is consistent with using obstacle-free terrain. Because covariance sampling is performed in the 2D horizontal plane relative to the prospective trajectories, Analytic CC-RRT samples will either all be feasible ($p_z > 0$) or all be infeasible ($p_z \leq 0$) at any given time step, and thus provide no new information to the planner. As a result, Analytic CC-RRT functions identically to RRT with mean wind in this scenario.

On the other hand, the BLG algorithm demonstrates significant improvements in accuracy, relative to both the RRT-based algorithms and its own performance on the valley terrain. Compared to the valley terrain, the mean miss distance has decreased by 75%, while the worst-case miss distance has decreased by over 80% (from nearly 600m to just over 100m). BLG also demonstrates up to 40% better accuracy than RRT with mean wind and Analytic CC-RRT, in both mean and worst-case miss distance (Table 2.5). The improvement is most noticeable between the 50th and 90th percentiles (Figure 2-11). Because of the absence of terrain features, the BLG algorithm is no longer susceptible to off-nominal terrain interactions in this scenario, creating the ideal environment for the algorithm to converge on optimal solutions. In flat terrain, finding feasible solutions, the primary strength of RRT-based algorithms, is a relatively simple task compared to more complex terrain environments. As a result, optimizing the planned trajectory, a task BLG is more effective at, is a more efficient use of available computational resources. In this scenario, replanning alone, without robustness modifications, is sufficient to counteract shifting wind conditions.

Based on this analysis, we consider how the performance of both BLG and Analytic CC-RRT varies as the "difficulty" of the terrain is changed. As stated in Section 2.7.1, this is done by considering scalings of the valley terrain, $\alpha T_{\text{valley}}(p_x, p_y)$. In addition to the cases of $\alpha = 1$ and $\alpha = 0$ already considered, we also consider "75% Valley Terrain," in which $\alpha = 0.75$, representing a terrain of intermediate difficulty. Figures 2-12(a) and 2-12(b) show the CDFs over 500 trials across all three terrains for BLG and Analytic CC-RRT, respectively, with tabular data provided in Tables 2.6 and 2.7, respectively.

Based on Figure 2-12(a) and Table 2.6, it is clear that the BLG algorithm is highly sensitive to the complexity and steepness of the terrain. As the terrain becomes more complex, feasible paths become more difficult to find, and thus cannot be optimized to the same extent. On the 75% valley terrain, there exists a regime of nominal performance, up to around the 80th percentile, where BLG performance matches or



(a) full CDF

(b) zoomed in to $(0, 200)\mathrm{m} \times (0.85, 1)$

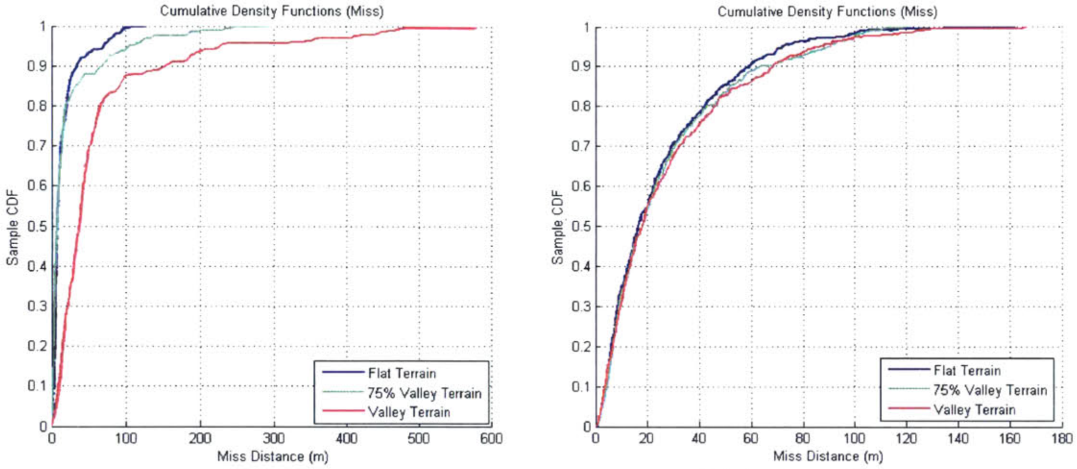
Figure 2-11: Miss distance CDF for flat terrain comparison, over 500 trials

Table 2.5: Miss distance data for flat terrain comparison, over 500 trials (in meters)

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| RRT with mean wind | 25.2 | 26.6 | 16.5 | 42.5 | 59.2 | 76.7 | 98.0 | 220 |
| Analytic CC-RRT | 25.3 | 24.5 | 16.5 | 41.8 | 58.9 | 72.2 | 97.8 | 162 |
| BLG | 15.9 | 19.7 | 8.9 | 20.7 | 35.2 | 71.3 | 86.1 | 107 |

even exceeds performance on flat terrain, with all miss distances under 40 m (Figure 2-12(a)). In these cases, finding a feasible solution is relatively straightforward, and BLG is able to spend significant time optimizing the solution. For the remaining approximately 100 trials, however, terrain interactions are a serious concern, and BLG miss distance begins to increase significantly relative to the flat terrain scenario (Figure 2-12(a)). Both the 98th percentile and worst-case BLG miss distances for the 75% valley terrain are more than twice their flat-terrain counterparts. Once the terrain scaling is increased to the full valley terrain (Section 2.7.2), the possibility of terrain interactions becomes much more significant, and BLG miss distances increase at all percentiles, especially the worst-case trials.

On the other hand, the performance of the Analytic CC-RRT algorithm (Figure 2-12(b) and Table 2.7) is largely invariant to the terrain scaling considered. The gap in mean performance between all three terrains is only 4 m or about 12%, while the gap in the 98th-percentile is only 17 m or about 9%. Indeed, there is little discernible difference between the shapes of the CDF curves for each terrain environment (Figure 2-12(b)). This suggests that Analytic CC-RRT is able to maintain consis-



(a) BLG (b) Analytic CC-RRT

Figure 2-12: Miss distance CDF for BLG and CC-RRT on various terrain, 500 trials

Table 2.6: Miss distance data for BLG on various terrain, 500 trials (in meters)

| **BLG** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| Flat Terrain | 15.9 | 19.7 | 8.9 | 20.7 | 35.2 | 71.3 | 86.1 | 107 |
| 75% Valley Terrain | 22.5 | 39.5 | 8.5 | 20.1 | 67.1 | 105 | 184 | 247 |
| Valley Terrain | 63.5 | 89.0 | 37.9 | 66.1 | 153 | 227 | 431 | 581 |

Table 2.7: Miss distance data for CC-RRT on various terrain, 500 trials (in meters)

| **Analytic CC-RRT** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| Flat Terrain | 25.3 | 24.5 | 16.5 | 41.8 | 58.9 | 72.2 | 97.8 | 162 |
| 75% Valley Terrain | 27.1 | 26.1 | 18.1 | 42.6 | 63.8 | 90.4 | 101 | 122 |
| Valley Terrain | 28.2 | 28.5 | 18.0 | 45.7 | 68.6 | 86.5 | 115 | 167 |

tent performance, regardless of the difficulty of the terrain scenario. While other algorithms may be able to leverage highly simplified terrain to improve accuracy, such as through BLG's direct optimization, Analytic CC-RRT can ensure reasonable performance even under worst-case terrain and wind conditions.

### 2.7.5 CC-RRT Invariance to Initial Altitude

One of the key advantages of Analytic CC-RRT relative to other parafoil terminal guidance algorithms is the ability to start planning from any initial altitude. Other approaches in the literature often require an upper limit on the initial altitude for terminal guidance to remain computationally tractable [3, 4, 26–28, 32]. Figure 2-13 and Table 2.8 present simulation results on the valley terrain when the initial

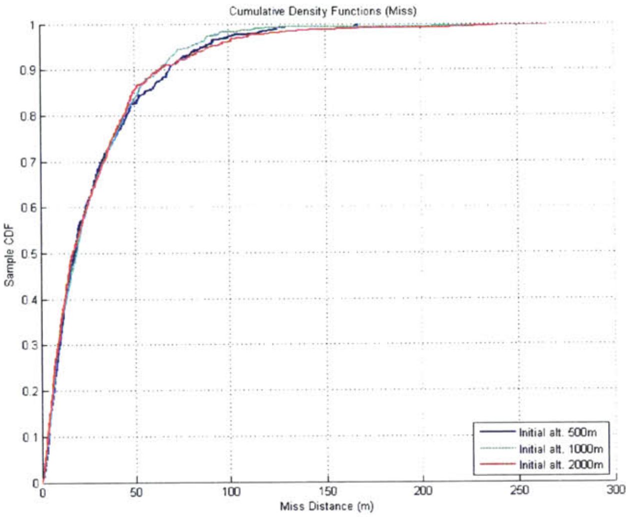


Figure 2-13: Miss distance CDF for Analytic CC-RRT from various initial altitudes, 500 trials

Table 2.8: Miss distance data for Analytic CC-RRT from various initial altitudes, 500 trials (in meters)

| **Init. Alt.** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| $p_{zI}$ = 500m | 28.2 | 28.5 | 18.0 | 45.7 | 68.6 | 86.5 | 115 | 167 |
| $p_{zI}$ = 1000m | 27.5 | 28.9 | 19.3 | 44.2 | 63.7 | 79.5 | 95.4 | 256 |
| $p_{zI}$ = 2000m | 28.6 | 33.2 | 17.0 | 44.0 | 63.3 | 91.2 | 122 | 266 |

altitude, $p_{zI}$, is varied from 500 m (as in Section 2.7.2) to 1000 m and 2000 m. All other conditions are the same as in Section 2.7.2.

From the results in Figure 2-13 and Table 2.8, the performance of Analytic CC-RRT is shown to be largely independent of the starting altitude. Compared to starting at 500 m, both the mean and worst-case accuracy increase only slightly at higher initial altitudes (Table 2.8). Again, the shapes of the CDF curves for all three cases are nearly the same (Figure 2-13). This data suggests that Analytic CC-RRT is capable of operating at higher altitudes without major deterioration in performance.

## 2.8 Summary

This chapter has presented a new approach to online trajectory planning and robust obstacle avoidance for the parafoil terminal guidance problem. The proposed Analytic Chance-Constrained Rapidly-exploring Random Trees (Analytic CC-RRT) algorithm has been shown to demonstrate several key strengths compared to existing approaches in the parafoil terminal guidance literature. By building on rapidly-exploring random trees (RRT), the proposed approach can quickly generate robust trajectory solutions with a high degree of geometric flexibility. Likewise, through the use of an intelligent cost-to-go function, the algorithm can operate from a wide range of initial conditions, and in particular has been shown to be largely invariant to initial altitude. The addition of speed penalties within the cost-to-go function allows for terminal heading conditions to be specified. Simulation results demonstrate that these penalties can significantly reduce parafoil landing speed in order to encourage upwind landings, with minimal effect on target accuracy. By utilizing a multi-class wind model trained on prior wind observations, the Analytic CC-RRT planner achieves robustness to highly dynamic wind conditions that cannot be obtained through replanning alone. Finally, a novel approach for sampling analytic state uncertainty distributions ensures robust collision avoidance with arbitrary, non-convex, mapped terrain, which is demonstrated through simulations to maintain real-time tractability.

# Chapter 3

# Chance-Constrained Band-Limited Guidance (CC-BLG)

## 3.1 Introduction

The primary contribution of this chapter is the development of a new algorithm for autonomous parafoil guidance which combines the benefits of analytic chance constraints and wind uncertainty modeling into an optimized trajectory planning framework. Leveraging the advantages in robust trajectory design produced by Analytic CC-RRT in Chapter 2, and the success of the state-of-the-art BLG algorithm for trajectory optimization, this new algorithm entitled Chance-Constrained Band-Limited Guidance (CC-BLG) is designed to overcome the limitations of both terminal guidance strategies in order to provide improved parafoil performance across a wide range of wind and terrain environments. This chapter begins with a discussion of motivating factors, followed by a description of the three core components of the CC-BLG algorithm. First, a method of weighted analytic uncertainty sampling is proposed to refine the estimated probability of constraint violation. Next, a novel risk-based objective function is developed for trajectory optimization. By directly incorporating the probability of constraint violation, this objective function enables the planner to efficiently balance the parafoil performance metrics of landing accuracy and landing speed, with the risk of off-nominal terrain collisions caused by future wind disturbances. Finally, the principles of discrete reachability set approximation are applied to the CC-BLG algorithm for robust obstacle detection, and avoidance, during fixed-horizon trajectory planning in constrained terrain environments. Through the design of an additional cost-to-go component, the reachability approximation accounts for the effect of the parafoil heading and the wind uncertainty in determining feasibility over future planning horizons.

## 3.2 Motivation

### 3.2.1 Analytic CC-RRT as a Parafoil Guidance Strategy

A principal motivation for the development of the Analytic CC-RRT algorithm presented in Chapter 2 is the widespread dependence of state-of-the-art parafoil guidance algorithms on a reactive replanning strategy to compensate for the effects of future wind disturbances [3, 4, 13, 32–34, 37]. The BLG algorithm, which relies solely on this approach, was found to be vulnerable when a failure to predict the consequences of off-nominal wind perturbations resulted in unanticipated collisions with the terrain environment. As indicated in Section 2.7.2, such collisions can occur even when the rate of replanning is high. Given these limitations, the proactive wind uncertainty model and covariance sampling technique developed through Analytic CC-RRT were shown to significantly improve planning robustness by anticipating the extent to which variable wind disturbances can increase risk. Building on the framework of rapidly-exploring random trees (RRT), the proposed algorithm was also shown to be efficient at generating robust trajectory solutions with high geometric flexibility in order to guide the parafoil toward the target location. These components enabled Analytic CC-RRT to provide superior mean and worst case landing performance when compared to BLG in challenging terrain scenarios.

Yet despite this success in planning robustness, one of the limitations of Analytic CC-RRT relative to state-of-the-art parafoil terminal guidance algorithms is the sub-optimal nature of the RRT-based trajectory design. Although the RRT algorithm is

Table 3.1: Miss distance comparison for Analytic CC-RRT on flat terrain with various samples per growth cycle, 500 trials (in meters).

| # Samples | Mean | StDev | 50% | 80% | 90% | 95% | 98% | Max |
|---|---|---|---|---|---|---|---|---|
| 150 | 27.5 | 29.0 | 17.2 | 43.1 | 67.3 | 83.5 | 101.7 | 236.1 |
| 300 | 23.7 | 23.3 | 16.1 | 41.3 | 54.7 | 67.9 | 86.2 | 143.5 |
| 500 | 22.6 | 21.1 | 17.3 | 43.2 | 54.2 | 63.2 | 69.2 | 109.8 |
| 1000 | 21.0 | 20.4 | 15.1 | 32.4 | 49.8 | 60.6 | 78.2 | 121.3 |

probabilistically complete, it has been shown in Ref. [66] that the lowest-cost RRT solution does not converge to the globally optimal cost solution, even as the number of state space samples approaches infinity. This is due, in part, to the way in which the RRT planning framework limits the capacity for trajectory refinement [67]. While most optimal planning algorithms maintain the ability to iteratively redesign the state trajectory in order to produce a lower cost solution, the random sampling and tree expansion techniques of RRT do not enable existing trajectories to be incrementally refined except through the addition of new trajectory nodes. As described in Ref. [67], this can potentially result in low-quality RRT trajectories that may be heavily biased by the initial tree growth.

The simulation results in Table 3.1 examine the change in Analytic CC-RRT landing accuracy over perfectly flat terrain as the number of samples per growth cycle of Algorithm 1 in Section 2.5 is increased. Regardless of the increase in the number of environmental samples and corresponding tree size, the results indicate that the RRT-based planner produces an average landing accuracy which is markedly inferior to that of an optimal planning algorithm such as BLG, shown in Table 2.5. As a result, one of the primary objectives of this chapter is to develop an autonomous parafoil guidance algorithm which incorporates the advantages of trajectory optimization for improved landing accuracy, while maintaining planner robustness to future wind uncertainty.

From the analysis presented in Chapter 2, the use of chance constraints to generate trajectories with guaranteed probabilistic feasibility bounds provided an effective framework for robust parafoil guidance in complex terrain. However, one component that is not included within the design of the Analytic CC-RRT algorithm is a method for explicitly minimizing the risk of constraint violation. In this way, the cost associated with paths that are only marginally feasible (i.e., $0 < p_{\text{collide}} \leq 1 - p_{\text{safe}}$) could be compared with the cost of paths that avoid terrain proximity completely (i.e., $p_{\text{collide}} = 0$). This concept has been explored for RRT-based algorithms in Ref. [67] by applying additional "soft constraints" to penalize risk within the objective function. If a path exists in the tree that is both risk free and terminates at the target location, the incorporation of risk-based penalties would allow the planner to identify this trajectory as the optimal cost solution. Unfortunately, given the asymptotically sub-optimal nature of the Analytic CC-RRT algorithm, the addition of risk-based penalties within the objective function cannot guarantee that the risk of constraint violation will be driven to zero over successive planning iterations [67]. In the absence of an optimization routine, this approach may therefore result in undesirable tradeoffs between incurred risk and guidance performance. The CC-BLG algorithm described in this chapter is designed to overcome these limitations by developing a cost model which can incorporate risk directly into the process of trajectory optimization.

An alternative approach to robust sampling-based motion planning known as CC-RRT* [67, 68] was also considered as a potential replacement for the Analytic CC-RRT algorithm. Building upon RRT* [69, 70], this algorithm extends the capabilities of RRT to provide asymptotic optimality guarantees for the lowest-cost trajectory solution, while using chance constraints to ensure probabilistic feasibility. Asymptotic optimality is achieved by iteratively rewiring paths within the tree in order to reduce the cost of each partial trajectory solution. Additionally, by penalizing risk within the CC-RRT* objective function, paths can be rewired to minimize the probability of constraint violation [67]. While these characteristics allow CC-RRT* to address the limitations of CC-RRT, several key challenges exist for the implementation of CC-RRT* to the parafoil terminal guidance problem. First, the tree rewiring procedure requires the development of a steering law in order to precisely connect any two states in space. Due to the complex and underactuated nature of parafoil dynamics, this steering law may be difficult to derive for higher order models. Even for the case of simplified parafoil dynamics, designing a trajectory between nodes that satisfies terminal constraints on both position and heading appears to be as difficult as solving the original two-point boundary value problem (TPBVP) between the parafoil's current state and the target [4]. Furthermore, the additional computation required to rewire CC-RRT* trajectories has been shown to increase average runtime relative to CC-RRT by more than a factor of 10 [67]. Due to the limited computational resources onboard most parafoil systems, this result may hinder the application of CC-RRT*. For these reasons, initial work on the CC-RRT* algorithm was abandoned in favor of the BLG trajectory optimization framework.

### 3.2.2 BLG as an Optimal Planning Framework

Given the motivations outlined in Section 3.2.1, this chapter considers the use of the Band-Limited Guidance (BLG) algorithm [32] as an optimal planning framework with which to incorporate robustness to the wind uncertainty. While several optimal planning approaches exist in the literature for parafoil terminal guidance [13, 30, 33], one of the primary advantages provided by the BLG algorithm is the formulation of parafoil terminal guidance as an unconstrained optimization problem. This formulation offers additional flexibility in complex terrain environments where it may be difficult to precisely satisfy the terminal boundary constraints. An additional benefit of the BLG algorithm, as described in Section 1.2, is its use of heading rate `sinc` functions for trajectory design. By specifying a heading rate frequency threshold that is sufficiently below the system bandwidth, this technique ensures accurate trajectory tracking by the controller while permitting the design of complex trajectory shapes. Lastly, the computational efficiency of the BLG algorithm makes it well suited for online trajectory optimization, and allows for a high rate of replanning in order to compensate for the effects dynamic wind disturbances [32]. These components of trajectory tracking and design are critical to the performance of a parafoil guidance algorithm in the presence of challenging terrain geometries [37].

One consideration that must be addressed in the implementation of the BLG algorithm is the issue of optimization scalability to states spaces of increasing size and dimensions. In the case of the parafoil terminal guidance problem, real-world terrain environments such as canyons and mountain ranges may require the initialization of terminal guidance from high altitudes in order to permit obstacle avoidance during the descent. As a result, the success of parafoil terminal guidance in these scenarios depends on the planner's ability to quickly and consistently converge upon an optimized trajectory solution given online planning constraints. In the BLG approach to parafoil terminal guidance, the issue of optimization scalability at high initial altitudes is addressed by dividing terminal guidance into two disjoint planning phases. This approach limits the size of the search space that must be considered during each terminal guidance phase, in order to both improve the rate of optimization convergence, and reduce the potential for local cost minima. Throughout this thesis, these two segments of BLG terminal guidance will be referred to as Phase 1 and Phase 2 respectively.

In Phase 1 of BLG terminal guidance, the planner uses the current best estimate of the mean wind $\overline{\mathbf{w}}$, in order to compute a desirable upwind transition point $\mathbf{p}_{\mathrm{G1}} = (p_{\mathrm{x,G1}}, p_{\mathrm{y,G1}}, p_{\mathrm{z,G1}})$. By designating this upwind location as the goal state for Phase 1 planning, the process of BLG trajectory optimization attempts to minimize the distance from the parafoil to the point $\mathbf{p}_{\mathrm{G1}}$ when the transition to Phase 2 terminal guidance begins. The transition between guidance Phase 1 and 2 is determined based on a preselected altitude, where the BLG trajectory plan that terminates at $\mathbf{x}_{\mathrm{F1}}$ for Phase 1 is reinitialized from the current parafoil state. With the exception of the initial Phase 2 state, the trajectory plans over both phases of terminal guidance are therefore disjoint. The BLG algorithm resumes the process of trajectory optimization during Phase 2 in order to guide the parafoil to the final target location, $\mathbf{p}_{\mathrm{G2}}$. This approach to terminal guidance enables trajectory planning and obstacle avoidance to begin from above the peak altitude of the terrain, while ensuring that the initial state for Phase 2 guidance is positioned upwind within the parafoil glide-slope cone [28].

In the design of the CC-BLG algorithm described in Section 3.3, this multi-stage formulation of terminal guidance is used for trajectory optimization in constrained terrain environments. The optimization of the CC-BLG cost function is based upon the method of Nelder-Mead simplex search, which agrees with the implementation of the BLG algorithm described in [32]. As the additional components of CC-BLG are introduced, the underlying structure from the existing BLG algorithm will remain unchanged. This allows for an accurate comparison to be drawn between the CC-BLG and nominal BLG algorithms in Chapter 5, wherein a series of Monte Carlo simulation experiments will be used to evaluate the performance of each parafoil guidance strategy.

## 3.3 Algorithm Development

### 3.3.1 Description of Approach

The approach to terminal guidance presented in this chapter is based upon the design of a novel risk-based objective function that enables the planner to efficiently balance the parafoil performance metrics of landing accuracy and landing speed, with the risk of off-nominal terrain collisions caused by future wind disturbances. This section describes the three core components of the CC-BLG algorithm which are derived through the application of chance constraints to enforce probabilistic feasibility within an optimized trajectory planning framework.

The wind model and state uncertainty distribution derived in Sections 2.3 and 2.4 are applied within the CC-BLG algorithm in order to estimate the probability of constraint violation at each time step. From the BLG parafoil kinematics provided in (1.2), we assume the 2-D wind estimates $w_x$ and $w_y$ may be represented as the combination of a deterministic mean wind $\overline{\mathbf{w}}$, and stochastic variational component $\delta\mathbf{w}$, according to the approach described in (2.17) and (2.21), respectively. In this way, the discrete-time variational dynamics model (2.42) can be used within CC-BLG to define the parafoil's state distribution at all future time steps, from the set of deterministic trajectory states in (1.2), and state covariance matrix $Q_t$ in (2.47). Applying the support vector machines (SVM) introduced in Section 2.3.3, the current best estimate of the wind is classified online into one of three variational models provided in Section 2.3.2. An augmented approach to analytic uncertainty sampling is then used to define a set of equi-spaced covariance samples along the state distribution, and check for collisions with the terrain environment. Probabilistic feasibility will also be evaluated at each time step through the chance constraint $\mathbb{P}_{\mathbf{v}}(\mathbf{x} \in \mathcal{X}) \geq p_{\text{safe}}$, provided in (2.9). This wind classification and uncertainty sampling technique borrowed from Analytic CC-RRT serves as one of the core robustness modifications to the BLG algorithm in order to approximate the probability of future collisions against arbitrary, non-convex terrain [46].

In the description of the CC-BLG algorithm presented below, although the method of generating analytic covariance samples will be similar to the approach provided in Chapter 2, the application of this technique to the trajectory generation process is fundamentally different. By incorporating additional cost penalties into the objective function for each covariance sample that violates the terrain constraints, CC-BLG will attempt to directly minimize the level of risk incurred at each time step along the trajectory in order to drive the planner towards solutions that are robust to future wind uncertainty. In Section 3.3.2, an alternative method of analytic uncertainty sampling is introduced in order to quantify the risk of constraint violation using the probability density function of the state distribution. Section 3.3.3 presents the CC-BLG objective function, which is used to incorporate the risk of constraint violation directly into the process of trajectory optimization. Lastly, Section 3.3.4 discusses some of the additional challenges that must be addressed during multi-stage CC-BLG path planning in complex terrain environments, such that vehicle safety can be maintained over consecutive planning horizons.

Within the context of a robust motion planner, the hard constraint bounds on probabilistic feasibility $p_{\text{safe}}$ in (2.9) can potentially limit the solution space of valid trajectories, and must be considered carefully during the process of algorithm design. This result has been demonstrated in the work by Luders et al. [67] through the analysis of several variants of both the CC-RRT and CC-RRT* algorithms over a range feasibility conditions specified by the user. As a result, this thesis first considers relaxing the hard feasibility constraints (2.9) during the development of the CC-BLG algorithm, in favor of using "soft," cost-based chance constraints to penalize the risk of constraint violation. Although probabilistic feasibility guarantees will be lost, this approach is shown to provide additional flexibility for optimized trajectory planning in tight geometry environments, and better coincides with the model of unconstrained optimization used within BLG. Chapter 4 examines the effects of reintroducing hard constraints on probabilistic feasibility in combination with the penalty-based formulation presented in Section 3.3.3.

### 3.3.2 Weighted Analytic Uncertainty Sampling

Based on the principles of analytic uncertainty sampling introduced in Chapter 2, this section presents a new method for approximating the probability of constraint violation at each time step by developing an alternative set of weighted covariance samples along the state distribution. These sample weights are derived using the probability density function (PDF) of the bivariate normal distribution, which represents the parafoil's $\delta\mathbf{p}_x$ and $\delta\mathbf{p}_y$ position about each nominal trajectory state. For a vector $\mathbf{x}$ of arbitrary length, with vector mean $\bar{\mathbf{x}}$ and covariance $\Sigma$, the general form of the multivariate Gaussian PDF is expressed as

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right). \tag{3.1}$$

Recall that when the value of $(\mathbf{x}-\bar{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\bar{\mathbf{x}})$ is constant, this distribution can be used to characterize contours of equal probability—represented in 2-Dimensions by the uncertainty ellipses introduced in Section 2.4. For the bivariate normal distribution of $(\delta\mathbf{p}_x, \delta\mathbf{p}_y)$, the boundary of each uncertainty ellipse with covariance $Q$ is described in the $XY$-frame by

$$c = \frac{\delta p_x^2}{\sigma_x^2} - \frac{2\rho\delta p_x\delta p_y}{\sigma_x\sigma_y} + \frac{\delta p_y^2}{\sigma_y^2} \tag{3.2}$$

relative to the nominal trajectory state $\widehat{\mathbf{x}}$, where $\rho$ is the correlation coefficient and $c$ is a constant value.

In the previous application of analytic uncertainty sampling provided in Chapter 2, the probability of constraint violation at a given time step was determined using

(2.53), as the fraction of total covariance samples that violate the terrain constraint, $p_{z,t} \leq T(p_{x,t}, p_{y,t})$. By approximating probabilistic feasibility in this way, all covariance samples are assigned an equal probability weight of $P(S) = 1/N_S$, where $N_S$ is the total number of covariance samples positioned around each trajectory state. While this method has been shown to provide an efficient estimate of collision probability using a small number of covariance samples, such an approach does not take into account the variation in the probability distribution of $(\delta p_x, \delta p_y)$ as the distance from the expected trajectory state, $\widehat{\mathbf{x}}$ is increased. In this chapter, we seek a higher fidelity probability estimate for the CC-BLG algorithm which can better differentiate the relative levels of risk over the parafoil state distribution as the number of covariance samples and ellipses are increased. This probability estimate is used in Section 3.3.3 to improve the quality of risk minimization along the parafoil trajectory plan.

The proposed approach to weighted uncertainty sampling presented in this section considers reassigning probability weights,

$$\begin{aligned} P(S_{ij}) &= \frac{F(\sigma_j)}{N_{S_j}}, && \text{for } \ i \in \{1, ..., N_{S_j}\}, \ \ j \in \{1\} \\ P(S_{ij}) &= \frac{(F(\sigma_j) - F(\sigma_{j-1}))}{N_{S_j}}, && \text{for } \ i \in \{1, ..., N_{S_j}\}, \ \ j \in \{2, ..., (N_E - 1)\} \\ P(S_{ij}) &= \frac{(1 - F(\sigma_{j-1}))}{N_{S_j}}, && \text{for } \ i \in \{1, ..., N_{S_j}\}, \ \ j \in \{N_E\} \end{aligned} \tag{3.3}$$

to each covariance sample which are proportional to the region of the uncertainty distribution contained within each covariance ellipse. For any number of $\sigma_j$-ellipses $j = \{1, ..., N_E\}$, and samples per ellipse $N_{S_j}$, this method divides the probability mass inside each ellipse among the number of covariance samples positioned both along and within its boundary. From these weighted covariance samples, an estimate of the probability of collision at a given time step will be developed by approximating the integral of the probability density function $f_{\mathbf{X}}(\delta p_x, \delta p_y)$ over the region of the uncertainty distribution that intersects the terrain.

Through the conic relaxation $\delta p^T Q^{-1} \delta p = 1$, presented in Section 2.4.2, the probability that a point $(\delta p_x, \delta p_y)$ is located within an uncertainty ellipse with covariance

$Q$, and covariance scale factor $\sigma$, may be represented by the CDF

$$F(\sigma) = 1 - e^{-\sigma^2/2} \tag{3.4}$$

of a Chi-squared distribution with two degrees of freedom, $\sigma^2 \sim \mathcal{X}_2{}^2$ [71]. The probability weights $P(S_{ij})$ in (3.3) are equal for all covariance samples $i \in \{1, ..., N_{S_j}\}$ along the boundary of a particular $\sigma_j$-ellipse, and the covariance samples are positioned at equally spaced intervals. In this way, each $i$th sample along ellipse $\sigma_j$ can be used to represent an equal area, $A_{ij}$, of the total probability distribution. The integral of the PDF $f_{\mathbf{X}}(\delta p_x, \delta p_y)$ over each $A_{ij}$ partition results in an equivalent expression for the probability weights,

$$P^* = P(S_{ij}) = \int\int_{A_{ij}} f(\delta p_x, \delta p_y) d\delta p_x d\delta p_y. \tag{3.5}$$

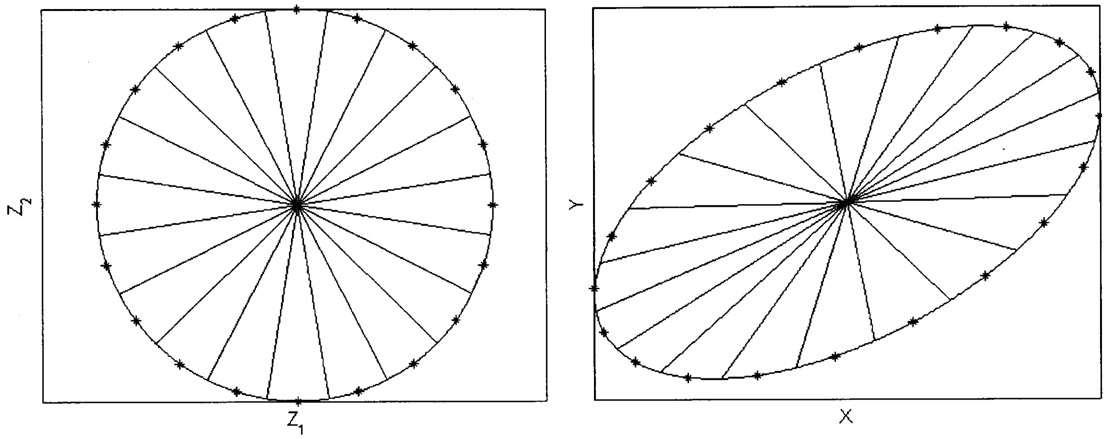
Figure 3-1 demonstrates how the discretization of the probability distribution within a given $\sigma_j$-ellipse is used to assign the corresponding weights for each covariance sample. Here we recognize that any Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma)$, can be transformed into a zero mean Gaussian random vector $\mathbf{z} \sim \mathcal{N}(0, I)$, with identity covariance matrix, $I$, through the linear transformation

$$\mathbf{z} = (\sqrt{\Sigma})^{-1}(\mathbf{x} - \bar{\mathbf{x}}). \tag{3.6}$$

In the same way, the reverse transformation can be applied to specify a Gaussian random vector with mean $\bar{\mathbf{x}}$ and covariance $\Sigma$ according to

$$\mathbf{x} = \bar{\mathbf{x}} + \sqrt{\Sigma}\,\mathbf{z}. \tag{3.7}$$

Given this transformation, Figure 3-1(a) illustrates that covariance samples can be placed at equally spaced angular intervals in the variables $Z_1$ and $Z_2$ for a distribution $\mathbf{z} \sim \mathcal{N}(0, I)$ using (2.49), where the angle bisectors between covariance samples serve to divide the 1-$\sigma$ unit circle into equal areas. Applying (3.7), the location



(a) standard normal ellipse $(Z_1, Z_2)$ (b) example covariance ellipse $(X, Y)$

Figure 3-1: Transformation of 2D uncertainty ellipse from $\mathbf{z} \sim \mathcal{N}(0, I)$ to $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma)$

of the transformed covariance samples and their bisectors can be determined in the $XY$-plane along any uncertainty ellipse with covariance matrix $\Sigma$, such that equal areas are maintained in the space between samples. These areas in Figure 3-1 correspond to each region $A_{ij}$ of the probability distribution. Figure 3-1(b) displays this transformation from $(Z_1, Z_2)$ to a 1-$\sigma$ ellipse in the $XY$-plane.

Because Gaussian distributions are preserved through linear transformations, it follows from (3.7) that the probability masses $P^*$, represented by the integral (3.5) over each area partition $A_{ij}$ in the variables $(Z_1, Z_2)$ or $(X, Y)$, are equivalent for the 1-$\sigma$ ellipses drawn in Figure 3-1(a) and Figure 3-1(b), respectively. Such a relationship holds for any two uncertainty ellipses in coordinates $\mathbf{z}$ and $\mathbf{x}$ of equal standard deviation/covariance scale factor, $\sigma$. This result allows (3.4) to be used to determine the probability mass $F(\sigma)$ within any $XY$ uncertainty ellipse, $\sigma_j$, in terms of the identical probability mass within the corresponding $\sigma_j$-circle of the standard normal distribution $\mathbf{z} \sim \mathcal{N}(0, I)$—wherein $\sigma$ conveniently represents both the Euclidean and Mahalanobis distance $\sigma = \sqrt{Z_1^2 + Z_2^2} = \sqrt{(\mathbf{z} - \bar{\mathbf{z}})^T I^{-1} (\mathbf{z} - \bar{\mathbf{z}})}$ in the $(Z_1, Z_2)$ frame. To understand how the probability mass within each covariance ellipse changes as a function of $\sigma$ (i.e., distance from the mean), Figure 3-2 displays the CDF graph of $F(\sigma)$ as $\sigma$ is varied over the interval $\{0, 6\}$. From Figure 3-2 it can be seen that the probability $F(\sigma)$ increases most rapidly in the vicinity of the mean for $\sigma < 2$, such

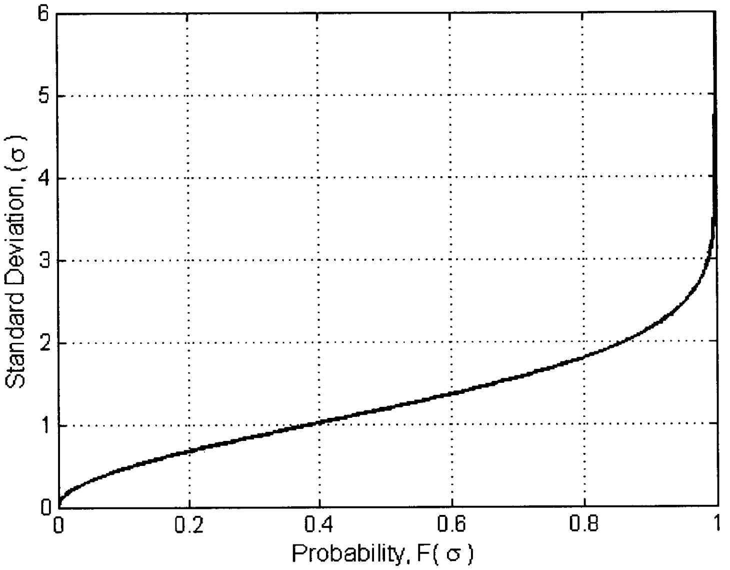


Figure 3-2: Chi-squared cumulative distribution function (CDF), $F(\sigma)$

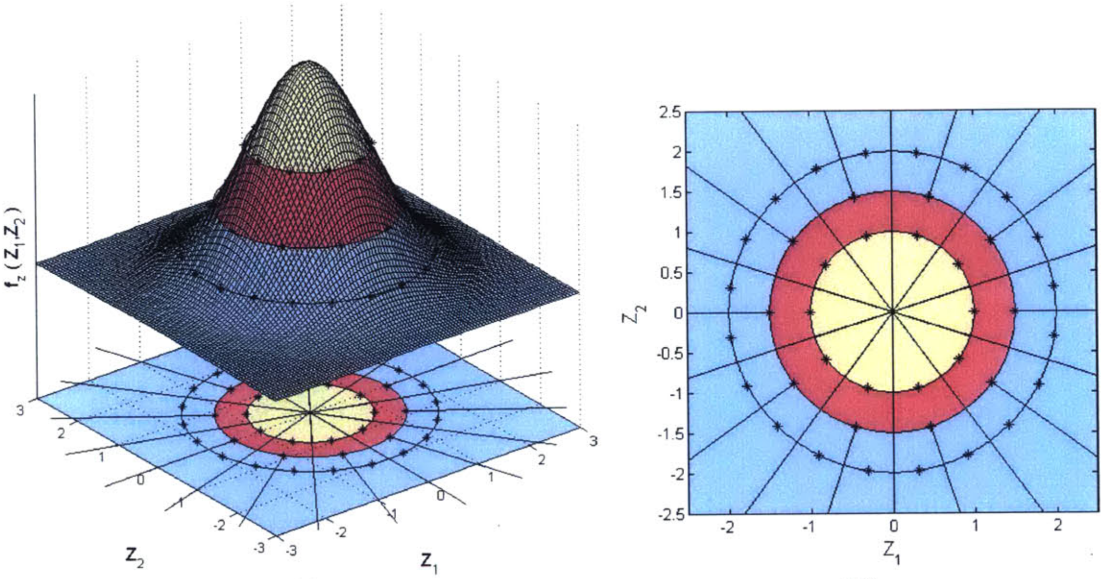
that the distribution $(\delta p_x, \delta p_y)$ is non-uniform across varying levels of $\sigma$.

### Probability of Terrain Collision

By applying the method outlined above, the resulting probability of collision at each time step along the parafoil trajectory is approximated as the sum of the probability weights, $P(S_{ij})$ for all covariance samples which violate the terrain constraint, $p_{z,t} \leq T(p_{x,t}, p_{y,t})$, where

$$p_{collide} = \begin{cases} \sum\limits_{j=1}^{N_E} \sum\limits_{i=1}^{N_{S_j}} P(S_{ij})\mathbb{I}(S_{ij}) & , \quad \text{for } \mathbb{I}(\widehat{\mathbf{x}}) = 0 \\ 1 & , \quad \text{for } \mathbb{I}(\widehat{\mathbf{x}}) = 1 \end{cases} . \tag{3.8}$$

To demonstrate how this estimate can be used, consider the case of three covariance ellipses for $\sigma = 1$, 1.5, and 2 respectively, which are positioned about a nominal trajectory state. In this example, a set of 10 equally spaced covariance samples are placed along the boundary of both the 1-$\sigma$ and 1.5-$\sigma$ ellipses, and 20 covariance samples are placed along the boundary of the outermost 2-$\sigma$ ellipse. Figure 3-3 represents this set of $\sigma$-ellipses and covariance samples in the $(Z_1, Z_2)$ frame. As



(a) (b)

Figure 3-3: Discretization of $f_{\mathbf{z}}(Z_1, Z_2)$ for covariance samples within each $\sigma$-ellipse

described above, the total probability distribution is partitioned into sections whose integrals correspond to the probability weights $P^* = P(S_{ij})$ associated with each covariance sample. Using (3.4), the probability that the parafoil is located within the region of the 1-$\sigma$ ellipse can be computed directly as $F(\sigma) = 0.393$. This is represented by the integral of $f_{\mathbf{z}}(Z_1, Z_2)$ over the yellow region of the total probability distribution shown in Figure 3-3(a). By equally dividing the probability mass of the 1-$\sigma$ ellipse among the 10 covariance samples positioned along its boundary, the corresponding probability weight assigned to each sample is 0.0393. The result signifies that for the discretization described above, the estimated probability of collision in (3.8) will increase by 0.0393 for each covariance sample in the 1-$\sigma$ ellipse which violates the terrain constraint. This approach differs from Section 2.5, in which all 40 covariance samples would have been assigned an equal probability weight of $1/N_S = 0.025$.

The probability region represented by covariance samples along the second 1.5-$\sigma$ ellipse is $F(1.5) - F(1) = 0.282$. This region of the probability distribution is shown by the red area in Figure 3-3, in which the overlapping probability region inside the 1-$\sigma$ ellipse has been removed from the total area of the 1.5-$\sigma$ ellipse. In this way, the 10 covariance samples along the 1.5-$\sigma$ ellipse are each assigned a probability weight of

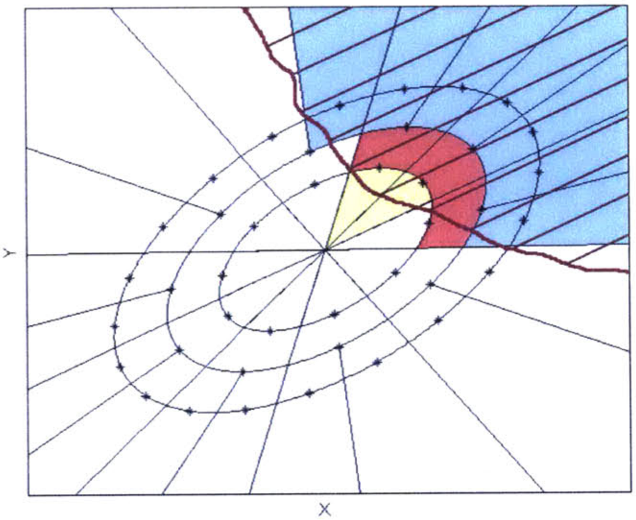


Figure 3-4: Example of probability region captured by collided covariance samples

0.0282. To complete the discretization, the remaining probability region of $1 - F(1.5)$ is divided equally among the 20 covariance samples along the third 2-$\sigma$ covariance ellipse, as shown by the blue area in Figure 3-3, where each covariance sample is assigned a weight of 0.0162 according to (3.3)–(3.4).

With this arrangement of probability weights, Figure 3-4 illustrates how the probability of collision is determined in the presence of terrain. In this example, 12 of the covariance samples have intersected the terrain boundary represented by the brown hashed region in Figure 3-4: 2 from the 1-$\sigma$ ellipse, 3 from the 1.5-$\sigma$ ellipse, and 7 from the 2-$\sigma$ ellipse, for a total estimated probability of collision of 0.277.

By representing the contours of the state uncertainty distribution using covariance samples, the parafoil proximity to terrain along the nominally propagated trajectory can only be estimated from the location of covariance samples which intersect the terrain boundary. As a result, Figure 3-4 demonstrates that the terrain proximity of the expected state, $\widehat{\mathbf{x}}_t$, will always be less than or equal to the distance $d = \sqrt{\delta p_{x,t}^2 + \delta p_{y,t}^2}$ to any collided covariance samples. In order to account for this effect, the probability weights described by (3.3) represent the integral over the maximum obstructed region of the probability distribution which can exist between the boundaries of the remaining collision free samples. This can be observed in Figure 3-4 by the yellow probability region captured within the boundaries of the two collided covariance samples in the innermost 1-$\sigma$ ellipse. The formulation of (3.3) therefore provides a favorably conservative measure of the probability of collision at each time step of the trajectory, and helps to ensure the CC-BLG algorithm will not underestimate the level of environmental risk. This level of conservatism approaches zero as the number of covariance samples and covariance ellipses increases to infinity, until the weight of each covariance sample can be expressed by the value $f(\delta p_{x,t}, \delta p_{x,t}) d\delta p_{x,t} d\delta p_{y,t}$. The sum of probability weights $P(S_{ij})$ over all collided covariance samples then converges to the true value of the CDF integral over the region of the distribution which intersects the terrain. As a result, this implementation of weighted analytic uncertainty sampling produces a higher fidelity model of the risk incurred due to future wind disturbances than the previous Analytic CC-RRT approach (2.53).

### 3.3.3 Risk-based Trajectory Optimization

Central to the development of the CC-BLG algorithm is the design of a risk-based objective function for enforcing the probabilistic feasibility of planned parafoil trajectories. The terms of this objective function are selected to transform the probability of constraint violation at each time step into a discrete cost penalty by utilizing both the uncertainty sampling technique outlined in Section 3.3.2, and a novel variant of the cost-to-go model presented in Section 2.6.

The cost function for the CC-BLG algorithm is designed to satisfy three main objectives for trajectory optimization and collision avoidance in constrained terrain environments. First, the cost associated with each trajectory should adequately reflect the discrete probability of a terrain collision at time step $t$ during the parafoil's descent. Second, the augmented cost penalty for the risk incurred at each trajectory state should increase proportionally with the nominal BLG costs associated with that state. In this way, the penalty on risk can be used to forecast the BLG terminal state cost that would result from a terrain collision at each point in the trajectory, so as to discourage potentially high cost maneuvers. Finally, in order to account for the underactuated nature of the parafoil dynamics, the planner should prioritize the minimization of short horizon risk over longer horizon risk. This reflects the notion that, when necessary, the planner should take corrective action to account for imminent danger before attempting to minimize long-term trajectory risk.

Given the design objectives outlined above, the cost function $J_T$ associated with each state of the parafoil trajectory takes the form

$$J_T = \begin{cases} \sum_{j=1}^{N_E} \sum_{i=1}^{N_{S_j}} \beta_{ij} P(S_{ij}) \mathbb{I}(S_{ij}) & , \quad \text{for } \mathbb{I}(\widehat{\mathbf{x}}) = 0 \\ p_{collide}\beta & , \quad \text{for } \mathbb{I}(\widehat{\mathbf{x}}) = 1 \end{cases} . \tag{3.9}$$

The estimated probability of constraint violation at each time step is incorporated into the penalty function $J_T$ through the sum over all covariance sample probability weights $P(S_{ij})$ for which $\mathbb{I}(S_{ij}) = 1$ indicates a terrain intersection has occurred. Likewise, in the case that the nominal planned trajectory state $\widehat{\mathbf{x}}_t$ has intersected the terrain (i.e., $\mathbb{I}(\widehat{\mathbf{x}}_t) = 1$), the probability of collision $p_{collide}$ is taken to be 1, as described previously in (3.8). The base probability weight of each covariance sample in (3.9) is then scaled by the additional weight factor,

$$\beta = \frac{\sqrt{(p_x - p_{x,G})^2 + (p_y - p_{y,G})^2 + (p_z - p_{z,G})^2}}{d_{norm}} = \frac{||(\Delta p_x, \Delta p_y, \Delta p_z)||}{d_{norm}} \tag{3.10}$$

representing the relative distance between the location of the sample $(p_x, p_y, p_z)$, and the location of the target $(p_{x,G}, p_{y,G}, p_{z,G})$.

To simplify the expression of the state cost, the parameter $d_{norm}$ in (3.10) acts as a normalization factor for the 3D Euclidean distance $||(\Delta p_x, \Delta p_y, \Delta p_z)||$ which bounds the cost of each covariance sample $S_{ij}$ within the range $\{0, 1\}$. This parameter can be tuned empirically for various parafoil applications in order to estimate the maximum 3D distance the parafoil will travel from the target during the terminal guidance phase. Throughout this work, a value of $d_{norm} = 750$ meters will be used which, in practice, successfully normalized the value of $\beta$ over most parafoil trajectory plans.

Scaling the base probability weights $P(S_{ij})$ by the distance factor $\beta$, the expression for the penalty on risk $J_T$ in (3.9) serves to efficiently forecast the nominal BLG cost of a collision at each point in the trajectory plan. Recall that for the BLG objective function outlined in Section 1.2, the cost associated with each trajectory is depicted as the weighted sum of quadratic costs associated with the position error, $\Delta p_x^2$, $\Delta p_y^2$, $\Delta p_z^2$, and heading error, $\sin(\Delta\psi/2)^2$, at the terminal trajectory state. Because the parafoil's distance from the target is one of the most relevant metrics when determining the cost of an unintended terrain collision, the component $\beta$ in (3.9) reflects the notion that risk incurred in situations where a collision is least desirable (i.e., furthest from the target) should receive a higher penalty than risk incurred in the vicinity of the target. This approach also acts as a heuristic to gradually decrease the total penalty on constraint violation as the parafoil approaches the target—thus shifting planning priority toward landing accuracy over risk aversion at lower altitudes.

The final cost function $J'$ associated with each CC-BLG trajectory is selected as the combination of the nominal BLG terminal state cost $J_{BLG}$ in (1.3), and the weighted sum $J_{CC}$ over all state penalties in the trajectory $J_{T,k}$ for $k = \{1, ..., N_T\}$,

$$J_{CC} = \sum_{k=1}^{N_T} \alpha_k J_{T,k} = \sum_{k=1}^{N_T} \left( \alpha_k \left\{ \sum_{j=1}^{N_E} \sum_{i=1}^{N_{S_j}} \beta_{ij} P(S_{ij}) \mathbb{I}(S_{ij}) \right\} \right), \tag{3.11}$$

$$\alpha_k = \alpha_0 e^{-\lambda|p_{z,0}-p_{z,k}|}, \tag{3.12}$$

$$J' = J_{CC} + J_{BLG}. \tag{3.13}$$

In this expression for the augmented $J_{CC}$ cost component, the minimization of short horizon risk is prioritized through the addition of the scale factor $\alpha_k$, which decays exponentially as a function of the altitude difference between the parafoil's current state $p_{z,0}$, and the altitude of each trajectory state $p_{z,k}$. The rate of decay is selected through the parameter $\lambda$, specifying the extent to which risk is penalized along the trajectory. For $\lambda > 0$, the scale factor $\alpha_k$ is largest at the current state, and smallest at the terminal trajectory state.

Given this relationship, if the risk of imminent terrain collision is high, the associated penalty on states $J_{T,k}$ dominate the $J'$ cost function and encourages the planner to make an immediate course correction so as to minimize the penalty of near term states. Additionally, by decaying the cost $J_{T,k}$ exponentially through the rate parameter $\lambda$, states at lower altitudes (i.e., longer time horizons) relative to the current parafoil state will tend to have a much smaller impact on the cost minimization. Because the CC-BLG algorithm updates the trajectory plan every $\Delta t = 1$ second in order to compensate for the effects of modeling errors and changing wind conditions, the trajectory produced at the end of each planning cycle may change considerably during the course of the parafoil's descent. This choice for the weight coefficient $\alpha_k$ accounts for the additional time the planner has in order to minimize risk at lower altitudes, by reducing the penalty associated with trajectory states that are most susceptible to change due to subsequent planning iterations.

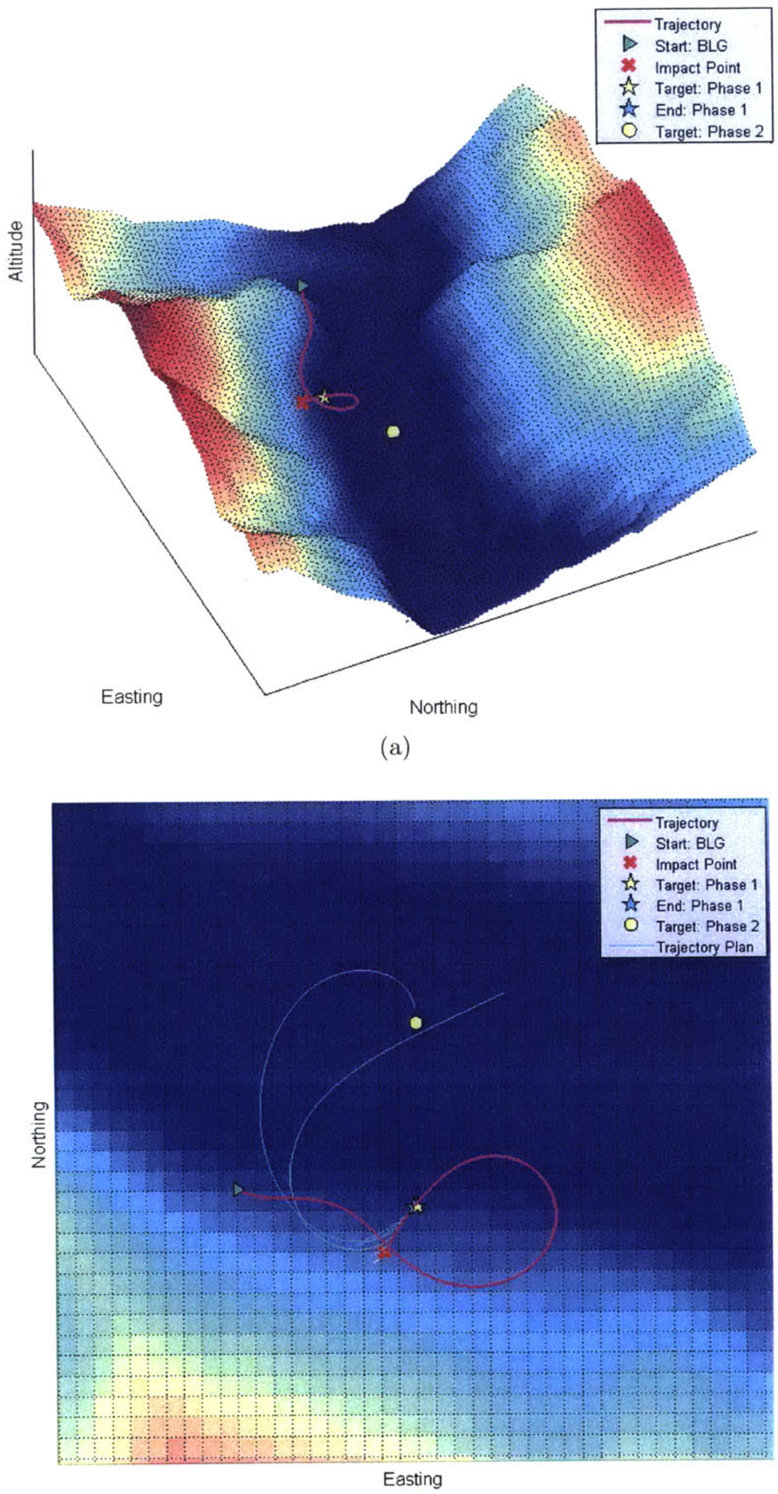
The relative weighting between the objectives of terminal landing accuracy and risk mitigation in (3.13) is determined through the selection of the constant coefficient $\alpha_0$ in (3.12), which scales the cumulative penalty $J_{CC}$ over all states in the planned trajectory. In many multi-objective optimization problems, the selection of optimization priority in the case of conflicting objectives is largely dependent on user preference, as expressed through the relative weight of each component in the final cost function. Through the formulation of (3.11), the minimization of risk at each time step along the trajectory need not conflict with the minimization of the $J_{BLG}$ terminal state cost. These objectives can both be satisfied if the feasible planning space is sufficiently large at all altitudes so as to include the region of the uncertainty distribution (3.2) contained within each state's largest covariance ellipse. However, as the degree of wind uncertainty or environmental hazards is increased, some level of risk may be required in order to minimize the $J_{BLG}$ cost objective associated with landing at the target, or orienting the parafoil into the wind. Thus, $\alpha_0$ serves a similar purpose to the user defined feasibility bound $p_{safe}$ introduced in Analytic CC-RRT, as a means to specify the user's desired level of risk tolerance while attempting to satisfy the terminal state objective. Chapter 4 describes the process of tuning the CC-BLG cost parameters of $\alpha_0$, $\lambda$, and the number of covariance samples $N_S$, in order to produce the desired balance between planner robustness and goal seeking behavior used throughout this thesis.

### 3.3.4 Reachability Approximation for Obstacle Avoidance

In Section 3.2.2, a high altitude variant of the BLG terminal guidance algorithm was introduced which divides path planning into two distinct phases for trajectory optimization. This multi-stage planning approach enables BLG to overcome some of the challenges related to optimization scalability over the range of initial altitudes common for parafoil terminal guidance, but can also create a potentially dangerous planning scenario in the presence of environmental obstacles. The threat to vehicle safety can increase when partial trajectory plans designed over limited time horizons do not adequately consider future feasibility requirements beyond each horizon's terminal trajectory state [72]. As a result, satisfying the feasibility constraints over the current planning horizon may not be sufficient to guarantee path feasibility during future time steps.

Figure 3-5 demonstrates how a failure to account for the continuation of Phase 2 planning during the design of the Phase 1 trajectory can result in an unanticipated collision with the terrain environment. This collision is caused by the direction of the parafoil approach to the upwind Phase 1 target $(p_{x,G1}, p_{y,G1}, p_{z,G1})$, resulting in a pathological initial condition for Phase 2 terminal guidance. Although the initial state for Phase 2 terminal guidance (blue star) does not violate the terrain constraints, the parafoil has insufficient time and control authority to avoid a collision with the terrain during subsequent planning iterations. This is illustrated in Figure 3-5(b) by the attempted BLG trajectory plans during the initial 5 seconds of Phase 2 terminal guidance in the valley terrain environment, introduced in Section 4.2.

In this section, the principles of discrete reachability set approximation introduced in Chapter 2 are extended to the process of multi-stage CC-BLG path planning. The proposed approach will provide an efficient strategy for detecting the presence of terrain obstacles beyond the current planning horizon in order to maintain trajectory feasibility, and mitigate the risk of future terrain collisions. This is achieved by reorienting the parafoil approach to the Phase 1 target location through the design of an additional risk-based cost-to-go function $J_{CTG}$, which considers the role of both



(a)

(b)

Figure 3-5: BLG collision during terminal guidance phase transition

the parafoil heading, and the wind uncertainty, in determining the feasibility of the final trajectory design.

In Section 2.6, the full reachability set of the parafoil was approximated over finite time using a set of $N_u$ constant control sequences, constrained within the limits $|u| < \omega_{max}$. Approximating the reachability set in this way, a method for detecting the presence of terrain obstacle over future planning horizons is incorporated into the CC-BLG algorithm by creating a discrete boundary of states $(p_{x_n}, p_{y_n}, p_{z_n})$ for $n \in \{1, ..., N_u\}$ which are propagated forward in time from the terminal state $\widehat{\mathbf{x}}_F$ of each Phase 1 candidate trajectory. For the set of $N_u$ equi-spaced constant control sequences defined in (2.62), the location of each state along the reachability horizon can be specified through the forward propagation of the parafoil dynamics model provided in (2.32)–(2.36) over the finite time interval $t = \{t_F, t_F + \tau\}$. Here, $t_F$ is the time at the final state $\widehat{\mathbf{x}}_F$ of the planned Phase 1 trajectory, and the propagation time $\tau$ will be taken as the minimum between the time to first terrain impact $t_Z$, and the finite propagation horizon $t_P$ selected by the user,

$$\tau = \min(t_Z, t_P). \tag{3.14}$$

By applying the dynamic propagation using (2.32)–(2.36), the reachability horizon $t_P$ can be extended to enable the planner to look further forward in time while retaining predictive accuracy regarding the location and feasibility of future parafoil states. This results in a better approximation of the reachability set than the previous analytic expression described in (2.63), in which both the parafoil lag dynamics and persistent wind estimate were neglected for simplicity over short propagation horizons. Although this improvement in model accuracy incurs an additional increase in computation time, in practice, this application of reachability set approximation remains efficient, and requires far fewer function evaluations when compared to the implementation provided in Analytic CC-RRT.

To account for the effects of possible variation in the future wind disturbances, the state uncertainty distribution about each nominal trajectory of the reachability

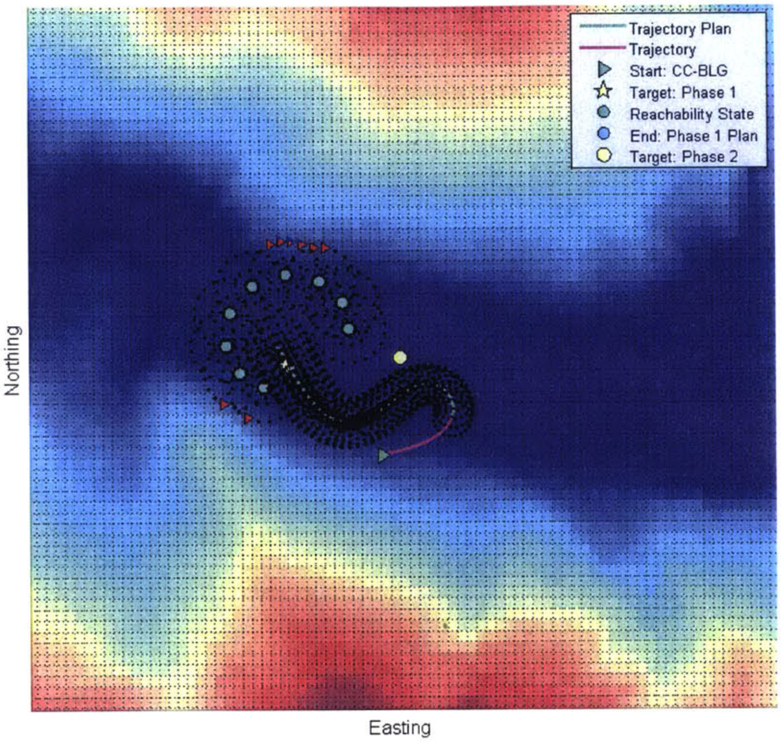


Figure 3-6: Example of reachability set approximation during Phase 1, $N_u = 9$

set is computed using (2.44) through the discrete time propagation of the covariance matrix, $Q_t$. In this way, (2.48) and (2.49) can be used to define a set of covariance samples at each time step in order to check the set of $N_u$ parafoil trajectories for the presence of terrain obstacles, beginning from the last state $\widehat{\mathbf{x}}_F$ of the current planning horizon. Figure 3-6 displays the states (green circles) at time $t = t_F + t_P$ along the reachability frontier for $N_u = 9$, which are propagated forward in time from the terminal state (light blue circle) of the Phase 1 trajectory plan. The Phase 1 target location is indicated by a yellow star. The black dots around each frontier reachability state in Figure 3-6 indicate the location of the covariance samples along the state distribution, whereas the red triangles indicate the location of covariance samples which have intersected the terrain. Covariance samples are also shown along the nominal Phase 1 trajectory plan (green line).

With the reachability approximation provided above, a form for the cost-to-go function $J_{CTG}$ is selected so as to compare potential position and heading configurations at the end of Phase 1 trajectory planning. To enforce that the current trajectory plan remains feasible over future planning iterations, a desirable form of the cost-to-go function is one that attempts to minimize the risk of terrain collision represented by the set of reachability states. Utilizing the risk model provided in (3.9), the cost-to-go function

$$J_{CTG} = \xi \sum_{n=1}^{N_u} \left( \sum_{k=1}^{N_T} \alpha_{k,n} J_{T_{k,n}} \right) = \xi \sum_{n=1}^{N_u} J_{R,n} \tag{3.15}$$

combines the weighted sum over the risk penalties of all reachability states $J_{T_{k,n}}$, for a given number of input sequences $n = \{1, ..., N_u\}$, and propagation steps per trajectory $k = \{1, ..., N_T\}$. The $J_{CTG}$ cost-to-go function is augmented to the CC-BLG objective function (3.11) during Phase 1 terminal descent. Within the resulting multi-objective optimization

$$J' = \begin{cases} J_{CC} + J_{BLG} + J_{CTG} & , \quad \text{for } p_{z,0} > p_{z,G1} \\ J_{CC} + J_{BLG} & , \quad \text{for } p_{z,0} \leq p_{z,G1} \end{cases} , \tag{3.16}$$

the constant coefficient $\xi$ in (3.15) serves as an additional tuning parameter, which implicitly specifies the importance of the final Phase 1 approach direction $\psi_F$ through the weighted cost of the reachability set approximation. If the current approach direction and wind classification indicate a high degree of risk in attempting to reach the Phase 1 target, the optimizer will attempt to redesign a trajectory which terminates in either a different orientation $\psi_F$, or with sufficient distance from the target, in order to improve the safety of the reachability states. The purpose of the cost-to-go function in (3.16) is therefore to ensure that the boundary of the reachability set is collision free over the range of control $\{-\omega_{\max}, \omega_{\max}\}$, such that the nearest terrain obstacle falls outside the horizon specified by $t_P$. Section 4.4 considers the selection of the propagation limit $t_P$, and number of control sequences $N_u$, in order to produce the desired behavior.

By formulating the cost-to-go function in this way, the parafoil approach to the Phase 1 target location is considered cost optimal if all $N_u$ control sequences incur no risk over the finite horizon $t = \{t_F, t_F + \tau\}$. Such a configuration of reachability states is advantageous by providing the greatest feasible planning space ahead of the parafoil during the initial iterations of Phase 2 terminal guidance. While alternative formulations of the cost-to-go function are available (e.g., minimum and maximum functions), this relatively straightforward approach in (3.15) emphasizes the desire for the entire finite horizon reachability set to be collision free, and does not assume the planner's ability to immediately identify and execute the lowest cost input sequence. As a result, this formulation of the cost-to-go function emphasizes vehicle safety, and is more conservative than the previous formulation provided in Section 2.6. The improvement of (3.15) for promoting future planning feasibility is demonstrated in Section 4.4, which considers alternative forms of the cost-to-go function.

## 3.4 Summary

This chapter presented the three core components of the CC-BLG parafoil terminal guidance algorithm. This algorithm builds upon the framework of Band-Limited Guidance (BLG) [32] by combining the benefits of wind uncertainty modeling and reachability set approximation for improved planning robustness in complex terrain and wind environments. First, a new approach to weighted analytic uncertainty sampling was introduced in order to efficiently approximate the probability of constraint violation at each time step. This method leverages the information within the state bivariate normal distribution to provide a higher fidelity estimate of the risk produced by future wind uncertainty than the previous analytic sampling approach (i.e., Section 2.4). Second, a novel risk-based objective function was developed for the CC-BLG algorithm which incorporates the probability of constraint violation directly into the process of trajectory optimization. In this way, CC-BLG trajectory plans are optimized for both landing accuracy and reduced landing speed, while minimizing the risk of off-nominal terrain collisions caused by future wind disturbances. Finally, the method of reachability set approximation introduced in Section 2.6 was applied to the process of multi-stage CC-BLG path planning. Through the design of an additional cost objective, this technique enables obstacle detection and avoidance between fixed planning horizons by considering the effects of the parafoil heading and wind uncertainty on future planning feasibility. Chapter 4 presents the analysis and tuning of each of these algorithm component within the optimized parafoil terminal guidance strategy.

# Chapter 4

# Design Analysis of CC-BLG

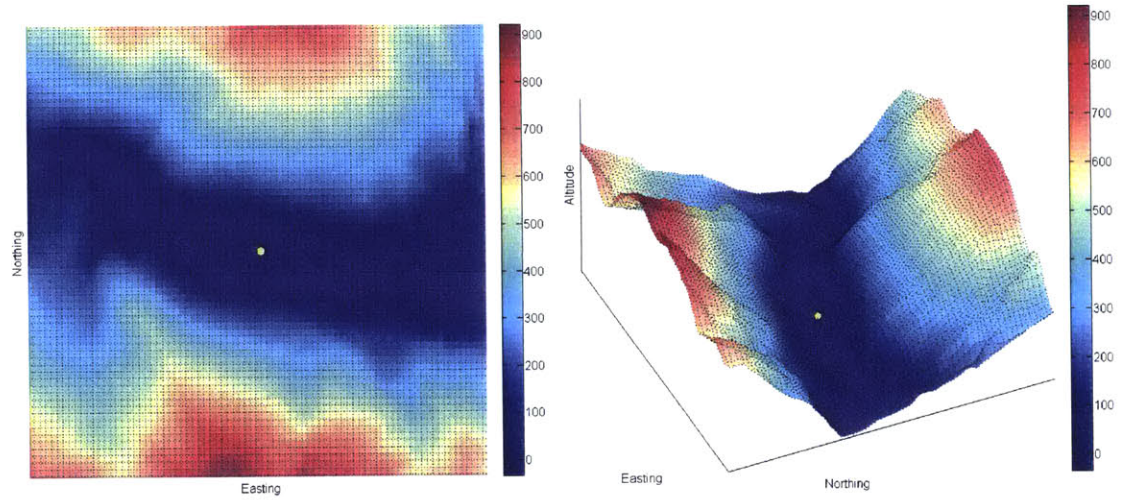
## 4.1 Introduction

This chapter analyzes the performance of each of the three components of the CC-BLG terminal guidance algorithm introduced in Chapter 3. Development and testing of CC-BLG is conducted over a series of Monte Carlo experiments using a high fidelity parafoil simulation environment presented in Section 4.2. First, Section 4.3 considers the effects of increasing the number of covariance samples and ellipses which define the contours of the state uncertainty distribution. Additionally, a comparison is made between the Analytic CC-RRT approach for estimating the probability of constraint violation (Section 2.5), and the weighted analytic uncertainty sampling approach presented in Section 3.3.2. Next, Section 4.4 considers the selection of the propagation horizon $t_P$, and number of control sequences $N_u$, used to define the boundary of the reachability horizon (Section 3.3.4). The CC-BLG objective function parameters of $\alpha_0$ and $\lambda$ (Section 3.3.3) are then tuned in Section 4.5 to create the desired balance between planning robustness and goal seeking behavior. Lastly, this chapter considers the effects of reintroducing hard user-defined bounds on probabilistic feasibility (Section 2.5) in combination with the "soft" penalty-based chance constraints introduced in Section 3.3.3. From the results of this analysis, a final set of guidance parameters are selected.

## 4.2 Implementation

Development and testing of the CC-BLG algorithm are conducted over a series of Monte Carlo experiments using Draper Laboratory's high fidelity simulation environment [48]. This simulator has been implemented in the recent work by Stoeckle [73] as a realistic training platform with which to design a Fault Detection, Isolation, and Recovery (FDIR) algorithm for autonomous parafoil guidance. In this simulation, the parafoil is represented using a full nonlinear dynamics model [73], which incorporates the effects of the parafoil aerodynamics described in [1] and [74]. Feedback is provided for guidance in the form of simulated GPS position and ground velocity measurements, such that an Extended Kalman Filter (EKF) is applied in order to estimate the true wind velocity during descent. The wind estimate provided by the EFK is then used to derive the parafoil heading and airspeed velocity according to [73]. This model will be used throughout the remainder of this thesis as a means to accurately represent the limited state information that is available to most fielded parafoil systems.

In the Monte Carlo experiments conducted in Chapters 4 and 5, the initial conditions of parafoil position, velocity, altitude, and heading are randomly varied during each simulation trial. The parafoil is simulated from the point of release at altitudes uniformly sampled over the range from 3,048–4,572 meters (10,000–15,000 ft), and lateral distances from 0–8,524 meters (0–28,000 ft), while a preselected altitude is used to specify where energy management ends and Phase 1 of terminal guidance begins. In this work, we begin Phase 1 and Phase 2 of BLG/CC-BLG terminal guidance at fixed altitudes of 650 meters and 300 meters, respectively. The parafoil system parameters including payload weight, turn rate bias, and lift-to-drag ratio are also randomized over a range of values suitable for each canopy type [22]. This chapter considers simulations using the *UltraFly* parafoil system (JPADS-ULW) developed by Wamore Inc. [18, 21]. The system weight is uniformly sampled within the range from 250–750 lbs, while the turn rate bias and lift-to-drag ratio are sampled from a Gaussian distribution centered about each nominal value with standard deviations of 0.1



(a) valley top view

(b) valley skew-side view

Figure 4-1: Valley terrain used in CC-BLG algorithm development (meters)

and 0.2, respectively. The presence of wind will be modeled in each simulation from the representative set of 25 wind profiles used for the design of Analytic CC-RRT, as described in Section 2.7.1. This is selected in order to maintain a consistent basis for comparison between each of the three parafoil guidance algorithms considered in this work. The Monte Carlo simulation procedure described above is used in preparation for flight test experiments, described in Chapter 6, as a means to evaluate CC-BLG performance over a range of flight conditions common to real-world drop scenarios.

Complex terrain is modeled through the use of Level 2 Digital Terrain Elevation Data (DTED2) which acts as a 3-Dimensional map of the environment for each set of simulations. During the design analysis of the CC-BLG algorithm presented in this chapter, the 1.5 km x 1.5 km valley terrain environment shown in Figure 4-1 is selected as a test environment for tuning the various algorithm components.

As described in Section 2.7.1, valley terrain represents a particularly challenging scenario for airdrop systems due to tight geometry constraints that can exist at low altitudes and within the vicinity of the target location. This is the case for the valley terrain shown in Figure 4-1, in which elevation is denoted in meters by the transition in color from dark blue (lowest altitude) to dark red (highest altitude). The target for these experiments is placed in the middle of the valley, as indicated

Table 4.1: Baseline CC-BLG parameters for tuning

| **Parameter** | **Value** |
|---|---|
| $\alpha_0$ | 1000 |
| $\lambda$ | 0.003 |
| $t_P$ | 15 $s$ |
| $N_u$ | 9 |

by the yellow circle. In this example, only a narrow band of terrain exists where the parafoil can perform a safe landing at the desired target altitude (colored dark blue), which runs from east to west along the middle of the map. Furthermore, the two mountain ridges on either side of the western portion of the valley create a potentially dangerous terrain bottleneck that continues to limit the feasible planning space throughout the parafoil's descent. The following development of the CC-BLG algorithm analyzes the extent to which wind uncertainty modeling can be used to minimize terrain impacts and generate improved terminal guidance performance over BLG in this difficult terrain environment. This procedure for algorithm development is similar to the approach described in [47] for Analytic CC-RRT, while adjusting to the increased fidelity of the Draper Laboratory simulation model.

Throughout this chapter, data will be presented primarily in the form of cumulative distribution functions (CDF) and tabular data of normalized parafoil miss distance performance. In each experiment, the data is normalized by the median landing accuracy from an identical set of Monte Carlo simulation trials using the nominal BLG algorithm. This enables a comparison of relative performance during the process of evaluating various guidance strategies for the CC-BLG algorithm. Table 4.1 provides the set of baseline CC-BLG parameters which will be used to initiate algorithm tuning in Section 4.3–4.5. As various trade studies are performed, these parameters will be replaced with alternative solutions in order to improve CC-BLG target accuracy and planning robustness.

## 4.3 Analytic Uncertainty Sampling

This section considers the effect on CC-BLG performance of varying the number of covariance samples and ellipses used to approximate the parafoil uncertainty distribution about each nominal trajectory state. In the description of the Analytic CC-RRT algorithm provided in Section 2.4.2, a set of 10 covariance samples were placed along the state distribution at standard deviations of $\sigma_1 = 0.7$ and $\sigma_2 = 1.75$ so as to check for collisions with the terrain environment. This arrangement of covariance samples was found to work well empirically for promoting planner robustness [46] while maintaining computational efficiency, demonstrated previously in Table 2.2. Leveraging the results of prior analysis, these values for $\sigma_1 = 0.7$ and $\sigma_2 = 1.75$ will be adopted as the basis for analytic uncertainty sampling within the CC-BLG algorithm.

From the parameters provided in Table 4.1, a series of Monte Carlo simulation experiments are first conducted within the valley terrain environment in order to examine the impact of increasing the number of covariance samples within each $\sigma$-ellipse. Table 4.2 compares the percentage of terrain collisions which resulted over sets of 1000 Monte Carlo trials of the CC-BLG algorithm, in which the number of covariance samples per ellipse is varied from 5 to 250 samples for both $\sigma_1$ and $\sigma_2$. As the number of samples is increased, the results in Table 4.2 demonstrate a gradual decrease in the CC-BLG crash percentage for the valley terrain scenario. However, for coarse levels of the discretization, the CC-BLG worst-case performance also appears sensitive to the number and arrangement of covariance samples. This is suggested by the way the crash percentage fluctuates over the range from 5 to 25 samples per

Table 4.2: Crash percentage vs. samples per ellipse, 1000 trials

| **Samples per $\sigma$-ellipse** | | **% Crash** |
|---|---|---|
| $\sigma_1$: 5, | $\sigma_2$: 5 | 5.1 |
| $\sigma_1$: 10, | $\sigma_2$: 10 | 4.3 |
| $\sigma_1$: 15, | $\sigma_2$: 15 | 4.7 |
| $\sigma_1$: 20, | $\sigma_2$: 20 | 4.1 |
| $\sigma_1$: 25, | $\sigma_2$: 25 | 4.7 |
| $\sigma_1$: 50, | $\sigma_2$: 50 | 3.5 |
| $\sigma_1$: 250, | $\sigma_2$: 250 | 3.5 |

Table 4.3: Average computation time per state for CC-BLG

| **Samples, $N_S$** | **Time per state ($\mu s$)** |
|---|---|
| 10 | 2.098 |
| 20 | 2.974 |
| 30 | 4.080 |
| 40 | 4.885 |
| 50 | 5.799 |
| 100 | 9.985 |
| 500 | 43.33 |

ellipse. When the number of samples per ellipse is further increased to both 50 and 250 samples, the percentage of terrain collisions stabilizes over the final two sets of 1000 Monte Carlo trials. Further increasing the number of covariance samples for this selection of $\sigma_1$ and $\sigma_2$ is not expected to yield additional improvements.

These results indicate that refining the discretization of each $\sigma$-ellipse may have a limited impact on CC-BLG robustness when the total number of covariance ellipses is small. This can be explained by the way each $\sigma$-ellipse represents only a single contour of the continuous bivariate Gaussian distribution of $(\delta p_x, \delta p_y)$. Nevertheless, Table 4.2 reveals that the collision percentage of the CC-BLG algorithm is small, and remains less than 5% for all but the coarsest value of 5 samples per ellipse. In particular, we observe that the same arrangement of 10 $\sigma_1$ and $\sigma_2$ covariance samples used by Analytic CC-RRT results in a CC-BLG collision percentage that is well below the probability threshold $(1 - p_{safe}) = 0.1$ specified previously in Section 2.7.

In prior work by Sugel [47], the effect of varying the number of standard deviations $\sigma$ within the outermost covariance ellipse was evaluated for the Analytic CC-RRT algorithm in order to determine the scale factor which empirically satisfied the user-defined probability constraint. As the value of $\sigma$ was increased, this analysis indicated a favorable reduction in the number of Analytic CC-RRT collisions due to the planner's improved awareness of terrain proximity. Recall that in Section 2.4.2 and Section 3.3.2, the parafoil proximity to terrain can only be estimated from the location of the nearest collided covariance samples. As a result, this work considers the impact on CC-BLG performance of adding an additional covariance ellipse positioned beyond the standard deviations of both $\sigma_1$ and $\sigma_2$.

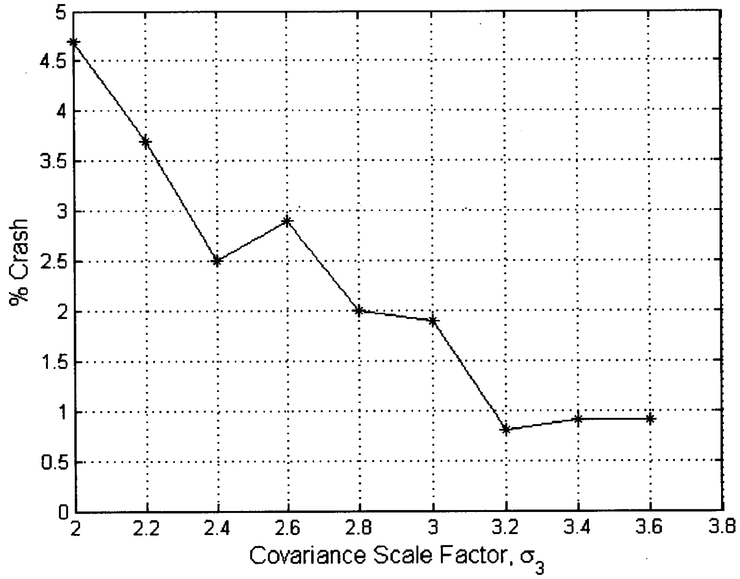


Figure 4-2: Crash percentage at various $\sigma_3$ standard deviation levels

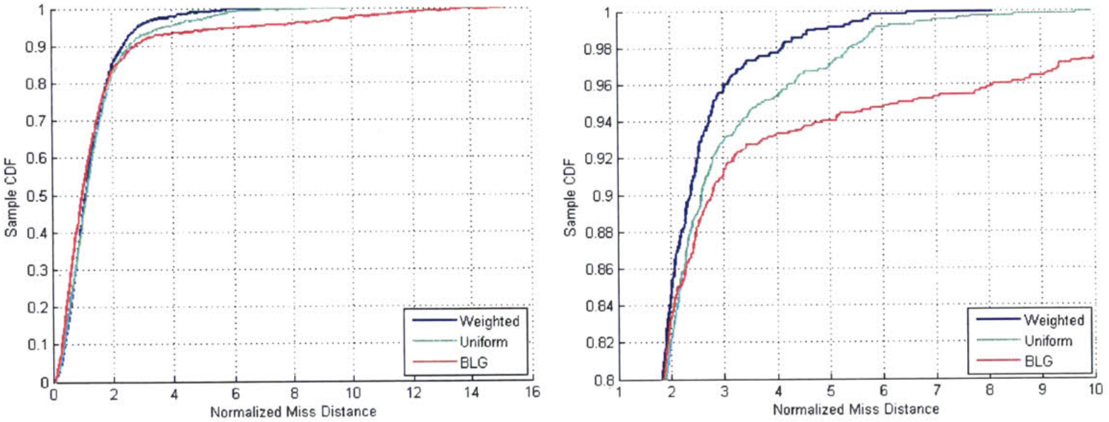
Table 4.3 provides the average computation time per state over each set of 1000 Monte Carlo simulations conducted previously in Table 4.2. These experiments were performed on a 3.20 GHz Intel i5-3470 desktop computer with 4 GB of RAM. The results shown in Table 4.3 indicate that the average computation time per state is quite small for the CC-BLG algorithm (i.e., microseconds) and scales favorably with the increase in the number of covariance samples. For most modern CPUs, the limited computation required to generate each covariance sample and check for intersection with the terrain can therefore be leveraged to provide adequate sampling definition over each of the $\sigma$-ellipses used within CC-BLG. In the following analysis, we restrict our attention to a maximum number of 40 covariance samples in order to study the effect of doubling the value of $N_S$ used previously by Analytic CC-RRT. The simulation experiments conducted in Figure 4-2 therefore consider introducing a third ring of 20 covariance samples with standard deviation $\sigma_3$, in addition to the two rings of 10 covariance samples provided above.

Figure 4-2 demonstrates the change in the percentage of CC-BLG collisions within the valley terrain environment as the value of $\sigma_3$ is varied along the state uncertainty distribution. In this experiment, an identical set of 1000 Monte Carlo simulation trials are performed for each value of $\sigma_3$, while all other parameters remain equal to those used previously in Table 4.2. The results shown in Figure 4-2 demonstrate a reduction in the number of terrain collisions as the value of $\sigma_3$ is increased from 2.0 to 3.6. This suggests that CC-BLG robustness to wind uncertainty improves in complex terrain environments as the planner is able to sample the state distribution at greater distances from the nominal trajectory plan. Likewise, the results in Figure 4-2 indicate that the addition of a third ring of covariance samples can significantly enhance CC-BLG performance while using only twice as many samples as the previous Analytic CC-RRT approach. As the value of $\sigma_3$ increases to 3.2, the CC-BLG crash percentage is reduced to less than 1%, while the effect of further increasing the value of $\sigma_3$ appears to subside. Because the probability of the parafoil being outside the $\sigma_3$ ellipse (3.4) has already been reduced to less than 0.6% for $\sigma_3 = 3.2$, this helps to explain the diminishing returns provided by increasing the standard deviation of the outermost covariance ellipse. From this analysis, the values of $\sigma_3 = 3.2$ and $N_S = 40$ are selected for use throughout the remainder of this thesis.

### 4.3.1 Weighted Sampling Analysis

With the addition of a third covariance ellipse, a final set of Monte Carlo simulation experiments are conducted in this section in order to evaluate the performance of the CC-BLG approach to weighted analytic uncertainty sampling introduced in Section 3.3.2. This approach considers approximating the probability of constraint violation at each time step using (3.8). The weighted sampling method is compared with the previous probability estimate used by Analytic CC-RRT in which all covariance samples are assigned uniform weights of $P(S) = 1/N_S$. As described in Section 2.5, the probability of constraint violation is approximated within Analytic CC-RRT as the fraction of total covariance samples which collide with the terrain environment.

Figure 4-3 and Table 4.4 present the results from 1000 Monte Carlo simulations of the CC-BLG algorithm in the valley terrain environment using both the "Weighted" and "Uniform" sampling strategies described above. For comparison, the results from an equivalent set of 1000 Monte Carlo simulations are also provided using the nominal



(a) full CDF

(b) zoomed in to $(1, 10) \times (0.8, 1)$

Figure 4-3: Normalized miss distance CDF for analytic sampling method, 1000 trials

Table 4.4: Normalized miss distance data for analytic sampling method, 1000 trials

| Sampling Method | Mean | StDev | 50% | 80% | 90% | 95% | 98% | Max |
|---|---|---|---|---|---|---|---|---|
| Weighted | 1.305 | 0.911 | 1.126 | 1.841 | 2.391 | 2.822 | 4.125 | 8.078 |
| Uniform | 1.428 | 1.193 | 1.147 | 1.914 | 2.595 | 3.713 | 5.376 | 9.952 |
| BLG | 1.625 | 2.225 | 1.000 | 1.861 | 2.788 | 6.298 | 10.713 | 15.133 |

BLG algorithm. The median miss distance of the BLG algorithm has been used to normalize each set of data in accordance with the approach described previously in Section 4.2.

From the CDF plots in Figure 4-3(a)–(b), the method of weighted analytic uncertainty sampling for CC-BLG is shown to provide superior miss distance performance over both the nominal BLG algorithm, and the CC-BLG algorithm using uniform sample weights. As indicated in Figure 4-3(a), the distribution of parafoil miss distances in almost identical for all three data sets up to the 80th percentile. For the remaining 200 Monte Carlo trials in the upper 20% of the distribution, the combined effect of complex terrain and shifting wind conditions has a noticeable impact on planning performance by increasing the risk of future terrain collisions. Over this set of trials, the weighted analytic sampling approach demonstrates significant improvements in CC-BLG performance by modeling the variation in the distribution of $(\delta p_x, \delta p_y)$ described in Section 3.3.2. In this way, the method of weighted sample probabilities allows for a higher fidelity estimate of environmental risk than the previous

"Uniform" approach. This improvement is evident in both the "Weighted" CC-BLG landing accuracy above the 80th percentile, as well as in the shorter distribution tail produced by decreasing the number of worst-case outliers apparent in both the nominal BLG algorithm, and "Uniform" CC-BLG approach. A detailed comparison of BLG and CC-BLG performance in complex terrain is reserved for Chapter 5.

Given the results provided in Table 4.4 and Figure 4-3, the weighted analytic uncertainty sampling approach is selected for use within the CC-BLG algorithm. The following sections consider the effect of modifying the parameters provided in Table 4.1 to examine the possibility for further CC-BLG performance improvements within the valley terrain environment.

## 4.4 Reachability Horizon and Cost-to-go

This section reexamines the choice of time horizon $t_P$ in (3.14), and number of control sequences $N_u$ in (3.15), which define the boundary of the CC-BLG reachability set approximation. The process of determining an appropriate upper bound on the propagation horizon will be considered first, and is based on an evaluation of the parafoil maximum rate of turn.

This thesis considers the presence of complex terrain features found in real-world drop environments such as valleys, canyons, and mountain ranges. Arbitrary, non-convex terrain environments of this type can make predicting vehicle safety over future planning horizons particularly difficult. Consider as an example the case of a "dead end" terrain corridor shown in Figure 4-4, which is positioned ahead of the parafoil prior to the transition to Phase 2 terminal guidance. Depending on the corridor width, and given the parafoil's current heading, this type of scenario may make it challenging, if not impossible, for the parafoil to turn around without producing a collision with the terrain environment. Likewise, if the parafoil is flying directly toward the nearest terrain obstacle (e.g., Figure 3-5), there may be insufficient time at the start of Phase 2 guidance to perform a successful turn maneuver due to the underactuated nature of the parafoil dynamics. To maintain parafoil safety, the value

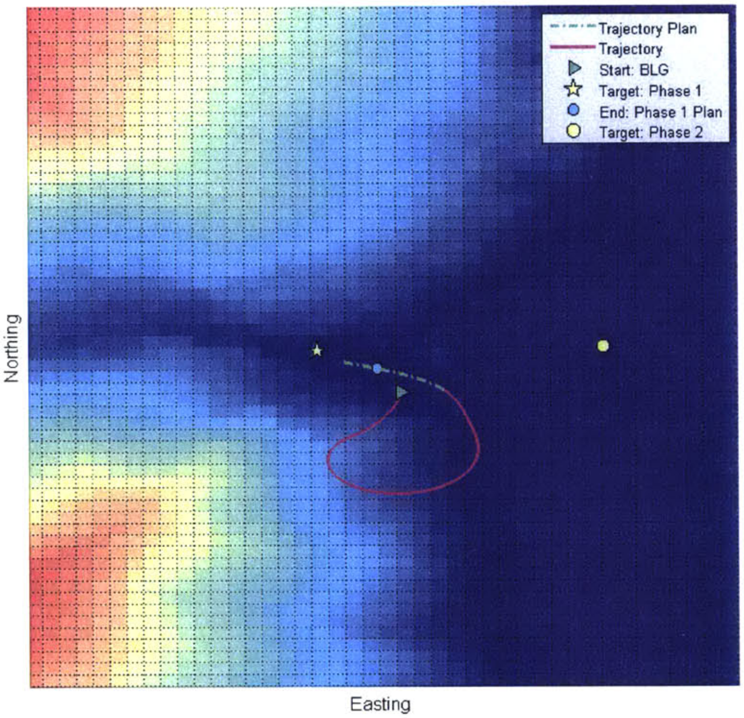


Figure 4-4: Dead end terrain scenario prior to BLG terminal guidance phase transition

of the propagation horizon $t_P$ must be sufficiently large such that dangerous terrain features will be detected by the reachability states, and avoided during the transition between terminal guidance phases.

This section considers the choice of four potential propagation horizons for the reachability set, which are selected based on the maximum change in heading $|\Delta\psi_{P_{\max}}|$ that can be achieved by the parafoil at the limits of the control $\omega_{\max}$. These turn maneuvers are selected as $|\Delta\psi_{P_{\max}}| = \pi/2$, $3\pi/4$, $\pi$, and $5\pi/4$ radians, respectively. The span of the parafoil reachability set is shown in Figure 4-5 for $\overline{\mathbf{w}} = 0$, as the upper bound on the propagation time $t_P$ is increased for each maneuver.

At the limits of the parafoil's control authority, $\omega_{\max}$, the time period to complete a half turn for $|\Delta\psi_{P_{\max}}| = \pi$ represents an intelligent choice for the propagation horizon given the shape of the reachability boundary shown in Figure 4-5. In the example of a dead end terrain corridor described above, a reachability horizon of $|\Delta\psi_{P_{\max}}| \geq \pi$ would discourage the parafoil from entering any region in which there is insufficient clearance to perform a full change in course direction of at least $\pi$ radians. Similarly,

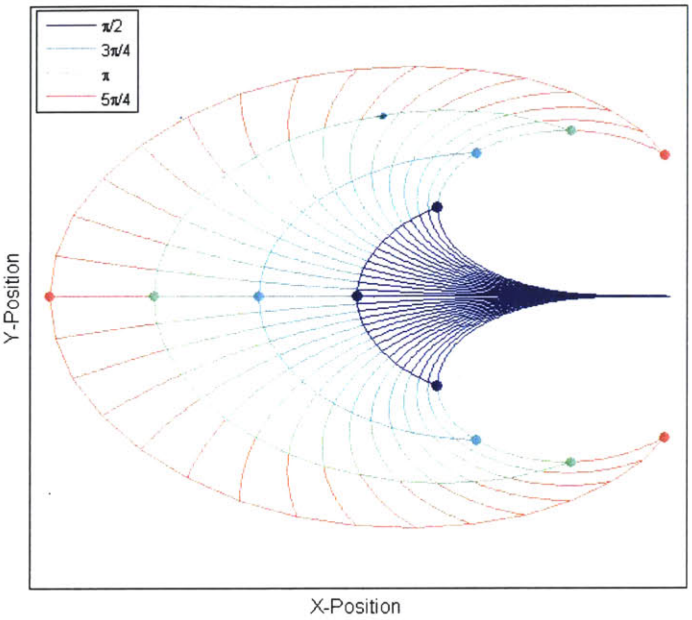
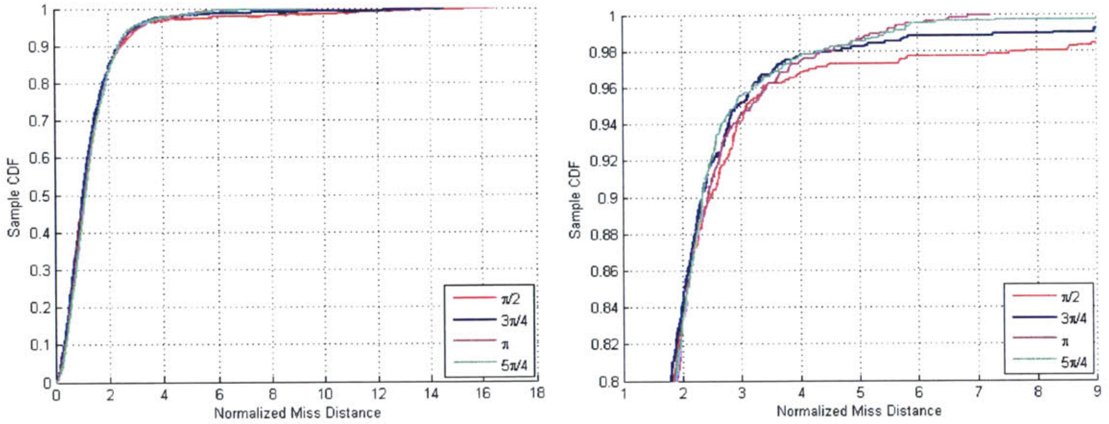


Figure 4-5: Parafoil reachability horizon as propagation time $t_P$ is increased, $\overline{\mathbf{w}}$=0

if the parafoil is far enough away from the terrain such that a turn of $|\Delta\psi_{P_{\max}}| = \pi$ can be completed safely in either direction, it is reasonable to assume that the terminal state of the Phase 1 trajectory may have at least one feasible Phase 2 solution.

Table 4.5 and Figure 4-6 present simulation results over 1000 Monte Carlo trials of the CC-BLG algorithm on the valley terrain environment using each of the maximum turn maneuvers provided above. Here it is assumed that propagation of the parafoil dynamics in (2.32)–(2.36) begins from an initial state of straight flight with symmetric line deflections such that the time to turn left and right remains equivalent for each pair of $N_u$ control options, as shown in Figure 4-5. In these experiments, a value of $N_u = 9$ is selected so as to ensure sufficient coverage for obstacle detection across the boundary of the reachability set. The values of $\alpha_0 = 1000$ and $\lambda = 0.003$ were also used according to Table 4.1.

From the normalized CDF of parafoil miss distance shown in Figure 4-6, the effect of changing the propagation horizon $t_P$ is most noticeable above the 90th percentile of the distribution. In this region, the shape of the reachability boundary has an impact on parafoil performance by determining the extent to which the CC-BLG



(a) full CDF

(b) zoomed in to $(1, 9) \times (0.8, 1)$

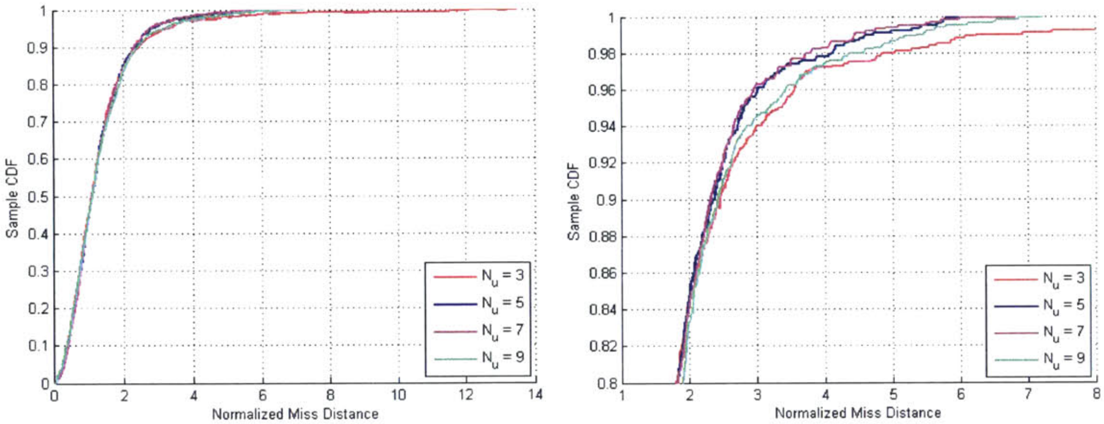
Figure 4-6: Normalized miss distance CDF for maximum turn $|\Delta\psi_{P_{\max}}|$, 1000 trials

Table 4.5: Normalized miss distance data for maximum turn $|\Delta\psi_{P_{\max}}|$, 1000 trials

| **Reachability Horizon** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| $\lvert\Delta\psi_{P_{\max}}\rvert = \pi/2$ | 1.439 | 1.745 | 1.053 | 1.851 | 2.471 | 3.095 | 7.814 | 16.480 |
| $\lvert\Delta\psi_{P_{\max}}\rvert = 3\pi/4$ | 1.334 | 1.407 | 1.042 | 1.800 | 2.339 | 2.926 | 4.521 | 14.552 |
| $\lvert\Delta\psi_{P_{\max}}\rvert = \pi$ | 1.331 | 0.991 | 1.129 | 1.897 | 2.428 | 3.184 | 4.352 | 7.220 |
| $\lvert\Delta\psi_{P_{\max}}\rvert = 5\pi/4$ | 1.349 | 1.020 | 1.146 | 1.883 | 2.349 | 2.911 | 4.277 | 9.633 |

algorithm can successfully avoid terrain collisions during phase transition. The results in Table 4.5 and Figure 4-6 reveal that the length of the propagation horizon for $|\Delta\psi_{P_{\max}}| = \pi/2$ is insufficient to provide robust detection of environmental obstacles. Likewise, the selection of $|\Delta\psi_{P_{\max}}| = 3\pi/4$ results in poor CC-BLG performance above the 98th percentile and produces a maximum miss distance that is over 14 times the median BLG recorded accuracy on the valley terrain. This maximum miss distance was produced by an unanticipated terrain collision during phase transition, and thus is considered unacceptable for improving BLG performance. As a result, the value of $|\Delta\psi_{P_{\max}}| = \pi$ will be selected in this work in order to generate the largest improvement in both the mean and worst-case performance of the CC-BLG algorithm on the valley terrain. Given the parafoil dynamics model provided in (2.32)–(2.36), this maximum turn maneuver translates into a propagation horizon of $t_P = 17.6$ seconds.

With the selection of the propagation horizon $t_P$, an adequate number of control sequences $N_u$ must be determined in order to allow for robust obstacle detection along the reachability frontier. Using constant control sequences, the limits on the



(a) full CDF

(b) zoomed in to $(1, 9) \times (0.8, 1)$

Figure 4-7: Normalized miss distance CDF for reachability states $N_u$, 1000 trials

Table 4.6: Normalized miss distance data for reachability states $N_u$, 1000 trials

| **Reachability States** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| $N_u = 3$ | 1.377 | 1.322 | 1.111 | 1.811 | 2.465 | 3.317 | 4.851 | 13.490 |
| $N_u = 5$ | 1.298 | 0.886 | 1.126 | 1.838 | 2.370 | 2.816 | 4.092 | 6.480 |
| $N_u = 7$ | 1.291 | 0.862 | 1.115 | 1.831 | 2.317 | 2.771 | 3.735 | 6.817 |
| $N_u = 9$ | 1.331 | 0.991 | 1.129 | 1.897 | 2.428 | 3.184 | 4.352 | 7.220 |

control $\{-\omega_{\max}, \omega_{\max}\}$ produce a finite window in which to discretize the reachability boundary. This also serves to decrease the marginal information benefit of each additional reachability state as the value of $N_u$ becomes large [47]. Thus we seek to determine the minimum number of dynamic state propagations required for obstacle detection, so as to reduce unnecessary computation during each Phase 1 planning iteration. Following the procedure outlined in Ref. [47], only odd values of $N_u$ are considered in this work such that the $\omega = 0$ control maneuver for straight flight is always included within the reachability set.

The results of an additional set of Monte Carlo simulations within the valley terrain are shown in Table 4.6 and Figure 4-7 as the value of $N_u$ is increased from 3 to 9. It can be concluded from Figure 4-7(b) that when $N_u = 3$, the separation between reachability states at $t_P = 17.6$ seconds can become too large for effective obstacle detection. This produces a decline in CC-BLG performance above the 90th percentile of the CDF distribution. In contrast, the values of $N_u = 5$ and $N_u = 7$ successfully improve the maximum miss distance achieved by CC-BLG during each set of Monte

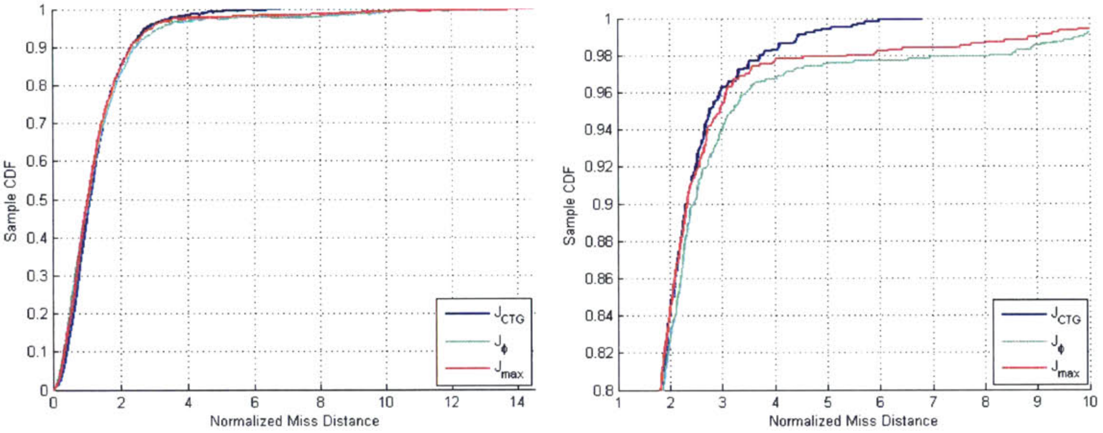
Carlo simulations, and result in nearly identical performance on the valley terrain—with $N_u = 7$ showing a marginal advantage above the 95th percentile for all but the worst-case trial. Raising the number of control sequences to $N_u = 9$ generates slightly worse performance in Table 4.6, as the frequency of terrain detection is increased due to the additional number of reachability states and covariance samples. While this high rate of detection from collided covariance samples continues to maintain parafoil safety, the results of this experiment suggest that further increasing the value of $N_u$ can discourage goal seeking behavior during Phase 1 terminal guidance, and may cause CC-BLG to position the parafoil further away from the desired upwind target. This can potentially limit final target attainability during Phase 2 guidance, resulting in larger parafoil miss distances. If greater definition of the reachability set is required, this conservatism can be offset by decreasing the weight on $\xi$ within the cost-to-go function (3.15); however this is not considered further in this work. A value of $N_u =$ 7 is therefore selected which demonstrates the best overall performance in Figure 4-7 and Table 4.6, while limiting the maximum number of reachability states required.

### 4.4.1 Cost-to-go Function

In Section 3.3.4, the risk-based cost-to-go function $J_{CTG}$ was introduced into the process of CC-BLG trajectory optimization to enforce obstacle avoidance and future planning feasibility during the transition between terminal guidance phases. This section compares the performance of the $J_{CTG}$ approach in (3.15) with two alternative formulations of the cost-to-go function, $J_\phi$ and $J_{\max}$,

$$\begin{aligned} J_\phi &= \max\left(J_{T,F}, \min(J_{R,1}, J_{R,2}, \ldots, J_{R,N_u})\right), & (4.1) \\ J_{\max} &= \max(J_{R,1}, J_{R,2}, \ldots, J_{R,N_u}). & (4.2) \end{aligned}$$

The $J_\phi$ variant is based upon the design of the cost-to-go function $\phi_F$ in (2.60) which was used in the development of the Analytic CC-RRT algorithm. This approach takes the maximum between the cost at the terminal state of the Phase 1 trajectory $J_{T,F} = J_{T,N_T}$, and the minimum cost $J_{R,n}$ for $n \in \{1, \ldots, N_u\}$ over all state trajectories



(a) full CDF

(b) zoomed in to $(1, 10) \times (0.8, 1)$

Figure 4-8: Normalized miss distance CDF for reachability cost-to-go, 1000 trials

Table 4.7: Normalized miss distance data for reachability cost-to-go, 1000 trials

| **Cost-to-go Function** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| $J_{CTG}$ | 1.291 | 0.862 | 1.115 | 1.831 | 2.317 | 2.771 | 3.735 | 6.817 |
| $J_\phi$ | 1.407 | 1.559 | 1.073 | 1.870 | 2.468 | 3.177 | 8.097 | 14.274 |
| $J_{\max}$ | 1.347 | 1.403 | 1.063 | 1.819 | 2.344 | 2.974 | 5.378 | 15.189 |

along the reachability horizon. If at least one reachability set control sequence incurs zero risk over the finite time horizon, the solution provided by $J_\phi$ is considered cost optimal. As a result, the $J_\phi$ cost-to-go function assumes the planner's ability to design and execute a trajectory at the start of Phase 2 guidance which approximates the path of the lowest cost control solution. However, by using only the minimum value of $J_{R,n}$, this approach does not penalize the potential levels of risk which may exist over the remaining reachability states.

In contrast, the cost-to-go variant $J_{\max}$ is designed to penalize the maximum cost $J_{R,n}$ over all trajectories of the reachability horizon, regardless of whether a control sequence exists which avoids terrain proximity (i.e., $J_{R,n} = 0$). The $J_{\max}$ approach encourages the entire set of reachability states to be risk free, in a similar manner to the $J_{CTG}$ formulation, but does not consider penalizing the cumulative risk which may exist across multiple reachability set trajectories simultaneously.

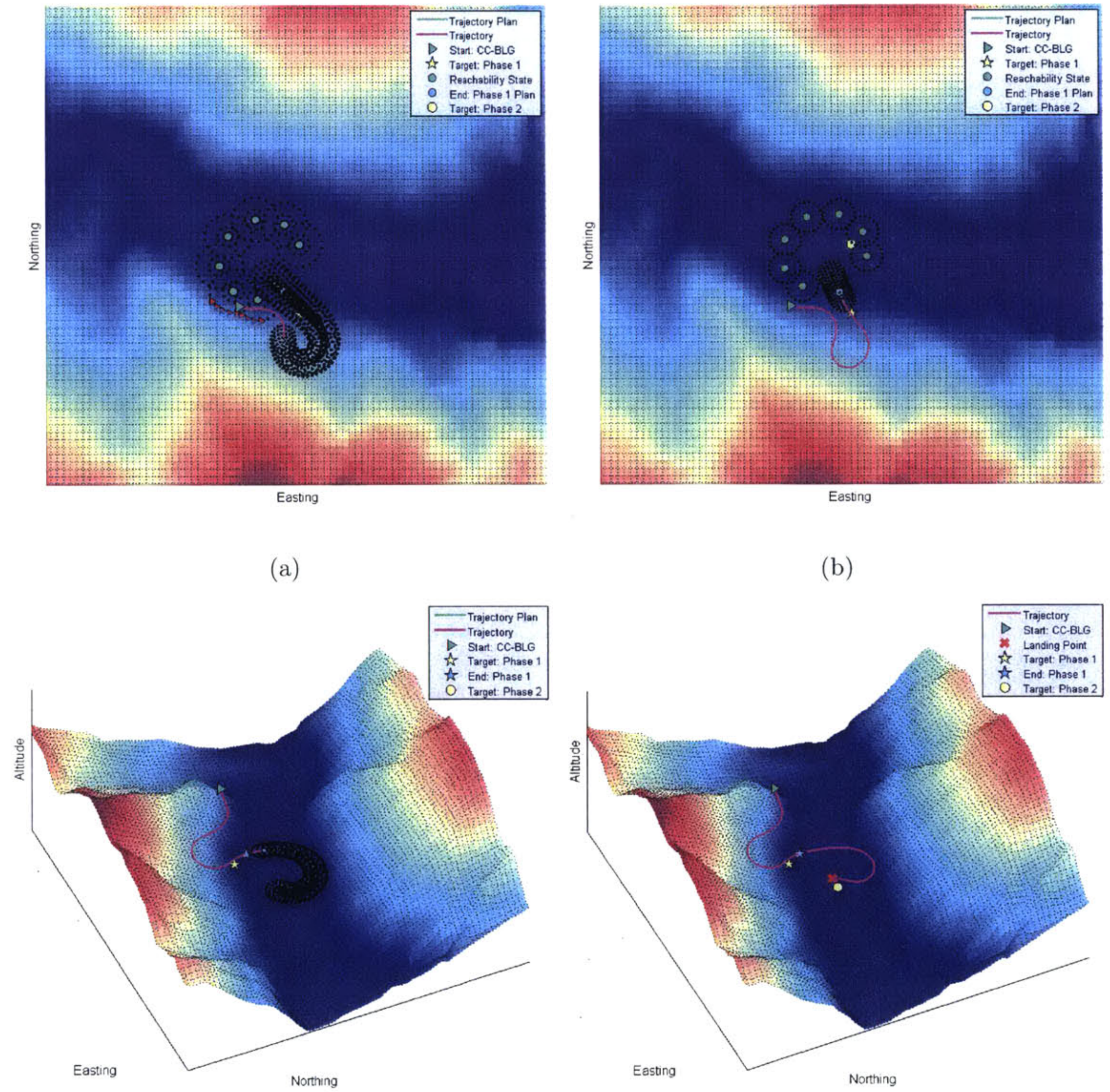
Figure 4-8 and Table 4.7 provide the normalized miss distance distributions from 1000 Monte Carlo trials of CC-BLG using each of the reachability set cost-to-go functions. The results in Figure 4-8 demonstrate that the CDFs produced by all three strategies remain largely the same below the 80th percentile. However, for the 10% of trials above the 90th percentile of the distribution, the $J_\phi$ cost-to-go function generates larger miss distances than either the $J_{\max}$ or $J_{CTG}$ approaches. This reveals that penalizing only the minimum cost trajectory over the reachability horizon may not be sufficient to represent the true level of environmental risk. Additionally, the CDFs for the $J_{\max}$ and $J_\phi$ cost-to-go functions contain large distribution tails above the 98th percentile due to the increased number of terrain impacts caused by both strategies during terminal guidance phase transition.

On the other hand, the $J_{CTG}$ function provides improved miss distance performance compared to $J_\phi$ and $J_{\max}$ above the 90th percentile, while helping to remove the large distribution tail. Recall that in (3.15), the formulation for $J_{CTG}$ penalizes the sum of risk penalties over all reachability states. This is selected in order to produce the largest feasible planning space ahead of the parafoil during the terminal guidance phase transition. From the simulation results in Figure 4-8 and Table 4.7, we conclude that the $J_{CTG}$ approach to obstacle avoidance yields superior robustness over both $J_\phi$ and $J_{\max}$, which ignore the combined level of risk across multiple reachability states.

### 4.4.2 Risk Mitigation

Figure 4-9 demonstrates how the reachability set approximation and augmented cost-to-go function can be used to reorient the parafoil approach to the Phase 1 target location. In this simulation example, the same initial conditions and Phase 1 target have been used as in Figure 3-5 for the nominal BLG algorithm. Figure 4-9(a)-(b) display the states of the reachability approximation at $t = t_F + \tau$, when the upper bound on the propagation time $t_P = 17.6$ seconds, and the number of control sequences $N_u = 7$. Covariance samples are denoted by the black dots in Figure 4-9, which are placed at three distinct scale factors for $\sigma = 0.7$, 1.75, and 3.2 respectively.

Figure 4-9(a) displays the reachability set approximation during Phase 1 terminal guidance in which the detection of terrain obstacles is shown by the red collided



(a)

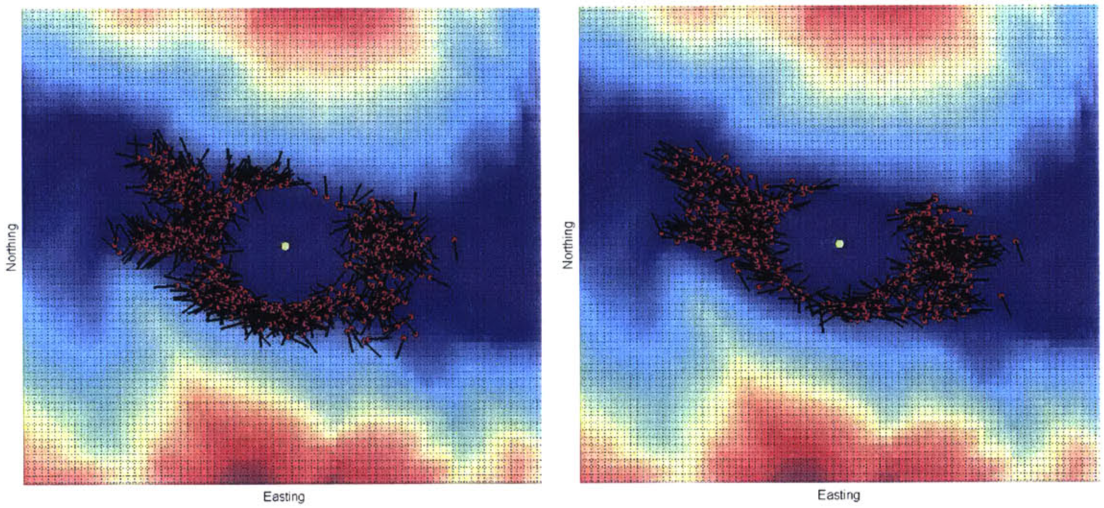
(b)

(c)

(d)

Figure 4-9: Risk mitigation during guidance phase transition using reachability set

covariance samples. With the addition of the $J_{CTG}$ component in (3.16), Figure 4-9(a) and Figure 4-9(b) illustrate that the parafoil approach direction to the Phase 1 target now faces toward the open interior of the valley, and away from the nearest terrain obstacles. Despite this change in heading, Figure 4-9(b) reveals that the parafoil is still capable of approaching the desired upwind target (yellow star) in preparation for Phase 2 terminal guidance. As compared to Figure 3-5 however, the risk of collision



(a) BLG (b) CC-BLG

Figure 4-10: Phase 2 initial conditions ($d \geq 250$ $m$) produced by BLG and CC-BLG

has been significantly reduced. Figure 4-9(c) shows the CC-BLG trajectory plan after two iterations of Phase 2 terminal guidance, indicating a path toward the final target location has been successfully identified. The final trajectory executed by the parafoil is shown in Figure 4-9(d), demonstrating that the target altitude has been reached, and all terrain obstacles have been avoided.

Figure 4-10(a)-(b) compare the initial conditions for Phase 2 terminal guidance produced by the BLG and CC-BLG algorithms over 1500 Monte Carlo simulation trials in the valley terrain environment. In these figures, each parafoil position is marked by a red colored circle, while the direction of the parafoil ground velocity is indicated using a black arrow. To aid in the visualization, Figure 4-10 displays only those initial conditions in which the parafoil distance from the target $d = \sqrt{\Delta p_x^2 + \Delta p_y^2} \geq 250$ meters, representing an increased risk of terrain proximity at the start of Phase 2 terminal guidance.

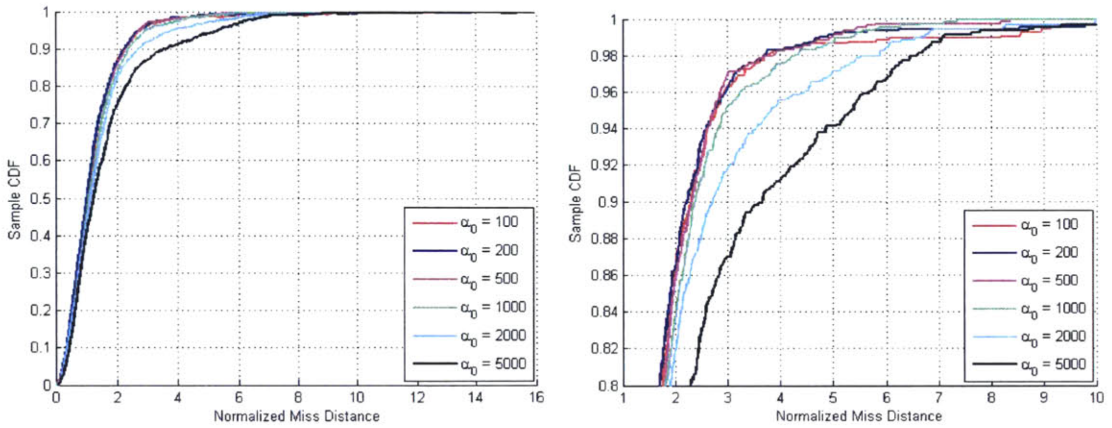
The results for the nominal BLG algorithm are shown in Figure 4-10(a). In this set of Monte Carlo simulations, 867 initial conditions for Phase 2 terminal guidance began from a distance of over 250 meters from the target location. From Figure 4-10(a) it can be observed that the BLG algorithm produces numerous initial conditions for which the parafoil is both in close proximity to the terrain, and oriented in the direction of the nearest terrain obstacle (i.e., the blue-green terrain elevation band). From the total set of 1500 simulations, 56 terrain collisions occurred at high altitude during the BLG transition to Phase 2 terminal guidance. As described in Section 3.3.4, this is a consequence of finite horizon trajectory planning that does not adequately consider future feasibility requirements.

Conversely, Figure 4-10(b) displays 662 initial conditions produced by the CC-BLG algorithm at the start of Phase 2 terminal guidance. Here it can be observed that the CC-BLG reachability approximation has reduced the number of initial conditions within close proximity to terrain. The vast majority of initial conditions have also been reoriented to face either toward the middle of the map, or along the axis of the east-west valley. As a result, the parafoil's approach direction to the Phase 1 target tends to face away from the nearest terrain obstacles, so as to maximize the feasible planning space during the initial iterations of Phase 2 trajectory planning. In this set of 1500 CC-BLG Monte Carlo simulations, not a single high altitude collision occurred during phase transition. The results shown in Figure 4-9 and Figure 4-10 highlight the improvements to vehicle safety produced by the reachability set approximation for finite horizon path planning in the presence of environmental obstacles.

## 4.5 Multi-Objective Optimization Parameters

The CC-BLG multi-objective function (3.16) introduced in Section 3.3.4 combines the nominal BLG cost objective $J_{BLG}$ provided in (1.3), with the two risk-based cost components $J_{CC}$ and $J_{CTG}$ presented in (3.11) and (3.15), respectively. This section considers the appropriate tuning of the augmented cost parameters $\alpha_0$ and $\lambda$ in (3.11) to produce the desired balance of performance between risk minimization, and the objectives for both landing accuracy and landing speed.

To select an appropriate value for the risk penalty parameter $\alpha_0$, a series of Monte Carlo simulation experiments are performed within the valley terrain environment. In these experiments, 1000 simulation trials are conducted for each value of $\alpha_0 \in$



(a) full CDF

(b) zoomed in to $(1, 10) \times (0.8, 1)$

Figure 4-11: Normalized miss distance CDF for $\alpha_0$, 1000 trials

Table 4.8: Normalized miss distance data for $\alpha_0$, 1000 trials

| **Weight, $J_{CC}$** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| $\alpha_0 = 100$ | 1.297 | 1.198 | 1.089 | 1.767 | 2.298 | 2.782 | 3.782 | 13.649 |
| $\alpha_0 = 200$ | 1.235 | 1.131 | 1.024 | 1.707 | 2.216 | 2.735 | 3.688 | 15.103 |
| $\alpha_0 = 500$ | 1.283 | 0.962 | 1.107 | 1.809 | 2.330 | 2.746 | 3.868 | 12.737 |
| $\alpha_0 = 1000$ | 1.329 | 1.033 | 1.090 | 1.809 | 2.369 | 2.972 | 4.270 | 14.258 |
| $\alpha_0 = 2000$ | 1.452 | 1.290 | 1.141 | 1.906 | 2.717 | 3.821 | 5.875 | 11.260 |
| $\alpha_0 = 5000$ | 1.709 | 1.637 | 1.238 | 2.288 | 3.655 | 5.340 | 6.559 | 15.939 |
| BLG | 1.625 | 2.225 | 1.000 | 1.861 | 2.788 | 6.298 | 10.713 | 15.133 |

$\{100, 200, 500, 1000, 2000, 5000\}$ using a value of $\lambda = 0$, such that all trajectory states are weighted equally during the (3.11) cost minimization. These results are provided in Figure 4-11 and Table 4.8. The results produced from an equivalent set of 1000 Monte Carlo simulation trials using the nominal BLG algorithm, shown previously in Table 4.4, are repeated for convenience in Table 4.8.

Figure 4-11(a) reveals that the distribution of CC-BLG miss distances remains relatively constant below the 60th percentile as the value of $\alpha_0$ is increased from 100 to 5000. In this section of the distribution, the increasing penalty on risk has only a minor effect on the planner's ability to identify a feasible trajectory solution to the target that successfully minimizes terrain interaction. However, as the value of $\alpha_0$ becomes large, the CC-BLG level of conservatism is also increased. Under such conditions, the planner may choose to design a trajectory to an alternative landing point further away from the target so as to better reduce the risk to parafoil safety.

This effect is most noticeable in Figure 4-11(b), which highlights the region of the distribution above the 80th percentile. For $\alpha_0 = 2000$ and $\alpha_0 = 5000$, the increase in planning conservatism helps to improve the number of safe landings at the target altitude, but also introduces larger miss distances which degrade CC-BLG accuracy in the upper 20% of the distribution.

Despite the effect on planning conservatism, the results in Table 4.8 demonstrate an improvement in CC-BLG performance over BLG above the 95th percentile of the distribution for all of the values of $\alpha_0$ considered. At smaller values such as $\alpha_0 = 100$, 200, 500, and 1000, this improvement extends to all Monte Carlo trials above the 80th percentile, and also results in better CC-BLG mean landing performance. This effect is caused by the introduction of the $J_{CC}$ component within the objective function to reduce the number of off-nominal terrain collisions, and will be analyzed further in Chapter 5. For this work, a value of $\alpha_0 = 500$ is selected which provides an appropriate balance between conservatism and landing accuracy, as shown in Table 4.8, while reducing some of the worst-case outliers that were not eliminated at smaller values.

The selection of the decay rate factor $\lambda$ is conducted empirically using a similar process of Monte Carlo simulation experiments. Recall that in (3.11), this factor serves to prioritize the minimization of short horizon risk over long horizon risk by de-weighting the penalty $\alpha$ with increasing altitude separation $|p_{z,0} - p_{z,k}|$ from the current parafoil state $(p_{x,0}, p_{y,0}, p_{z,0})$. The values of $\lambda$ considered in this section are derived by selecting a desired minimum for the ratio $\alpha^* = \alpha(p_{z,G})/\alpha(p_{z,0})$, where the function $\alpha(b)$ represents (3.12) evaluated at the altitude $p_{z,k} = b$. The value of $\alpha^*$ specifies the lower bound on the ratio of $\alpha$-weights between the target altitude $\alpha(p_{z,G})$ and current altitude $\alpha(p_{z,0}) = \alpha_0$ based on the total altitude change $\Delta p_z{}^*$ within each terminal guidance phase. For a given ratio of $\alpha^*$ and total altitude change $\Delta p_z{}^*$, the corresponding value of $\lambda$ is found by rearranging (3.12) as

$$\lambda = \frac{-\ln(\alpha^*)}{\Delta p_z{}^*}. \tag{4.3}$$

To determine the effects of the $\lambda$ parameter, five values of $\alpha^* \in \{0.05, 0.1, 0.2, 0.4, 0.8\}$

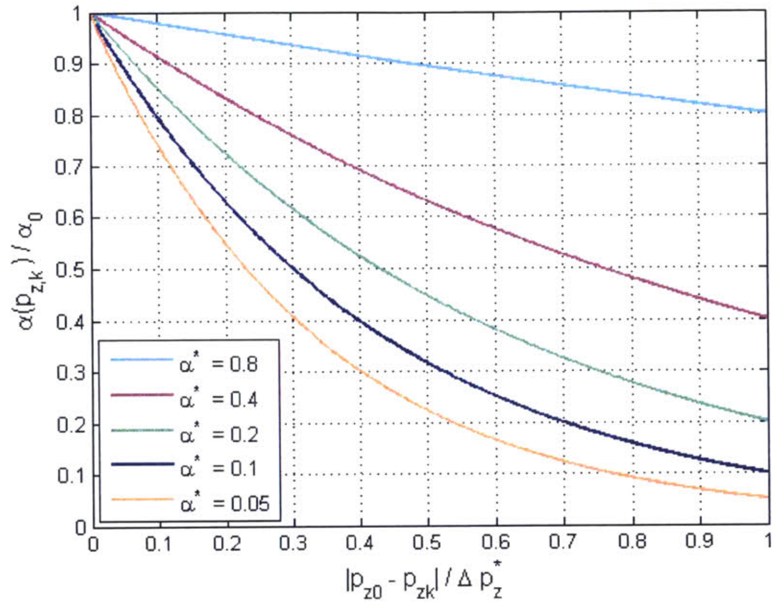
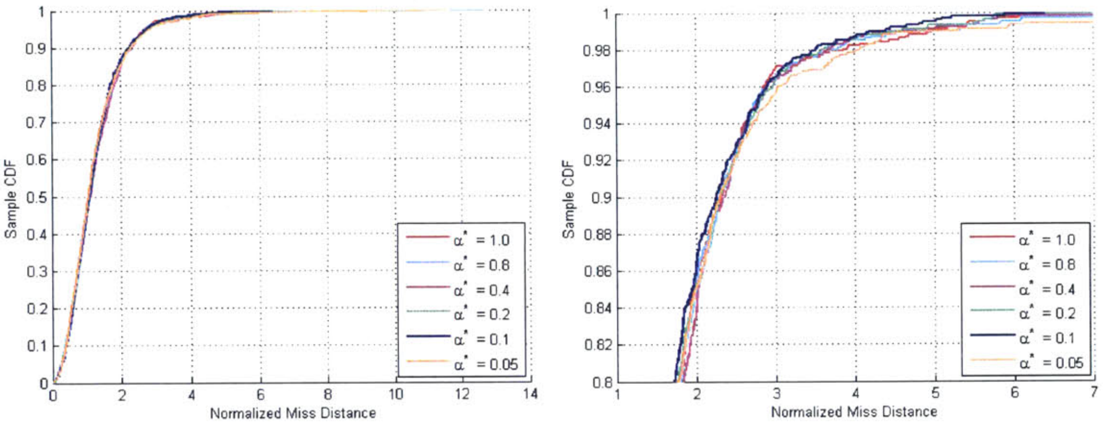


Figure 4-12: Decay of $\alpha$ with altitude change $|p_{z,0} - p_{z,k}|$ for each ratio of $\alpha^*$

are tested. Figure 4-12 displays how the ratio of $\alpha(p_{z,k})/\alpha_0$ changes over the altitude range $\Delta p_z{}^*$ for each value of $\alpha^*$ considered. When the parafoil is at the initial altitude of each terminal guidance phase (i.e., $p_{z,I} = p_{z,0}$ and $|p_{z,0} - p_{z,G}| = \Delta p_z{}^*$), the ratio of $\alpha(p_{z,F})$ at the altitude of the final trajectory state $p_{z,F}$, over $\alpha(p_{z,0})$ at the current altitude can be summarized by $\alpha(p_{z,F})/\alpha(p_{z,0}) = \alpha^*$ if a path to the target altitude is successfully identified by the planner (i.e., $p_{z,F} = p_{z,G}$). As the parafoil descends, the current value of $|p_{z,0} - p_{z,F}|$ decreases such that $|p_{z,0} - p_{z,F}| < \Delta p_z{}^*$, and the minimum possible value of $\alpha(p_{z,F})$ increases along the corresponding $\alpha^*$ exponential curve shown in Figure 4-12. Likewise, the value of $\alpha(p_{z,0}) = \alpha_0$ serves to upper bound the value of $\alpha$ at the root of each trajectory throughout the descent, as the horizon between future trajectory states $|p_{z,0} - p_{z,k}|$ decreases to zero. By computing the decay factor $\lambda$ according to (4.3), Figure 4-12 demonstrates how the selection of $\alpha^*$ also affects the shape of the exponential curves, ranging from near linear at $\alpha^* = 0.8$, to a much steeper exponential decay at $\alpha^* = 0.05$.

Table 4.9 and Figure 4-13 compare the normalized miss distance data over 1000 Monte Carlo simulation trials on the valley terrain using each value of $\alpha^*$ and $\alpha_0 =$



(a) full CDF

(b) zoomed in to $(1,7) \times (0.8,1)$

Figure 4-13: Normalized miss distance CDF for $\alpha^*$, 1000 trials

Table 4.9: Normalized miss distance data for $\alpha^*$, 1000 trials

| **Ratio, $\alpha$** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **99%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha^* = 0.05$ | 1.285 | 1.060 | 1.070 | 1.769 | 2.302 | 2.900 | 4.068 | 4.827 | 11.607 |
| $\alpha^* = 0.1$ | 1.248 | 0.818 | 1.099 | 1.730 | 2.253 | 2.780 | 3.513 | 4.182 | 6.391 |
| $\alpha^* = 0.2$ | 1.268 | 0.914 | 1.103 | 1.771 | 2.300 | 2.804 | 3.589 | 4.483 | 9.859 |
| $\alpha^* = 0.4$ | 1.272 | 0.936 | 1.114 | 1.810 | 2.265 | 2.777 | 3.654 | 4.442 | 11.356 |
| $\alpha^* = 0.8$ | 1.285 | 1.031 | 1.110 | 1.806 | 2.284 | 2.733 | 3.711 | 4.773 | 12.737 |
| $\alpha^* = 1.0$ | 1.283 | 0.962 | 1.107 | 1.809 | 2.330 | 2.746 | 3.868 | 4.998 | 12.737 |

500 selected previously. For comparison, the miss distance data for $\alpha_0 = 500$ and $\alpha^*$ $= 1$ from Table 4.8 (i.e., no decay, $\lambda = 0$) is also included in Figure 4-13 and Table 4.9. From the results in Figure 4-13, the decay factor $\lambda$ is shown to have minimal effect on the shape of the miss distance CDF with the exception of the worst-case trials in the tail of the distribution. In this region, the ratio of $\alpha^* = 0.1$ provides the best miss distance performance above the 98th percentile and produces the shortest distribution tail, with a maximum miss distance of 6.391 times the BLG median accuracy. This represents nearly a factor of 2 reduction in the maximum miss distance produced by $\alpha^* = 1$ over the same 1000 Monte Carlo trials.

For a small fraction of trials within each set of Monte Carlo simulations, aspects such as shifting winds and tracking/modeling errors resulted in a potentially dangerous planning scenario within the valley terrain environment despite the uncertainty sampling and risk-based penalties examined above. The data in Table 4.9 and Figure 4-13 suggests that significantly decaying the risk penalty of long-term trajectory

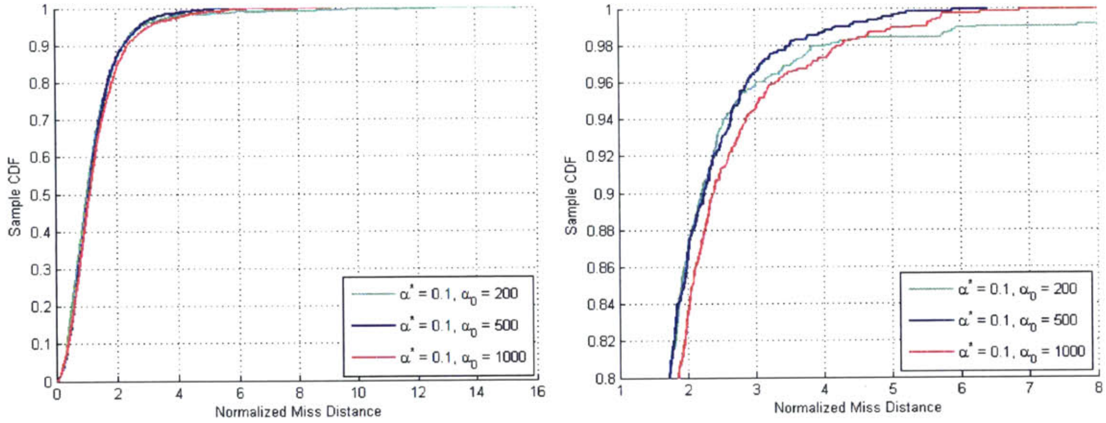
Table 4.10: Normalized miss distance data for refined $\alpha^*$ range, 1000 trials

| **Ratio** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **99%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha^* = 0.075$ | 1.262 | 0.945 | 1.077 | 1.760 | 2.311 | 2.843 | 3.779 | 4.580 | 9.794 |
| $\alpha^* = 0.1$ | 1.248 | 0.818 | 1.099 | 1.730 | 2.253 | 2.780 | 3.513 | 4.182 | 6.391 |
| $\alpha^* = 0.15$ | 1.259 | 0.915 | 1.064 | 1.781 | 2.306 | 2.774 | 3.627 | 4.304 | 9.478 |

states using $\alpha^* = 0.1$ can allow the planner to better account for imminent danger in these situations. When the cost $J_T$ of all trajectory states is equal (i.e., $\alpha^* = 1$), short-term course corrections by the planner to avoid terrain obstacles may incur a large penalty over the remaining trajectory states. This increase in cost can prevent CC-BLG from identifying trajectory solutions that successfully mitigate near term risk. On the other hand, as the rate of decay becomes large, as for $\alpha^* = 0.05$, the results in Table 4.9 suggest that the risk penalty on future trajectory states may not be high enough to enable long-term planning robustness against the most dangerous wind-terrain interactions. This degrades the performance of the CC-BLG algorithm by limiting the amount of useful cost information about the future wind uncertainty that is available during the design of the trajectory plan. Such an effect increases the number of outliers in the tail of the distribution due to worst-case terrain impacts. The remaining decay rates of $\alpha^* = 0.8$, 0.4, and 0.2 produce marginal improvements over $\alpha^* = 1$ above the 98th percentile, but are not very successful at removing the large distribution tail, as shown in Table 4.9.

Given these observations, a final series of Monte Carlo simulation trials are conducted to determine whether additional sensitivity exists for improved performance on the valley terrain. Table 4.10 compares the distribution of miss distances for 1000 Monte Carlo simulations using two additional values of $\alpha^* = 0.075$ and $\alpha^*= 0.15$. The results indicate that additional sensitivity to the value of $\lambda$ exists between $\alpha^* = 0.2$ and $\alpha^* = 0.05$, but that the distribution remains largely the same as in Table 4.9. The value of $\alpha^* = 0.1$ continues to provide the best mean and worst-case performance by reducing the number of CC-BLG terrain impacts above the 98th percentile.

The final experiment compares the value of $\alpha^* = 0.1$ using two alternative combinations of $\alpha_0 = 200$ and $\alpha_0 = 1000$. The results in Table 4.11 and Figure 4-14



(a) full CDF

(b) zoomed in to $(1, 8) \times (0.8, 1)$

Figure 4-14: Normalized miss distance CDF for final $\alpha^*$ and $\alpha_0$, 1000 trials

Table 4.11: Normalized miss distance data for final $\alpha^*$ and $\alpha_0$, 1000 trials

| **Weight** | **Decay Factor** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha_0 = 200$ | $\alpha^* = 0.1$ | 1.287 | 1.306 | 1.037 | 1.727 | 2.226 | 2.742 | 4.033 | 15.520 |
| $\alpha_0 = 500$ | $\alpha^* = 0.1$ | 1.248 | 0.818 | 1.099 | 1.730 | 2.253 | 2.780 | 3.513 | 6.391 |
| $\alpha_0 = 1000$ | $\alpha^* = 0.1$ | 1.333 | 0.975 | 1.125 | 1.865 | 2.390 | 3.063 | 4.271 | 9.145 |

Table 4.12: Summary of tuned CC-BLG parameters

| **Parameter** | **Value** |
|---|---|
| $\alpha_0$ | 500 |
| $\lambda_1$ | 0.00658 |
| $\lambda_2$ | 0.00768 |
| $t_P$ | 17.6 $s$ |
| $N_u$ | 7 |

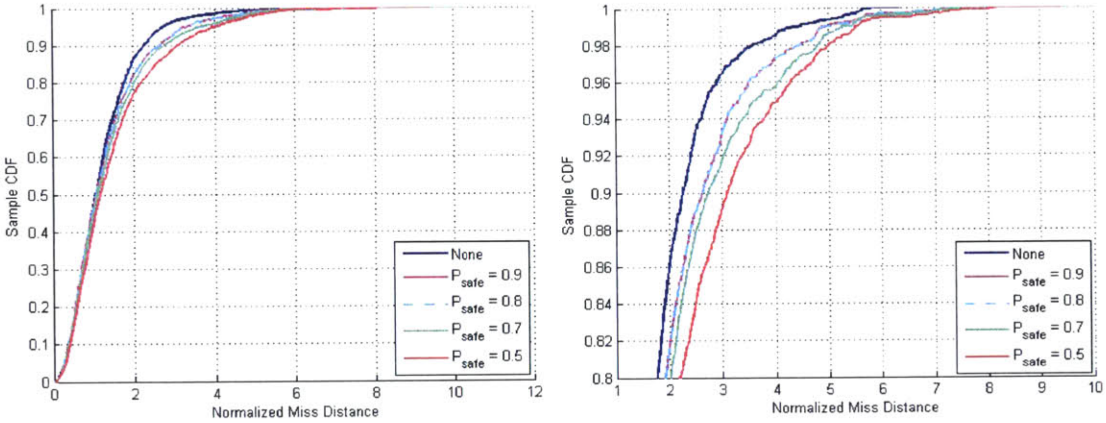
reveal that as the value of $\alpha_0$ is adjusted, the CC-BLG performance above the 95th percentile tends to degrade relative to $\alpha_0 = 500$. This suggests that the values of $\alpha_0$ $= 500$ and $\alpha^* = 0.1$ constitute an appropriate combination of cost parameters within the CC-BLG multi-objective function. In this work, Phase 1 and Phase 2 of CC-BLG terminal guidance are conducted over an altitude change of $\Delta p_{z,1}{}^* = 350$ meters and $\Delta p_{z,2}{}^* = 300$ meters, respectively. As a result, the selected value of $\alpha^* = 0.1$ corresponds to a value of $\lambda_1 = 0.00658$ for Phase 1 and $\lambda_2 = 0.00768$ for Phase 2, and will be used with $\alpha_0 = 500$ for all Monte Carlo simulation experiments throughout the remainder of this thesis. Table 4.12 provides a summary of all four tuned CC-BLG parameters examined in this chapter.

## 4.6 Chance Constraints

In the design of the CC-BLG algorithm, "soft" chance constraints are applied to enforce probabilistic feasibility through the development of a novel risk-based objective function (3.11) presented in Section 3.3.3. This section considers the effect of reintroducing the hard user-defined bound on probabilistic feasibility $p_{safe}$ in addition to the penalty terms on risk outlined above.

Figure 4-15 and Table 4.13 contain the results from 1500 Monte Carlo trials in the valley terrain environment for each value of $p_{safe} \in \{0.9, 0.8, 0.7, 0.5\}$. An additional set of trials (labeled "None") is also conducted for the scenario in which only soft chance constraints are used, as demonstrated previously in Section 4.3—4.5. The CDFs of normalized miss distance provided in Figure 4-15(a)–(b) indicate an improvement in CC-BLG landing accuracy as the feasibility bound $p_{safe}$ is tightened from 0.5 to 0.9. The distributions for $p_{safe} = 0.9$ and $p_{safe} = 0.8$ are almost exactly the same, which suggests that both feasibility bounds generate a similar solution space for the CC-BLG optimizer over each identical set of Monte Carlo trials. As the value of $p_{safe}$ is further reduced, the risk tolerance increases via the chance constraint $p_{collide} \leq (1 - p_{safe})$. Given the uncertainty in the wind environment, this can raise the level of danger in complex terrain by permitting the design of trajectories with low probabilities of safety. When hard constraints on probabilistic feasibility are employed, the results in Figure 4-15 and Table 4.13 suggest that decreasing the value of $p_{safe}$ produces an increasing number of large CC-BLG miss distances in the tail of the distribution due to off-nominal terrain collisions.

However, Table 4.13 and Figure 4-15 also demonstrate that using soft chance constraints alone, without any hard bounds on probabilistic feasibility, results in better CC-BLG performance above the 50th percentile than any other constraint scenario. This seemingly contradictory result can be explained by the way hard constraints on probabilistic feasibility limit the size of the solution space and may interrupt the process of trajectory optimization. Sudden changes in the wind that perturb the nominal trajectory plan can increase the number of covariance samples along the state dis-



(a) full CDF

(b) zoomed in to $(1, 8) \times (0.8, 1)$

Figure 4-15: Normalized miss distance CDF using $p_{safe}$ constraints, 1500 trials

Table 4.13: Normalized miss distance data using $p_{safe}$ constraints, 1500 trials

| **Weight** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| None | 1.242 | 0.840 | 1.066 | 1.766 | 2.267 | 2.728 | 3.592 | 6.345 |
| $p_{safe} = 0.9$ | 1.335 | 1.024 | 1.078 | 1.919 | 2.634 | 3.318 | 4.478 | 8.503 |
| $p_{safe} = 0.8$ | 1.337 | 1.028 | 1.079 | 1.921 | 2.637 | 3.318 | 4.478 | 8.894 |
| $p_{safe} = 0.7$ | 1.411 | 1.106 | 1.121 | 2.022 | 2.744 | 3.658 | 4.804 | 10.274 |
| $p_{safe} = 0.5$ | 1.522 | 1.196 | 1.188 | 2.192 | 3.097 | 4.057 | 5.021 | 9.420 |

tribution that intersect the terrain environment. When hard constraints are used, if the sum of probability weights from collided covariance samples at a given trajectory state (3.8) exceeds the limit on probabilistic feasibility $p_{collide} > (1 - p_{safe})$, the dynamic propagation of the parafoil trajectory is terminated, and the remainder of the previous trajectory plan is pruned. The high cost of $J_{BLG}$ incurred by terminating the trajectory plan above the target altitude is factored into the CC-BLG optimization and can therefore disrupt the sequence of trajectory refinement. Additionally, for situations in which some risk may be required in order to guide the parafoil away from nearby obstacles, hard bounds on probabilistic feasibility may prevent the optimizer from identifying any trajectory plan which satisfies the constraints. This can result in an immediate termination of trajectory propagation with no plan forward.

Conversely, when only soft constraints are used, temporary increases in risk are tolerated by the planner and trajectory propagation continues until the nominal trajectory plan $\widehat{\mathbf{x}}_t$ either intersects the terrain, or reaches the desired target altitude. This enables the optimizer to refine the current trajectory plan in subsequent iterations by removing the regions of increased risk. The results in Table 4.13 and Figure 4-15 suggest that tolerating temporary increases in risk without terminating trajectory propagation can ultimately improve the quality of the risk-based optimization used within CC-BLG. Based on these improvements, a strategy of soft chance constraints will be used throughout the remainder of this thesis in order to directly minimize risk in the absence of hard feasibility constraints.

## 4.7 Summary

This chapter presented a detailed design analysis of the CC-BLG parafoil terminal guidance algorithm. Through a series of Monte Carlo simulation experiments, Section 4.3 demonstrated the robustness benefits of increasing the size of the sampling standard deviation $\sigma$ using an additional set of covariance samples along the state uncertainty distribution. In order to estimate the probability of constraint violation, the weighted analytic uncertainty sampling approach presented in Section 3.3.2 was found to generate improved CC-BLG performance relative to the previous Analytic CC-RRT approach. In Section 4.4, the reachability set approximation and cost-to-go function $J_{CTG}$ were shown to be effective at reducing the number of terrain impacts caused by finite horizon trajectory planning in complex terrain. The CC-BLG method of unconstrained optimization using "soft" chance constraints was also found to provide greater planning flexibility over alternative formulations using hard feasibility constraints (Section 4.6). Lastly, from a parametric study of the CC-BLG reachability set approximation and risk-based objective function in Sections 4.4–4.5, final values were selected for the variables of $\alpha_0, \lambda, t_P$, and $N_u$ to be used throughout the remainder of simulation experiments. Chapter 5 proceeds with a direct comparison between the BLG and CC-BLG algorithms over a variety of complex terrain scenarios using Draper Laboratory's high fidelity parafoil simulator introduced in Section 4.2.

# Chapter 5

# Simulation Results

## 5.1 Introduction

This chapter presents simulation results demonstrating the effectiveness of the CC-BLG algorithm for robust parafoil guidance in complex terrain and wind environments. The improvements provided by the augmented CC-BLG components of real-time wind modeling and analytic uncertainty sampling are compared directly with the state-of-the-art Band-Limited Guidance algorithm (BLG) in three challenging terrain scenarios. Section 5.2 evaluates the performance of the final CC-BLG algorithm developed in Chapter 4 through a set of Monte Carlo simulations in the valley terrain environment. Two Grand Canyon terrain scenarios are also considered in Section 5.3 to further validate the CC-BLG algorithm in highly constrained terrain geometries. The results of these experiments demonstrate (1) superior CC-BLG mean and worst-case landing accuracy over BLG in all three terrain scenarios, and (2) a significant reduction in the number of CC-BLG collisions with the terrain environment. Despite the incorporation of risk-based cost objectives, this chapter also demonstrates that the CC-BLG algorithm retains nominal BLG miss distance performance on flat terrain (Section 5.4), while exhibiting either identical, or improved, mean and worst-case landing speed performance across multiple terrain scenarios (Section 5.5).

## 5.2 Valley Terrain Simulations

In this chapter, the high fidelity simulation environment and Monte Carlo testing procedure outlined in Section 4.2 are used to compare the BLG and CC-BLG algorithms over an identical series of randomized initial conditions. A large data set of 2500 Monte Carlo trials is collected for each terrain scenario in order to exhaustively test the performance of both algorithms, while allowing for the observation of possible outliers. To establish a comparison of relative terminal guidance performance, the miss distance data from these experiments is once again normalized using the median landing accuracy of the BLG algorithm. The software for the BLG and CC-BLG algorithms is written in C and executed on a 3.2 GHz Intel i5-3470 desktop computer using a maximum of 50 Nelder-Mead optimization iterations per second. Although both BLG and CC-BLG typically converged in less than 50 iterations, this upper bound ensured that each algorithm could be run in real-time using a 1 Hz planning/executing cycle with less than a 60% duty cycle.

Figure 5-1 provides the normalized miss distance CDF from 2500 Monte Carlo trials of the BLG and CC-BLG algorithms in the valley terrain environment. While both algorithms demonstrate almost identical performance below the 80th percentile, CC-BLG provides a significant improvement over BLG in the upper 20% of the distribution. In particular, the statistics in Table 5.1 reveal that CC-BLG produces an improvement in landing accuracy of more than 15% by the 90th percentile, and at least a factor of 2 reduction in miss distance for all worst-case trials above the 95th percentile. It can also be observed from Table 5.1 that although the median CC-BLG landing accuracy increases by 5% relative to BLG in this scenario, the mean landing accuracy is improved by more than 21%. The effect on median miss distance can be attributed to the trade-offs between the $J_{CC}$ and $J_{BLG}$ cost components of the CC-BLG objective function (3.13), which simultaneously balance trajectory risk with the parafoil performance metrics of landing accuracy and landing speed. Despite the additional robustness modifications during trajectory optimization, the results in Table 5.1 demonstrate that the overall effect on median landing accuracy is small.

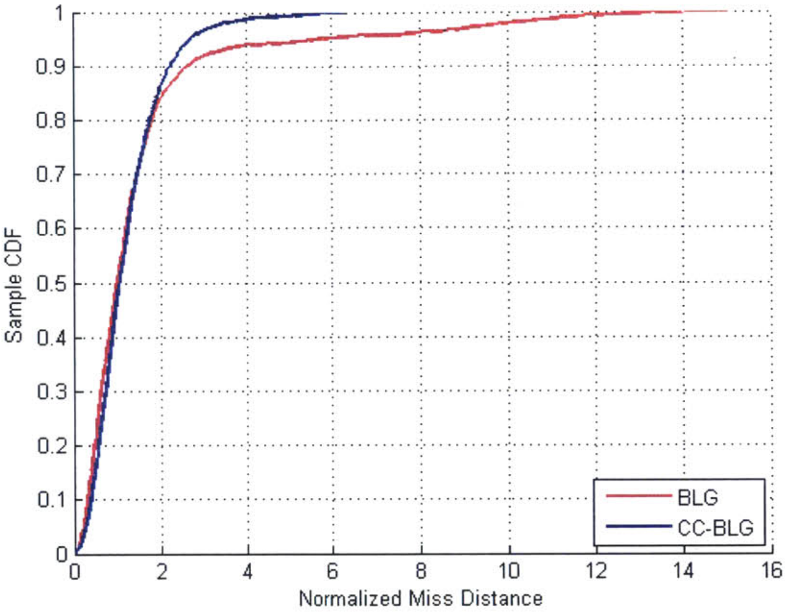


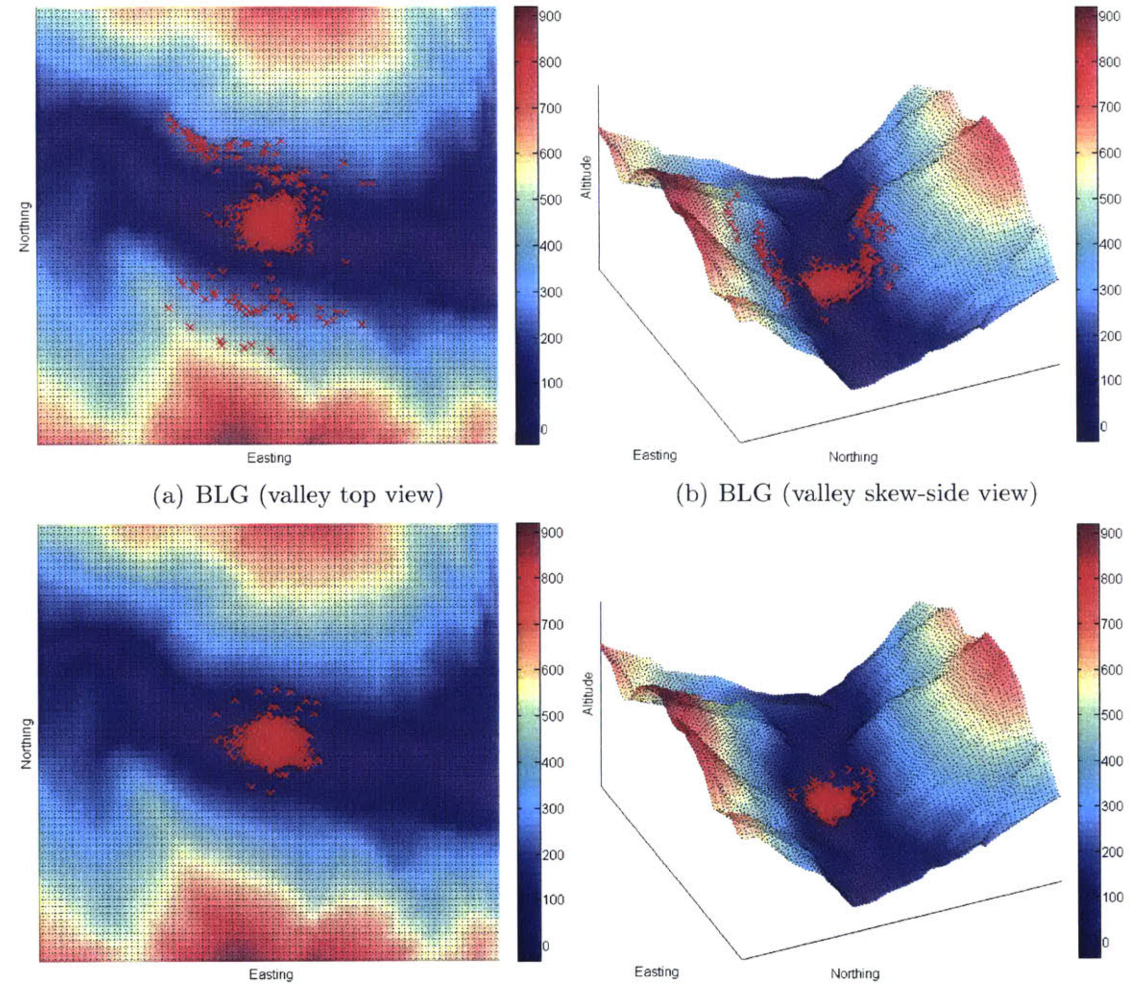
Figure 5-1: Normalized miss distance CDF for valley terrain, 2500 trials

Table 5.1: Normalized miss distance data for valley terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.56 | 2.10 | 1.00 | 1.81 | 2.64 | 5.62 | 10.18 | 15.04 |
| CC-BLG | 1.23 | 0.83 | 1.05 | 1.76 | 2.23 | 2.71 | 3.53 | 6.34 |

On the other hand, by incorporating soft chance constraints on the risk produced by future wind uncertainty, the CC-BLG algorithm reduces the number of large miss distances caused by off-nominal terrain collisions. This results in a CC-BLG standard deviation that is more than 2.5 times smaller than BLG's.

Figure 5-2 displays the impact points from each set of 2500 simulations of the BLG and CC-BLG algorithms described above. In Figure 5-2(a)-(d), the elevation change above the Phase 2 target/goal is denoted in meters by the transition of terrain color from blue (lowest altitude) to red (highest altitude), while each impact point is denoted using a red 'x'. For these simulations, the narrow terrain geometry and steep valley slope prevent the parafoil from approaching the target location along the north-south direction. Given the large parafoil turning radius and underactuated dynamics, this produces a challenging terminal guidance scenario that frequently places the parafoil within close proximity to the terrain throughout the descent.



(a) BLG (valley top view)

(b) BLG (valley skew-side view)

(c) CC-BLG (valley top view)

(d) CC-BLG (valley skew-side view)

Figure 5-2: BLG and CC-BLG impact points on the valley terrain, 2500 trials (meters)

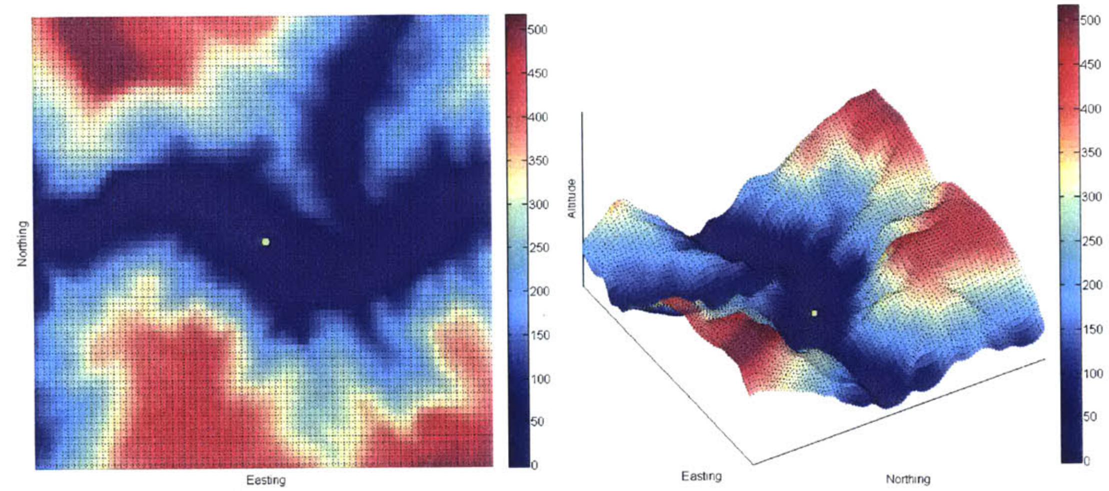
From the BLG impact points provided in Figure 5-2(a)-(b), numerous collisions can be observed within the valley environment. These collisions are caused by the effects of unanticipated wind disturbances that perturb the parafoil along the nominal trajectory plan. As shown in Figure 5-2(a)-(b), because the BLG approach to trajectory optimization does not consider the potential risk produced by future variations in the wind environment, robustness to uncertainty cannot be maintained.

Figure 5-2(c)-(d) display the impact points produced by the CC-BLG algorithm over the identical set of 2500 Monte Carlo trials shown in Figure 5-1 and Table 5.1 above. These results illustrate that the total number of terrain collisions has been significantly reduced relative to Figure 5-2(a)-(b) such that the distribution of CC-BLG impact points is tightly grouped in the vicinity of the target. Additionally, while numerous BLG collisions can be observed at high altitudes along both the yellow and light-blue terrain elevation bands, the vast majority of CC-BLG simulation trials in Figure 5-2(c)-(d) result in safe landings at the desired target altitude. Although a small number of outliers are present, all CC-BLG impacts occur at altitudes less than 250 meters above the target location. This demonstrates that the CC-BLG implementation of reachability set approximation (Section 3.3.4) has eliminated the collisions caused between terminal guidance phase transitions. The absence of high altitude impacts also indicates that the CC-BLG method of analytic uncertainty sampling successfully maintains parafoil safety during all Phase 1 terminal guidance segments when the penalty on risk (3.11) is highest. Over the entire set of 2500 trials in the valley terrain, CC-BLG produces a crash percentage of only 1.12%. As compared to the BLG crash percentage of 7.28%, this represents more than a factor of 6 reduction in the number of off-nominal collisions with the terrain environment.

## 5.3 Canyon Terrain Simulations

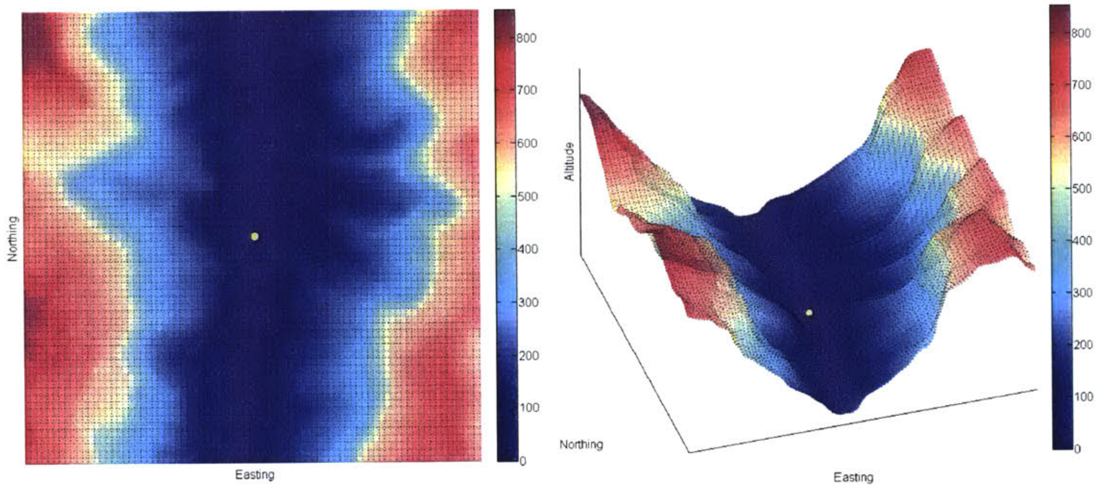
This section further demonstrates the effectiveness of the CC-BLG algorithm in tight geometry environments using two canyon terrain scenarios. Both scenarios, referred to in this work as canyon-1 and canyon-2, were developed using DTED2 data from the Grand Canyon, and are plotted for convenience in Figure 5-3 and Figure 5-4.

The target for each terrain scenario is placed at the bottom of the canyon as indicated by the yellow circle. The region of lowest altitude within the canyon-1 terrain is depicted in Figure 5-3 by the dark blue elevation band that runs primarily from east to west along the middle of the map. This environment contains many complex features including a tight terrain bottleneck both along the west end of the map, and in the vicinity of the goal, as well as a ravine to the northeast formed by two terrain ridges elevated roughly 200 meters above the target altitude. Such features represent potentially dangerous environmental hazards that complicate the planning



(a) canyon-1 top view (b) canyon-1 skew-side view

Figure 5-3: Canyon-1 terrain environment (meters)



(a) canyon-2 top view (b) canyon-2 skew-side view

Figure 5-4: Canyon-2 terrain environment (meters)

process and make robust parafoil guidance essential due to limited maneuverability at lower altitudes. The canyon-2 scenario (shown in Figure 5-4) is oriented along the north-south direction and provides extremely steep changes in elevation along the canyon walls. As a result, obstacle avoidance must begin from the initialization of Phase 1 terminal guidance in order to maintain parafoil safety in this terrain. Likewise, Figure 5-4 reveals that the region of terrain elevation at the target altitude (colored dark blue) is very narrow, and sinks sharply below the elevation of the canyon walls on either side. This constrains the parafoil final approach to be along

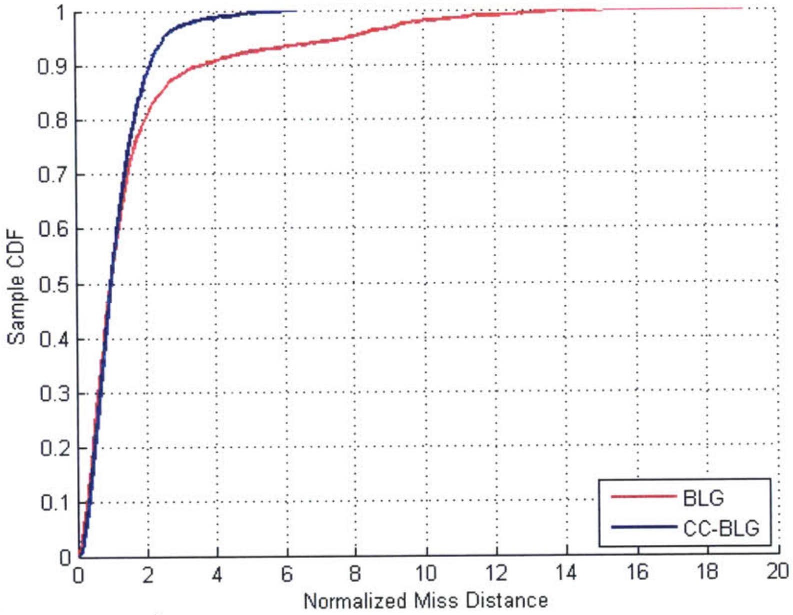


Figure 5-5: Normalized miss distance CDF for canyon-1 terrain, 2500 trials

Table 5.2: Normalized miss distance data for canyon-1 terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.74 | 2.35 | 1.00 | 1.99 | 3.67 | 7.89 | 10.10 | 19.10 |
| CC-BLG | 1.17 | 0.81 | 1.00 | 1.71 | 2.14 | 2.56 | 3.42 | 6.41 |

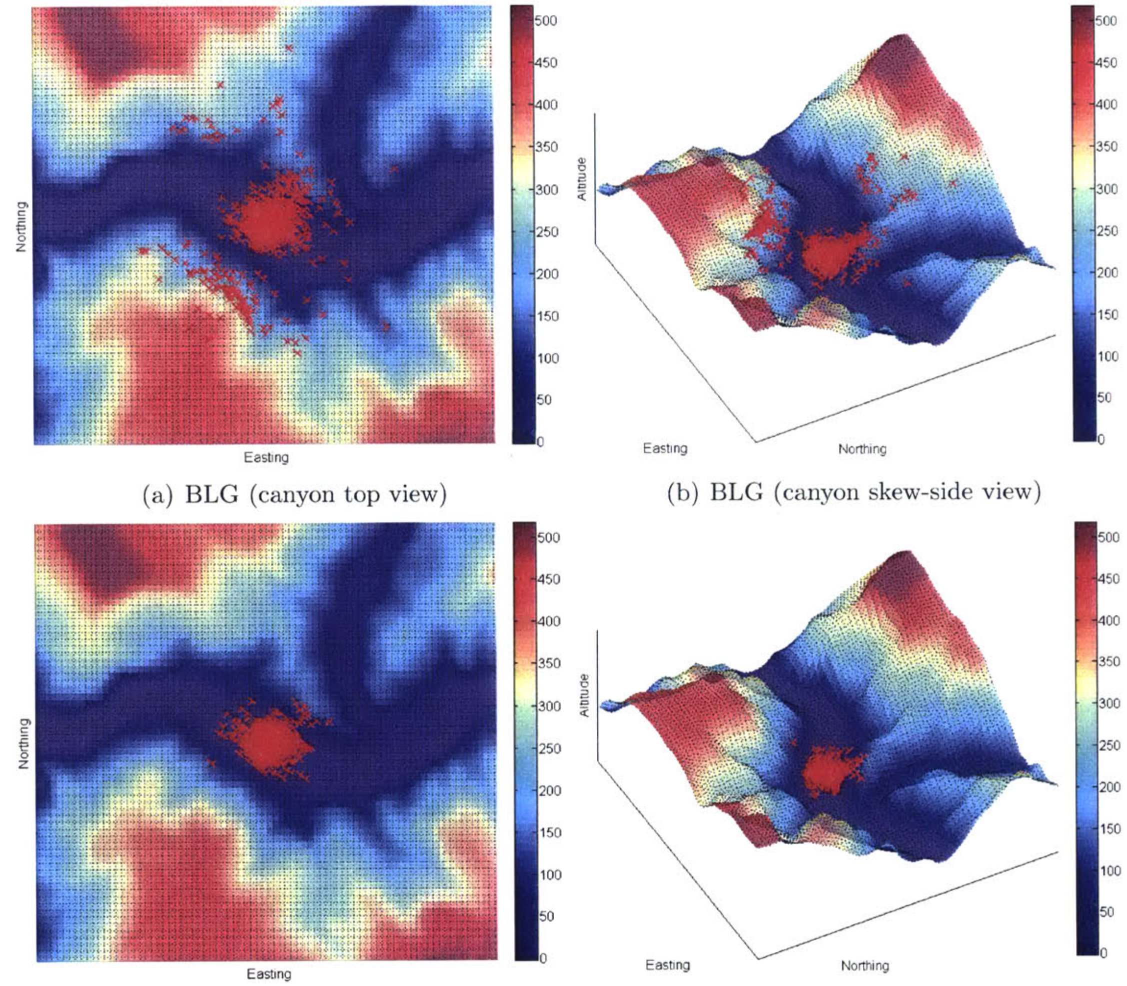
the north-south direction so as to obtain a safe landing at the desired target location.

Figure 5-5 and Table 5.2 provide the normalized miss distance CDF and statistics for 2500 Monte Carlo trials of the BLG and CC-BLG algorithms within the canyon-1 terrain. In this terrain scenario, both algorithms produce similar performance over the lower half of the CDF depicted in Figure 5-5. For the subset of trials below the 60th percentile, the level of terrain and wind interaction is minimal such that BLG and CC-BLG can both find feasible trajectories to the target which remain robust to future wind uncertainty. However, the results in Figure 5-5 demonstrate that the performance of the two algorithms begins to diverge around the 70th percentile as CC-BLG provides superior landing accuracy over the remainder of the distribution. By the 80th and 90th percentiles, the statistics in Table 5.2 indicate that the miss distances produced by CC-BLG are 14% and 41% lower than BLG, respectively.

The improvement in landing accuracy demonstrated by the CC-BLG algorithm is even more significant for the final 250 trials in the upper 10% of the distribution. Here, CC-BLG produces more than a factor of 3 reduction in miss distance over BLG by the 95th percentile, and between a factor of 2-3 reduction in miss distance for all trials above the 98th percentile. The shorter tail of the CDF distribution shown in Figure 5-5 can be attributed to the increased robustness of the CC-BLG algorithm to pathological wind uncertainty conditions. This reveals the effectiveness of both the wind classifier and analytic uncertainty model for providing reliable predictions of future environmental risk. Incorporating this risk directly into the process of trajectory optimization causes the CC-BLG mean landing accuracy to be improved by over 32% relative to the nominal BLG algorithm, while the median landing accuracy remains unaffected by the additional $J_{CC}$ and $J_{CTG}$ cost objectives.

The set of impact points in Figure 5-6 reveal the sensitivity of the nominal BLG algorithm to dynamic wind disturbances in complex and highly constrained terrain environments. Collisions produced by BLG in the canyon-1 scenario can be observed in Figure 5-6(a)-(b) both at high altitudes and at significant distances from the target location. A particularly high concentration of BLG terrain collisions exists along the south canyon wall at altitudes between 250–350 meters above the target, and along the north canyon ridge which acts to constrain the parafoil final approach direction. These sections of the map represent locations where the parafoil is frequently within close proximity to the terrain while attempting to turn toward the middle of the canyon. Unanticipated wind perturbations during these maneuvers have a significant effect on BLG performance when tight geometry constraints necessitate planning in the vicinity of environmental obstacles. Replanning alone, without robustness modifications, is insufficient to ensure parafoil safety in these situations.

The CC-BLG impact points in Figure 5-6(c)-(d) demonstrate the significant improvement in planning robustness provided by the $J'$ risk-based objective function introduced in Chapter 3. Over this set of simulations, CC-BLG dramatically reduces the number of collisions within the canyon-1 terrain environment. As in Figure 5-2, no CC-BLG collisions occur during Phase 1 terminal guidance or at altitudes greater



(a) BLG (canyon top view)

(b) BLG (canyon skew-side view)

(c) CC-BLG (canyon top view)

(d) CC-BLG (canyon skew-side view)

Figure 5-6: BLG and CC-BLG impact points in canyon-1 terrain, 2500 trials (meters)

than 250 meters above the target location. Figure 5-6(c)-(d) reveal that a small concentration of CC-BLG terrain impacts are present along the north canyon ridge due to the parafoil's limited maneuverability in the vicinity of the target. Despite the presence of these outliers, the overall distribution of CC-BLG impact points remains clustered around the target location. This indicates that the vast majority of trials arrive safely at the desired altitude such that flare can be performed, and soft landings achieved. The result is that the CC-BLG crash percentage of 1% is more than 10 times smaller than the BLG crash percentage of 10.7% for the same set of 2500 Monte Carlo trials in the canyon-1 terrain scenario.

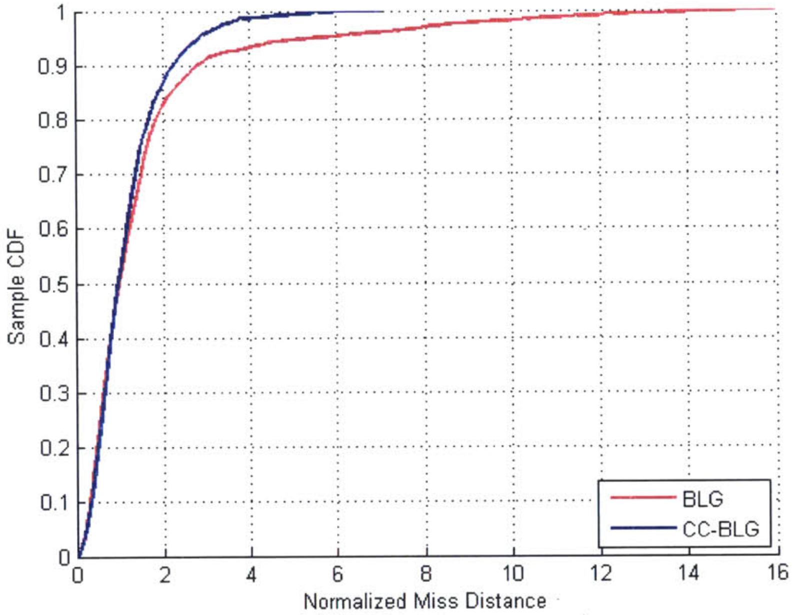


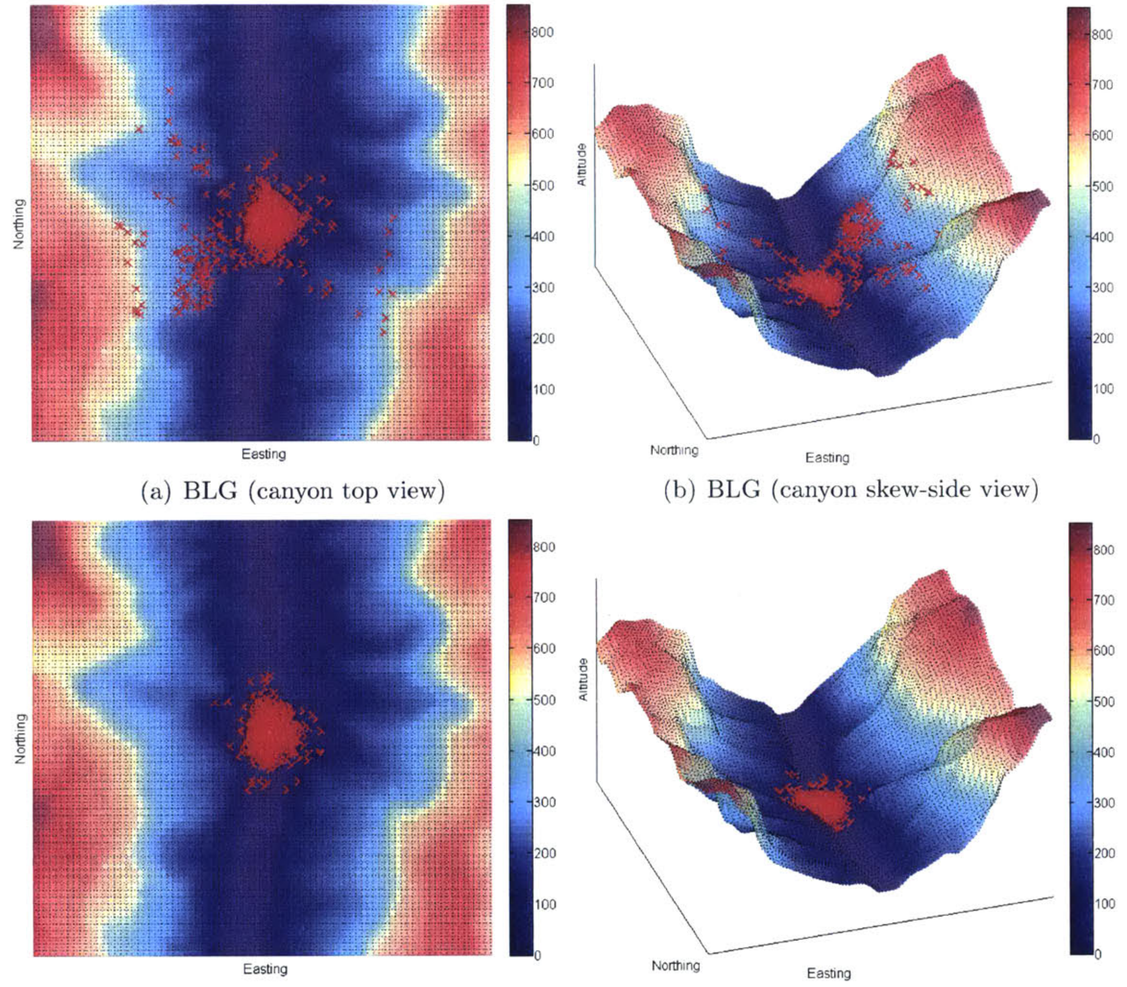
Figure 5-7: Normalized miss distance CDF for canyon-2 terrain, 2500 trials

Table 5.3: Normalized miss distance data for canyon-2 terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.57 | 2.08 | 1.00 | 1.84 | 2.80 | 5.39 | 9.57 | 15.97 |
| CC-BLG | 1.17 | 0.87 | 0.96 | 1.68 | 2.24 | 2.85 | 3.59 | 7.05 |

The results from the set of Monte Carlo simulations within the canyon-2 scenario demonstrate similar improvements in CC-BLG planning robustness throughout this third challenging terrain environment. The normalized CDF and statistics in Figure 5-7 and Table 5.3 reveal that CC-BLG provides superior landing accuracy over BLG beginning around the 50th percentile of the distribution. As a result, the CC-BLG median miss distance is slightly lower than BLG in this terrain scenario. The added robustness provided by the CC-BLG approach to trajectory optimization is once again shown to produce a shorter distribution tail (Figure 5-7), resulting in a 20% reduction in BLG miss distance by the 90th percentile, and more than a factor of 2 reduction in miss distance for all trials above the 98th percentile (Table 5.3). By reducing the number of off-nominal terrain impacts caused by future wind uncertainty, the CC-BLG mean accuracy is also improved relative to BLG by more than 25%.

The impact points over each set of 2500 Monte Carlo trials of the BLG and CC-



(a) BLG (canyon top view)

(b) BLG (canyon skew-side view)

(c) CC-BLG (canyon top view)

(d) CC-BLG (canyon skew-side view)

Figure 5-8: BLG and CC-BLG impact points in canyon-2 terrain, 2500 trials (meters)

BLG algorithms are displayed in Figure 5-8 for the canyon-2 terrain. The results suggest that for this terrain scenario, the steep elevation gradient along the canyon walls makes BLG trajectory planning particularly vulnerable to possible variations in the wind environment. The BLG impact points in Figure 5-8(a)-(b) indicate terrain collisions across numerous elevations bands beginning nearly 600 meters above the target altitude. These collisions are scattered throughout the canyon-2 terrain, with a high concentration of impacts also shown along the protruding section of canyon to the south-west of the target. In contrast, the CC-BLG impact points in Figure 5-8(c)-(d) demonstrate far fewer worst-case collisions within the canyon-2 terrain scenario. Here

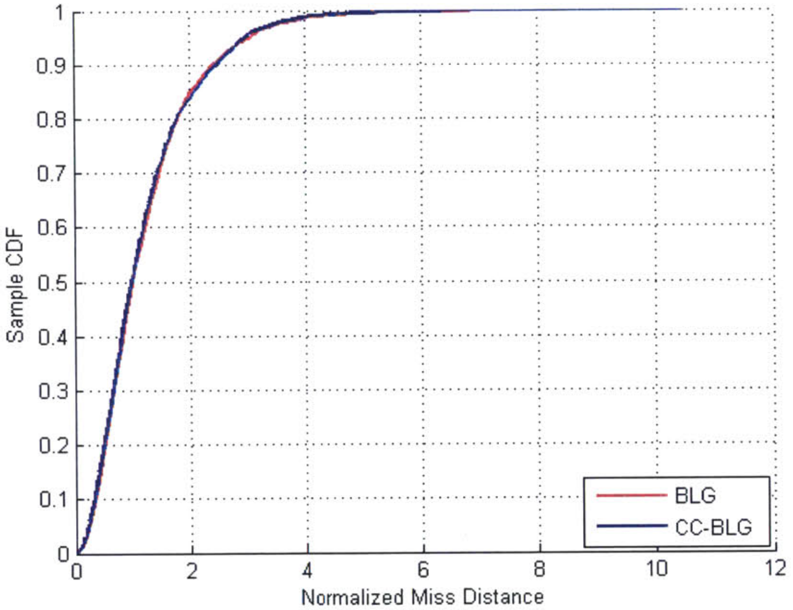


Figure 5-9: Normalized miss distance CDF for flat terrain, 2500 trials

Table 5.4: Normalized miss distance data for flat terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.28 | 0.93 | 1.00 | 1.80 | 2.42 | 3.07 | 3.84 | 10.48 |
| CC-BLG | 1.22 | 0.92 | 0.99 | 1.78 | 2.49 | 2.97 | 3.59 | 10.47 |

again we observe that CC-BLG has successfully eliminated the presence of terrain collisions at high altitude, while the majority of CC-BLG simulations land safely near the target at the bottom of the canyon.

## 5.4 Flat Terrain Simulations

Figure 5-9 and Table 5.4 provide the results from an additional set of 2500 Monte Carlo simulations of the BLG and CC-BLG parafoil terminal guidance algorithms conducted over flat terrain. In the absence of environmental obstacles, Figure 5-9 demonstrates that the distribution of BLG and CC-BLG miss distances are nearly identical, such that CC-BLG successfully maintains the landing accuracy of the nominal BLG planner. This result is expected given the formulation of the CC-BLG risk-based objectives $J_{CC}$ and $J_{CTG}$ presented in Chapter 3.

When the environment is obstacle-free, and significant elevation changes do not exist, the 2D analytic uncertainty sampling technique set forth by CC-BLG provides little to no additional information to the planner. Under these conditions, the set of covariance samples at any given time step will either all be feasible (i.e., $p_{z,t} > T(p_{x,t}, p_{y,t})$) or all be infeasible (i.e., $p_{z,t} \leq T(p_{x,t}, p_{y,t})$), as determined by the feasibility of the nominal trajectory state. Because the $J_{BLG}$ component of CC-BLG in (3.16) is identical to the objective function of the nominal BLG algorithm, the process of CC-BLG trajectory optimization is therefore reduced to the original BLG formulation in the absence of cost penalties from collided covariance samples. This claim is supported by the results in Figure 5-9 and Table 5.4. If robustness modifications are not required due to the lack of terrain variability, conditionals can easily be incorporated into the CC-BLG algorithm to reduce unnecessary computation during low risk guidance scenarios.

## 5.5 Landing Speed

Parafoil landing speed is penalized implicitly for the CC-BLG and BLG algorithms through the weighted error $(\sin(\Delta\psi/2))^2$ between the final parafoil heading and desired upwind heading at the terminal trajectory state. This section compares the distribution of landing speeds produced by both algorithms from the set of flat and valley terrain simulations provided in Section 5.4 and Section 5.2, respectively. Each data sct will be normalized using the median BLG landing speed in the flat terrain scenario. In this way, a comparison of relative performance is established between CC-BLG and BLG, and between the flat and valley landing speed distributions provided by each terminal guidance strategy.

The distributions of BLG and CC-BLG landing speeds in Figure 5-10 and Table 5.5 display no statistically significant differences in the flat terrain scenario. This result further supports that CC-BLG behaves identically to BLG when the risk-based penalties on terrain interaction are negligible throughout the descent. Such performance is desirable by retaining the BLG property of soft target impacts produced by

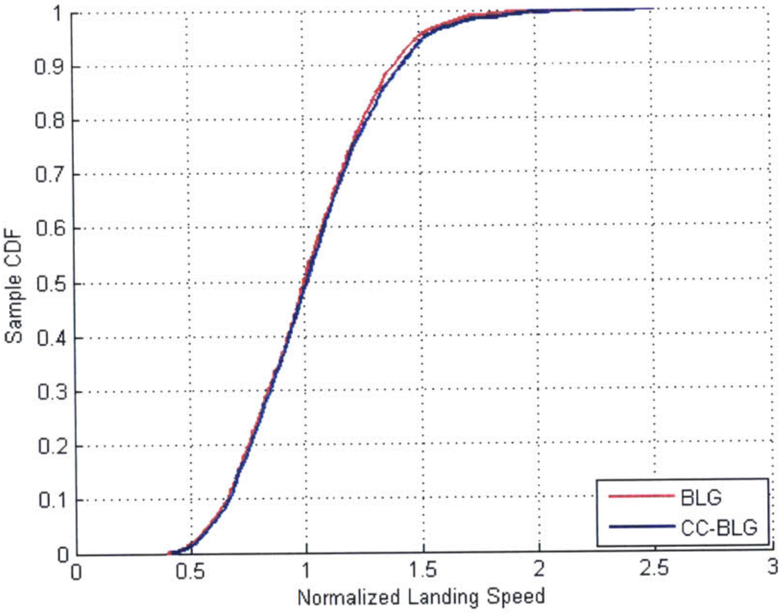


Figure 5-10: Normalized landing speed CDF for flat terrain, 2500 trials

Table 5.5: Normalized landing speed data for flat terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.02 | 0.29 | 1.00 | 1.26 | 1.40 | 1.50 | 1.65 | 2.43 |
| CC-BLG | 1.03 | 0.30 | 1.01 | 1.29 | 1.43 | 1.53 | 1.71 | 2.51 |

the $J_{BLG}$ terminal state objectives presented in (1.3).

Figure 5-11 and Table 5.6 reveal that BLG and CC-BLG also produce similar landing speed performance in the valley terrain scenario over the majority of the distribution below the 90th percentile. The augmented risk penalties within the CC-BLG objective function generate only a marginal increase in landing speed relative to BLG in the middle 50% of the distribution, and less than a 3% increase in median landing speed, as shown in Table 5.6. However, the superior robustness properties of the CC-BLG algorithm improve performance for the set of worst-case trials in the upper 10% of the distribution due to a significant reduction in the number of terrain collisions, shown previously in Figure 5-2. As a result, CC-BLG maintains BLG average landing speed in the valley terrain.

When environmental obstacles are present, the statistics in Table 5.6 indicate an increase in parafoil landing speed for both BLG and CC-BLG relative to their flat ter-

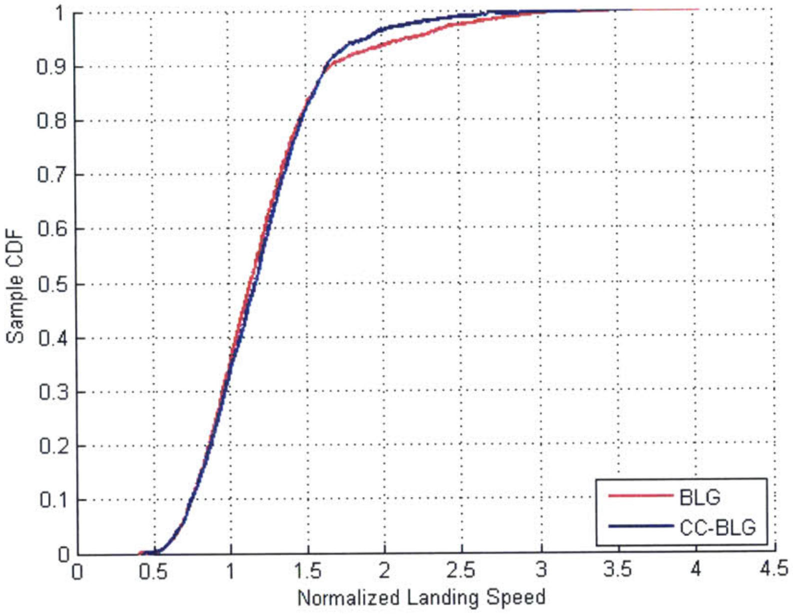


Figure 5-11: Normalized landing speed CDF for the valley terrain, 2500 trials

Table 5.6: Normalized landing speed data for valley terrain comparison, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.22 | 0.46 | 1.14 | 1.46 | 1.68 | 2.21 | 2.64 | 4.04 |
| CC-BLG | 1.21 | 0.40 | 1.17 | 1.48 | 1.65 | 1.91 | 2.30 | 3.61 |

rain counterparts. The tight geometry constraints in the valley scenario make landing into the wind more difficult while simultaneously maintaining both parafoil safety and target accuracy. The flexibility provided by the BLG and CC-BLG method of unconstrained optimization allows these trade-offs to be balanced efficiently throughout terminal guidance. As a result, the increase in mean landing speed for both algorithms is less than 20% relative to the flat terrain scenario. Although the maximum landing speeds produced by BLG and CC-BLG increase due to worst-case collisions within the valley environment, the increase in landing speed is significantly smaller for CC-BLG over all Monte Carlo trials above the 95th percentile.

## 5.6 Summary

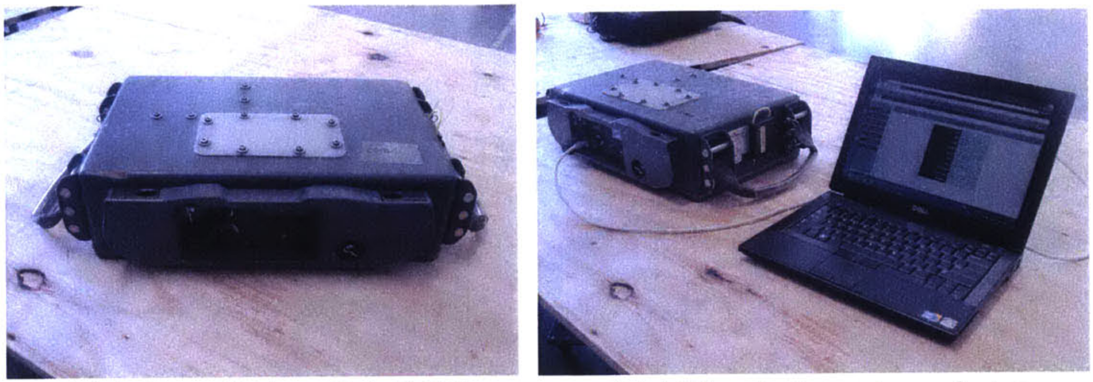
This chapter demonstrates the effectiveness of the CC-BLG algorithm for robust parafoil guidance in three challenging drop scenarios. The method of weighted analytic uncertainty sampling presented in Chapters 3-4 enables CC-BLG to establish a reliable estimate of trajectory feasibility against arbitrary, and potentially aggressive terrain map functions. By directly penalizing the risk of constraint violation during trajectory optimization, CC-BLG is shown to dramatically reduce the number of terrain collisions caused by wind disturbances relative to the state-of-the-art BLG algorithm. This results in superior CC-BLG mean and worst-case landing accuracy relative to BLG in all three constrained terrain environments considered in Sections 5.2 - 5.3. The wind uncertainty model and classification strategy borrowed from Analytic CC-RRT (Chapter 2) allow CC-BLG to dynamically adjust the level of planning conservatism online to reflect variability in the wind conditions. As a result, CC-BLG is able to maintain identical BLG performance when the risk from terrain and wind interaction is minimal. Lastly, CC-BLG is shown to provide identical or improved landing speed performance in the presence of environmental obstacles. This demonstrates that the robustness modifications of CC-BLG do not detract from the terminal state objectives of the nominal BLG algorithm.

# Chapter 6

# Flight Test Results

## 6.1 Introduction

The simulation experiments conducted in Chapters 4 and 5 were used to evaluate the performance of the CC-BLG algorithm in preparation for a flight test demonstration. This chapter presents the CC-BLG flight test results produced from 9 autonomous drops of a full-scale *UltraFly* parafoil system. During each flight test experiment, the presence of terrain was simulated within the onboard guidance software using the maps of the canyon-1 and canyon-2 environments presented previously in Chapter 5. These simulated terrain obstacles allowed for the performance and robustness of the CC-BLG algorithm to be assessed while flight testing, for convenience, over a flat real-world drop zone. Flat terrain drop zones facilitate efficient monitoring and recovery of parafoil systems, and are commonly used for testing developmental guidance strategies [17, 20]. This chapter begins with an overview of the parafoil system and flight test procedure provided in Section 6.2. A series of modifications to the CC-BLG algorithm are then discussed in Section 6.3 in order to comply with the limited computational resources of the available flight hardware. The results from each of the 9 flight test experiments are provided in Section 6.4. Lastly, Section 6.5 compares the statistical performance of the BLG and CC-BLG algorithms in simulation by recreating the flight test conditions from each drop experiment. The results



(a) Airborne Guidance Unit (AGU) (b) AGU and mission planner laptop

Figure 6-1: Airborne Guidance Unit developed by Wamore Inc. (author photo)

from this chapter demonstrate the capability of the modified CC-BLG algorithm to perform robust parafoil guidance subject to real-world testing conditions, hardware limitations, and complex terrain environments.

## 6.2 Implementation

The full-scale *UltraFly* parafoil system (JPADS-ULW) and Airborne Guidance Unit (AGU) developed by Wamore Inc. [21] were used for flight testing of the CC-BLG algorithm. The *UltraFly* AGU is shown in Figure 6-1 and includes the parafoil avionics and onboard guidance computer. The AGU also houses two servo motor actuators for parafoil steering, and a GPS receiver for measuring the parafoil position and ground velocity. The flight software for the guidance, navigation, and control (GN&C) of the *UltraFly* system was developed by Draper Laboratory and is loaded onto the AGU using a JPADS mission planner laptop [32], shown in Figure 6-1(b). The guidance architecture for the *UltraFly* AGU is similar to that of the *Megafly* system in Ref. [32], wherein the additional flight software developed for CC-BLG was used in place of the nominal BLG algorithm during parafoil terminal guidance. This software was written in C and run online using a 32-bit 180 MHz microcontroller (MCU) installed within the *UltraFly* AGU. Flight test data was also recorded onboard the AGU during the parafoil's descent, and was extracted for analysis after each drop experiment.

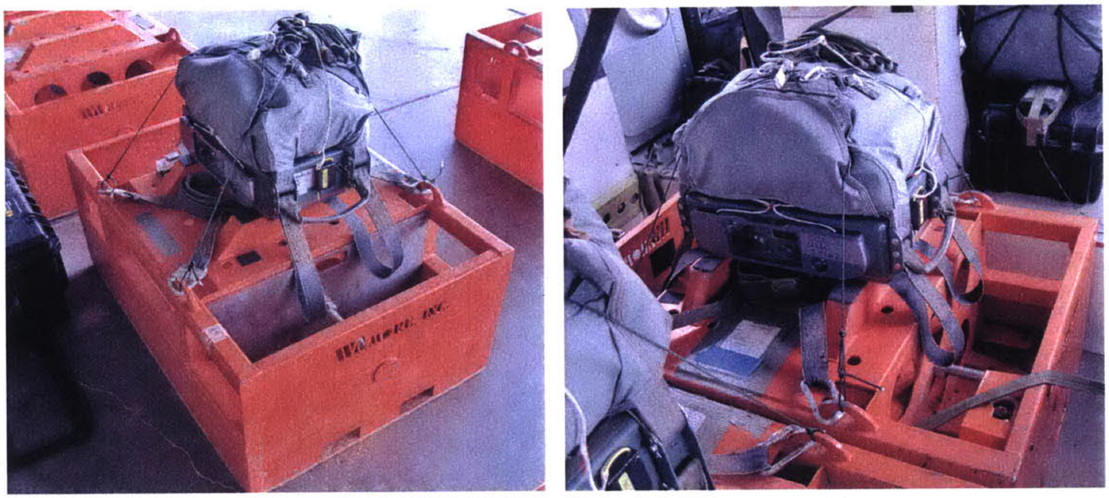

(a) (b)

Figure 6-2: Packed *UltraFly* parafoil and payload (author photo)

The payload used during testing consisted of a metal weight tub designed by Wamore Inc. as part of the *UltraFly* parafoil system [21]. This payload (shown by the orange box in Figure 6-2) is loaded with detachable steel plates so as to obtain the desired user-specified weight. For each of the 9 drop experiments presented in this chapter, the parafoil was rigged for a total system weight of 355 lbs, which includes the AGU and packed MC-5 ram-air canopy shown in Figure 6-2. The parafoil was deployed from a SC.7 Skyvan aircraft at altitudes between 2,469–3,124 meters above the target, and release distances ranging from 153–1,797 meters. Each flight test consisted of both a homing phase and an energy management phase coordinated by the Draper Laboratory GN&C software, followed by the initialization of CC-BLG terminal guidance. A final flare phase was also performed above the target altitude in order to slow the parafoil prior to impact. Figure 6-3 displays this flare maneuver for the deployed *UltraFly* parafoil system.

Flight testing of the CC-BLG algorithm was conducted outside of Eloy, Arizona over two 1-week test periods beginning January, 2015. The drop zone used for these experiments was a flat desert plain (shown in Figure 6-3) which enabled both reliable communications and the efficient recovery of the parafoil system. In order to test the robust guidance and obstacle avoidance features of the CC-BLG algorithm, simulated

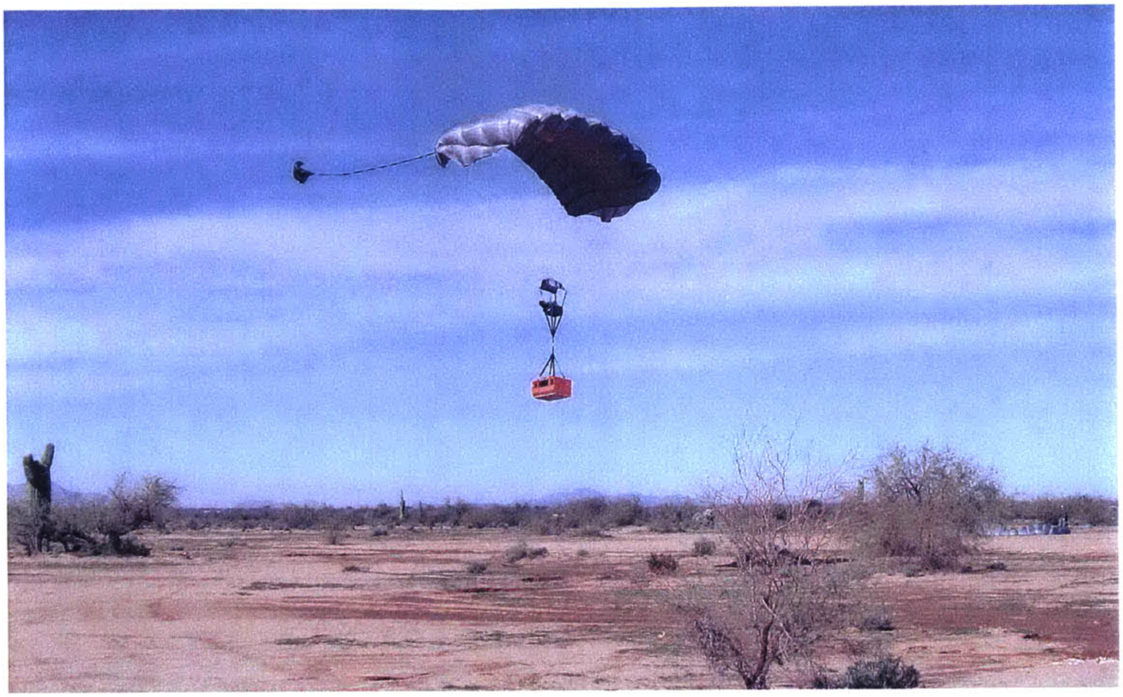


Figure 6-3: Deployed *UltraFly* parafoil system with MC-5 canopy during flare at Eloy test range (author photo)

terrain was loaded within the GN&C flight software using the DTED2 maps of the canyon-1 and canyon-2 environments presented in Chapter 5. To imitate parafoil terminal guidance within each of these simulated Grand Canyon locations, two new maps were created by overlaying the DTED2 data from the canyon-1 and canyon-2 environments onto the flat terrain map of the real-world drop zone. In this way, the target within each canyon-1 and canyon-2 map (denoted by the yellow circle in Figure 5-3 and Figure 5-4, respectively) was made to coincide with the true drop zone target location (i.e., latitude, longitude, and altitude), such that all simulated terrain was positioned relative to this point. For the flight test experiments presented in this chapter, the objective of the CC-BLG algorithm is therefore to design and execute a trajectory plan that guides the parafoil to the target location within the real-world drop zone, while avoiding terrain obstacles in the simulated software environment.

## 6.3 CC-BLG Flight Modifications

The available 180 MHz microcontroller (MCU) onboard the *UltraFly* AGU posed significant computational limitations for CC-BLG algorithm development when attempting to augment the existing Draper Laboratory flight software package. As a result, although the CC-BLG algorithm presented in Chapters 3–5 was shown to be computationally efficient relative to the computing power of standard modern day CPUs (as demonstrated in Table 4.3), adjustments to the algorithm had to be made in order to operate within the mandated 1 Hz GN&C software execution cycle using the MCU flight hardware. The largest amount of augmented computation for the CC-BLG algorithm exists during the process of generating covariance samples and checking for collisions against the terrain environment. Thus to reduce computational overhead, two primary changes were made to the CC-BLG formulation.

The first modification to the CC-BLG algorithm was to replace the method of dynamic propagation (2.32)–(2.36) used for generating the set of $N_u$ reachability trajectories, with the previous analytic approach developed in (2.63). This analytic expression was modified for use within the CC-BLG algorithm by incorporating the predicted displacement produced by the mean wind estimate $\overline{\mathbf{w}}$ over the propagation horizon $\tau$ according to

$$\begin{aligned}
p_{xi} &= p_{x0} + \left|\frac{v(p_{z0})}{\omega_i}\right| \Big(\cos(\psi_0 + \text{sign}(\omega_i)\frac{\pi}{2}) + \cos(\psi_0 + (2 + \text{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i\tau)\Big) \\
&\quad + \overline{\mathbf{w}}_x\tau, && (6.1) \\
p_{yi} &= p_{y0} + \left|\frac{v(p_{z0})}{\omega_i}\right| \Big(\sin(\psi_0 + \text{sign}(\omega_i)\frac{\pi}{2}) + \sin(\psi_0 + (2 + \text{sign}(\omega_i))\,\frac{\pi}{2} + \omega_i\tau)\Big) \\
&\quad + \overline{\mathbf{w}}_y\tau, && (6.2) \\
p_{zi} &= p_{z0} - \left(\frac{v(p_{z0})}{L_D} + \overline{\mathbf{w}}_z\right)\tau, && (6.3)
\end{aligned}$$

where $\tau$ is defined using (3.14), and the mean wind estimate is provided in (2.18).

To further reduce computation, reachability states and covariance samples are generated at only 2 discrete time points $\tau = \{2t_P/3, t_P\}$ for each of the $N_u$ trajectories, as opposed to every $\Delta t = 0.1$ seconds during the original dynamic propagation.

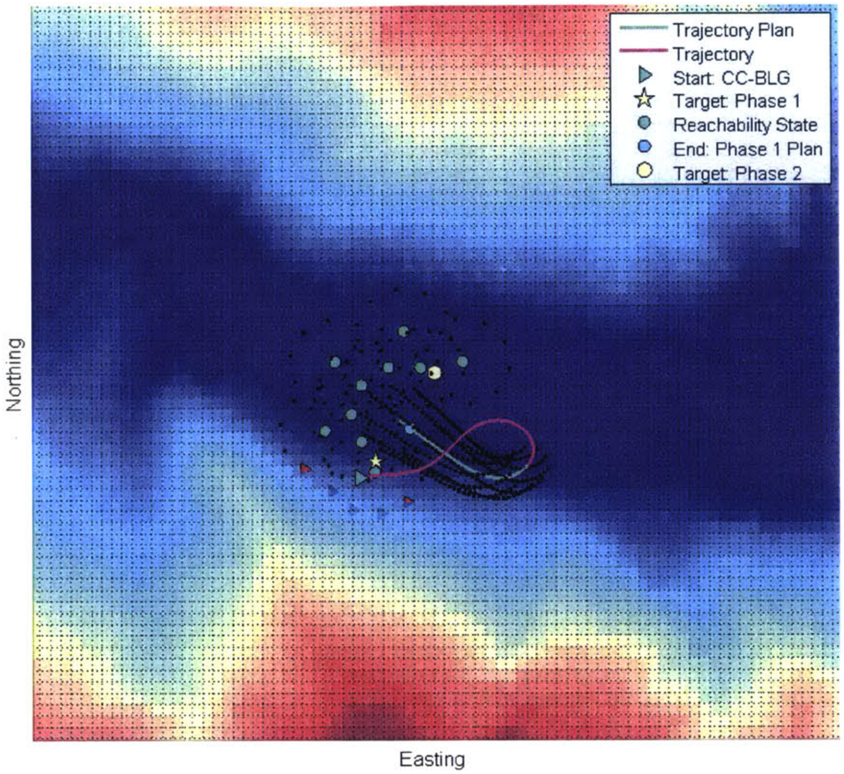


Figure 6-4: CC-BLG Flight modified reachability set approximation, $N_u = 5$

Additionally, the total number of reachability trajectories was reduced from $N_u = 7$ to $N_u = 5$. The result of these modifications is a significant reduction in the total number of covariance samples used for collision checking along the reachability horizon during each planning iteration. By neglecting the lag dynamics however, the ability of (6.1)–(6.3) to accurately predict the set of future parafoil states diminishes more quickly with increasing propagation time $\tau$. To compensate for this effect, the limits for the control were adjusted from $|u| \leq \omega_{max}$ to $|u| \leq 0.8\omega_{max}$ in (2.62) so as to produce wider turning circles, and offset the assumption of instantaneous heading rate response. These limits were selected manually by comparing the set of reachability states in (6.1)–(6.3) with those generated using (2.32)–(2.36) under the assumptions $\psi_0 = 0$ and $\dot{\psi}_0 = 0$, until the positions of the states approximately aligned. Although a more rigorous approach to tuning will be considered in future work, this approximation was found to work well in practice while significantly reducing the computation time required to generate each Phase 1 reachability set. The new reachability states (green circles) are shown in Figure 6-4 at times $t = \{t_F + 2t_P/3, t_F + t_P\}$ within the valley terrain environment.

The second set of modifications to the CC-BLG algorithm involved reducing both the number of optimization iterations per 1 Hz planning cycle, and the number of covariance samples $N_S$ generated around each trajectory state. Recall that a maximum limit of 50 Nelder-Mead optimization iterations was specified per planning cycle for the BLG and CC-BLG algorithms compared in Chapters 4–5. Throughout the simulation experiments conducted on a 3.2 GHz Intel i5-3470 CPU, this upper bound on iterations enabled both algorithms to satisfy the online computational requirements, such that the total time per cycle to converge upon a trajectory solution remained well below the 1 second threshold. It was also observed from simulation that the full set of 50 iterations was not typically required for optimization convergence after the initial planning cycles of Phase 1 and Phase 2 terminal guidance. Within the first several planning cycles, a larger number of optimization iterations may be required by the BLG and CC-BLG algorithms in order to refine the initial guess solution. However, by reinitializing each 1 Hz cycle using the previous best solution, the optimization was often found to converge more quickly, and in fewer iterations, during subsequent planning steps.

When comparing the performance of the CC-BLG algorithm on the 180 MHz MCU, the margin of computation available after the execution of the existing Draper GN&C software was insufficient for CC-BLG to remain within the 1 Hz planning cycle when a large number of optimization iterations were required. To reduce CC-BLG computation time, the maximum number of iterations allowable was decreased to 25 per cycle for the flight test version of the algorithm.

Figure 6-5 and Table 6.1 display the results from hardware-in-the-loop (HWIL) testing of CC-BLG on the 180 MHz MCU using Draper Laboratory's HWIL simulation facility [48]. In these experiments, a target threshold of 0.7 seconds was specified for the total computation time required to complete the GN&C cycle, including CC-BLG planning, so as to provide sufficient margin for the controller to successfully

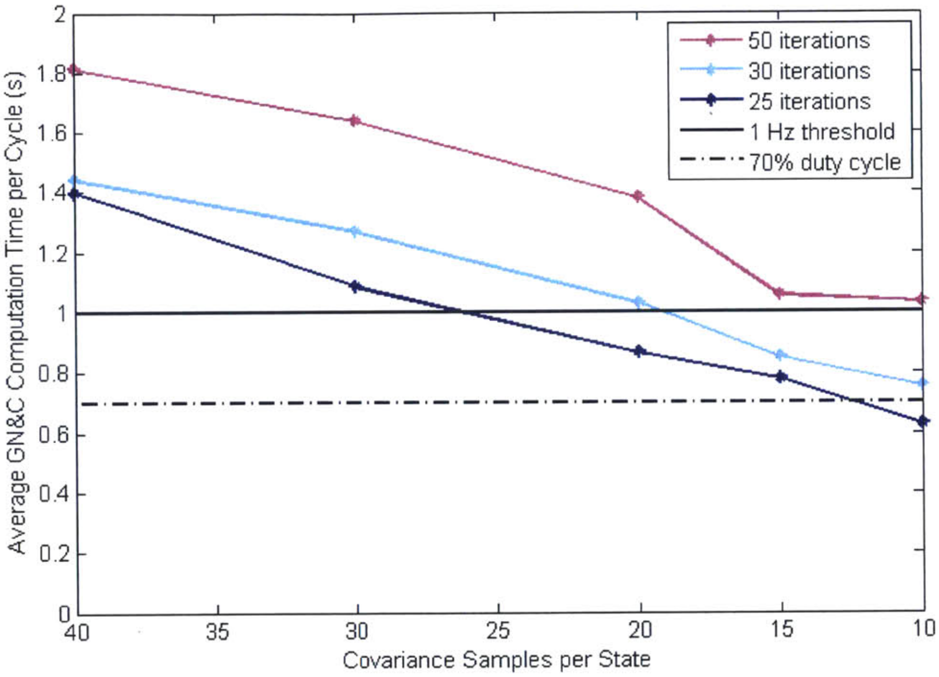


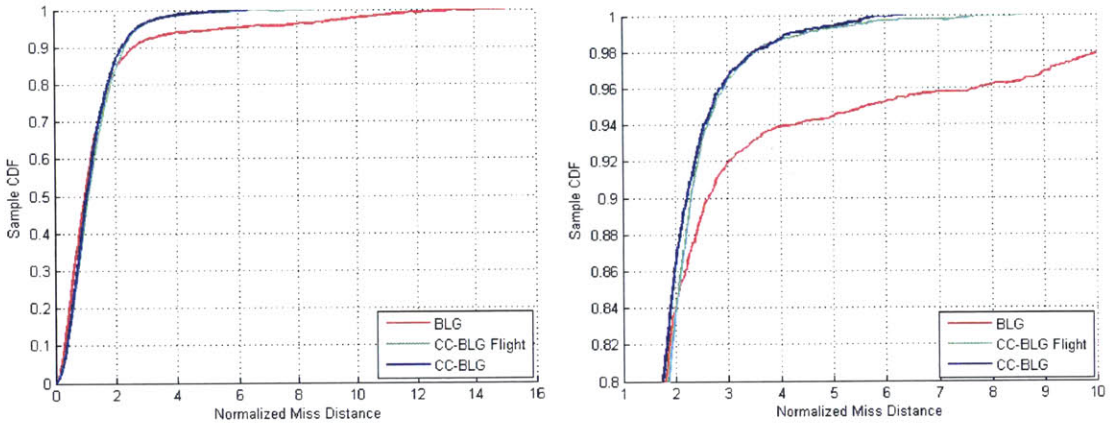
Figure 6-5: Average GN&C time per cycle for CC-BLG on 180 MHz microcontroller

Table 6.1: Computation time for CC-BLG on 180 MHz microcontroller (MCU)

| **Max Iterations** | **Samples per state** | **Average GN&C Time per cycle (s)** | **Average Time per state (ms)** |
|---|---|---|---|
| 25 | 10 | 0.628 | 0.473 |
| | 15 | 0.780 | 0.658 |
| | 20 | 0.868 | 0.842 |
| | 30 | 1.088 | 1.175 |
| | 40 | 1.399 | 1.545 |
| 30 | 10 | 0.755 | 0.485 |
| | 15 | 0.852 | 0.651 |
| | 20 | 1.032 | 0.844 |
| | 30 | 1.272 | 1.089 |
| | 40 | 1.443 | 1.538 |
| 50 | 10 | 1.033 | 0.472 |
| | 15 | 1.057 | 0.617 |
| | 20 | 1.382 | 0.836 |
| | 30 | 1.640 | 1.173 |
| | 40 | 1.814 | 1.526 |

execute commands at the designated 1 Hz frequency [32]. This represents a 70% GN&C duty cycle as indicated by the black dashed line in Figure 6-5. From the results shown in Figure 6-5 and Table 6.1, the average GN&C computation time of the CC-BLG flight software remains below the desired 0.7 second threshold when the total number of covariance samples per state $N_S$ is reduced to 10, and the maximum number of optimization iterations per cycle is set to 25. These modifications to the CC-BLG algorithm represent a significant cutback to the set of 40 samples per state and 50 maximum optimization iterations used in Chapters 4–5. The $N_S = 10$ covariance samples per state used for the flight test version of CC-BLG were positioned along a single covariance ellipse with $\sigma = \sigma_3 = 3.2$ (selected in Section 4.3), so as to provide sufficient warning to the planner about the location of environmental obstacles along the optimized parafoil trajectory. However, a potential side effect of this reduced arrangement of covariance samples is increased planning conservatism via the large probability weights $P(S) = 0.1$ assigned to the samples furthest away from the nominal trajectory state.

The normalized miss distance data of the modified flight test algorithm, referred to here as "CC-BLG Flight", is shown in Figure 6-6 and Table 6.2 over a series of 2500 Monte Carlo trials within the valley terrain environment using Draper Laboratory's high fidelity simulator, described in Section 4.2. The results produced by the CC-BLG Flight algorithm are compared alongside the BLG and CC-BLG simulation results reported in Table 5.1 and Figure 5-1, where the data from all three algorithms has once again been normalized using the BLG median miss distance for this terrain scenario. Notice in Figure 6-6 that CC-BLG Flight produces a miss distance CDF that is very similar to the original CC-BLG algorithm, while also demonstrating improved landing accuracy over BLG in the upper 15% of the distribution. With fewer covariance samples, and a smaller number of optimization iterations allowable per planning cycle, the CC-BLG Flight algorithm produces a slightly longer distribution tail than the CC-BLG algorithm due to the addition of several worst-case outliers. Additionally, the median miss distance of CC-BLG Flight is 12% larger than BLG in this terrain, whereas the median miss distance of the full CC-BLG algorithm is only



(a) full CDF

(b) zoomed in to $(1, 10) \times (0.8, 1)$

Figure 6-6: Normalized miss distance CDF for valley terrain comparison of CC-BLG Flight, 2500 trials

Table 6.2: Normalized miss distance data for valley terrain comparison of CC-BLG Flight, 2500 trials

| **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|
| BLG | 1.56 | 2.10 | 1.00 | 1.81 | 2.64 | 5.62 | 10.18 | 15.04 |
| CC-BLG Flight | 1.29 | 0.89 | 1.12 | 1.86 | 2.31 | 2.76 | 3.55 | 8.62 |
| CC-BLG | 1.23 | 0.83 | 1.05 | 1.76 | 2.23 | 2.71 | 3.53 | 6.34 |

5% larger, as shown in Table 6.2. However, the overall similarities between the performance of the CC-BLG and CC-BLG Flight algorithms in Figure 6-6 and Table 6.2 demonstrate that the robustness characteristics produced by the analytic chance constraints have been largely preserved, despite the modifications to reduce computation. The results of this experiment also highlight the effectiveness of the covariance samples and reachability set augmentations to the original BLG algorithm, such that as these features are removed, the performance of CC-BLG Flight is slightly diminished relative to the full CC-BLG algorithm. Throughout the following discussion of the 9 flight test experiments conducted with the CC-BLG Flight algorithm in Section 6.4, "CC-BLG Flight" will once again be referred to simply as "CC-BLG" for the purpose of convenience.

## 6.4 Flight Results

Table 6.3 summarizes the results from 9 CC-BLG flight test experiments performed during the months of January and March, 2015. The test procedure and CC-BLG flight software used for each drop experiment were preselected prior to both one week test periods, such that real-time modifications could not be made based on the observed flight results. The experiments were divided between drops using the canyon-1 and canyon-2 scenarios in order to evaluate CC-BLG robustness subject to challenging wind and terrain interaction. Throughout all 9 flight tests, the CC-BLG algorithm successfully guided the *UltraFly* parafoil to the target location while avoiding collisions within the simulated terrain environment. Figure 6-7 displays the landed *UltraFly* system at the Eloy drop zone after the first successful test flight of the CC-BLG parafoil terminal guidance algorithm.

Figures 6-8(a)–(b) display the impact points produced by each flight test experiment in the simulated canyon-1 and canyon-2 terrains. From the locations of these impact points (denoted with a red 'x') it can be observed that all 9 drops of CC-BLG resulted in a safe parafoil landing along the region of dark blue elevation at the bottom of each canyon environment. The impact points in Figure 6-8(b) demonstrate the effect of the constrained geometry produced by the walls of the canyon-2 terrain. These impact points are positioned along a line running from north to south, corresponding to the unconstrained direction of the parafoil final approach. Figures 6-8(a)-(b)

Table 6.3: Summary of CC-BLG flight test results

| **Test** | **Date** | **Terrain** | **Release Alt. (ft. MSL)** | **Release Distance (m)** | **Miss Distance (m)** |
|---|---|---|---|---|---|
| 1 | 1/6/2015 | canyon-1 | 11,673.1 | 877.0 | 35.9 |
| 2 | 1/7/2015 | canyon-2 | 11,523.5 | 719.1 | 140.3 |
| 3 | 1/7/2015 | canyon-2 | 11,535.9 | 945.5 | 86.3 |
| 4 | 1/8/2015 | canyon-1 | 10,515.6 | 973.0 | 70.9 |
| 5 | 1/8/2015 | canyon-2 | 9,638.6 | 582.6 | 80.5 |
| 6 | 3/10/2015 | canyon-1 | 11,699.3 | 153.2 | 31.9 |
| 7 | 3/11/2015 | canyon-2 | 11,811.2 | 148.4 | 26.6 |
| 8 | 3/11/2015 | canyon-1 | 11,623.2 | 318.7 | 20.4 |
| 9 | 3/12/2015 | canyon-2 | 11,752.5 | 1797.1 | 14.1 |

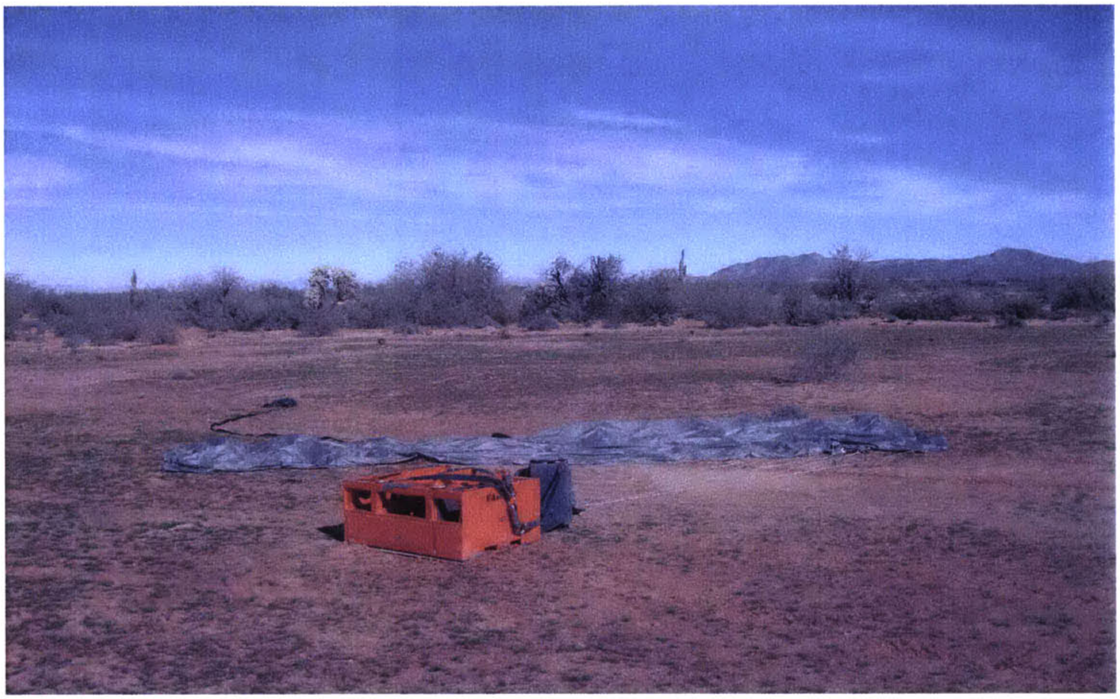
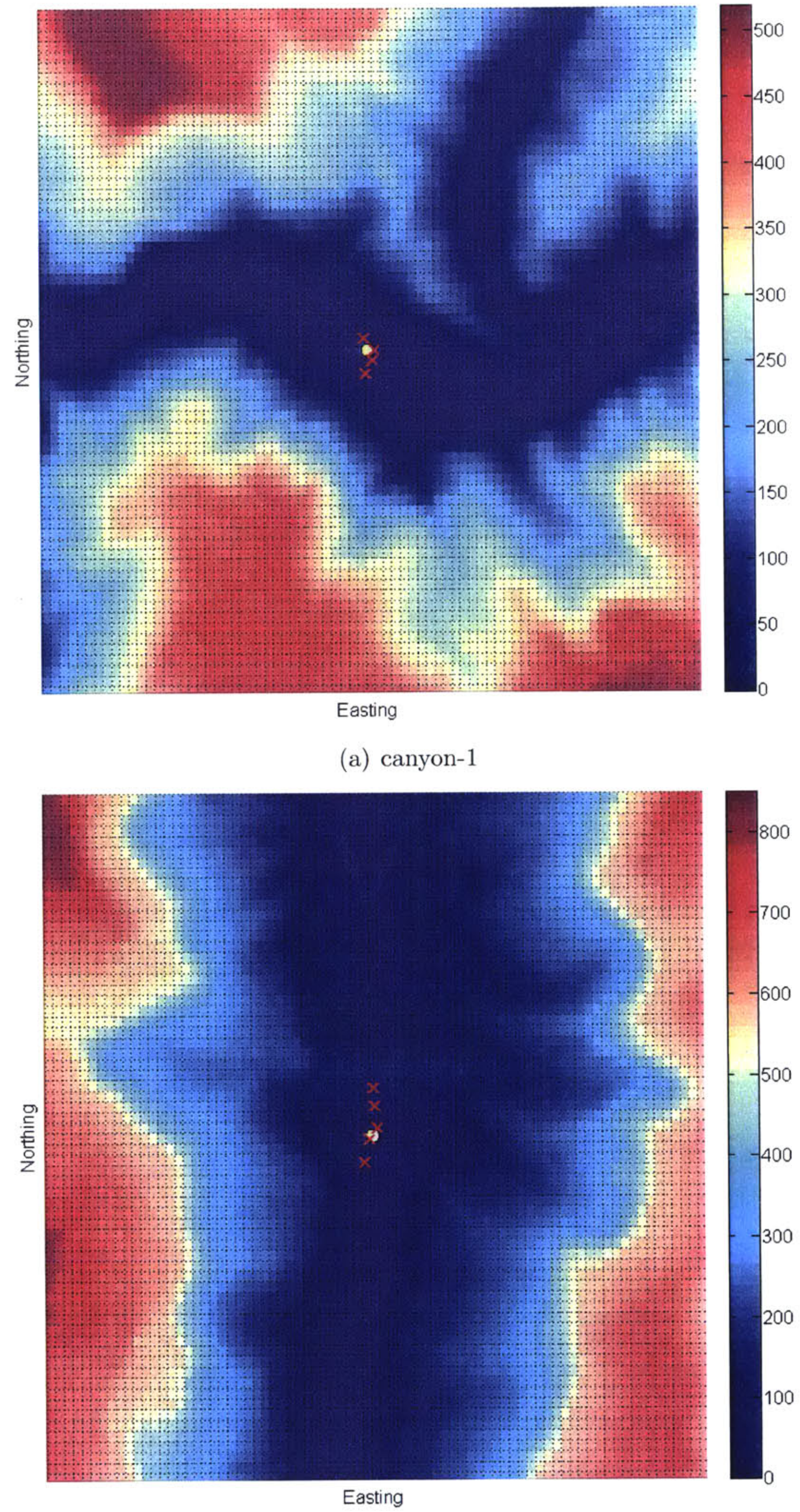


Figure 6-7: Landed parafoil at Eloy drop zone after CC-BLG Test 1 (author photo)

reveal that the landing locations produced by the CC-BLG algorithm are grouped in the vicinity of the target for each terrain environment, with the largest miss distance shown in Figure 6-8(b) by the northernmost impact point produced during Test 2. Significant GPS altitude bias was found to be present in the Test 2 experiment, causing the parafoil to land short of the planned terminal trajectory state. Regardless of this effect, CC-BLG maintained safety and landed the parafoil in the obstacle free region at the bottom of the simulated canyon-2 environment.

### Test 4: Analysis

Figure 6-9 displays the GPS ground track of the parafoil produced during Test 4, beginning with the energy management phase. For clarity, the Phase 1 and Phase 2 terminal guidance portions of the CC-BLG trajectory are also plotted in Figure 6-10(a)-(b) within the simulated canyon-1 environment. Figure 6-10 illustrates that the Test 4 parafoil trajectory produced by CC-BLG remains robust to possible wind uncertainty while successfully maneuvering in the vicinity of the terrain. In particular, Figure 6-10(b) reveals that the upwind Phase 1 target for this experiment (shown by



(a) canyon-1

(b) canyon-2

Figure 6-8: Flight test impact points within the canyon-1 and canyon-2 terrain

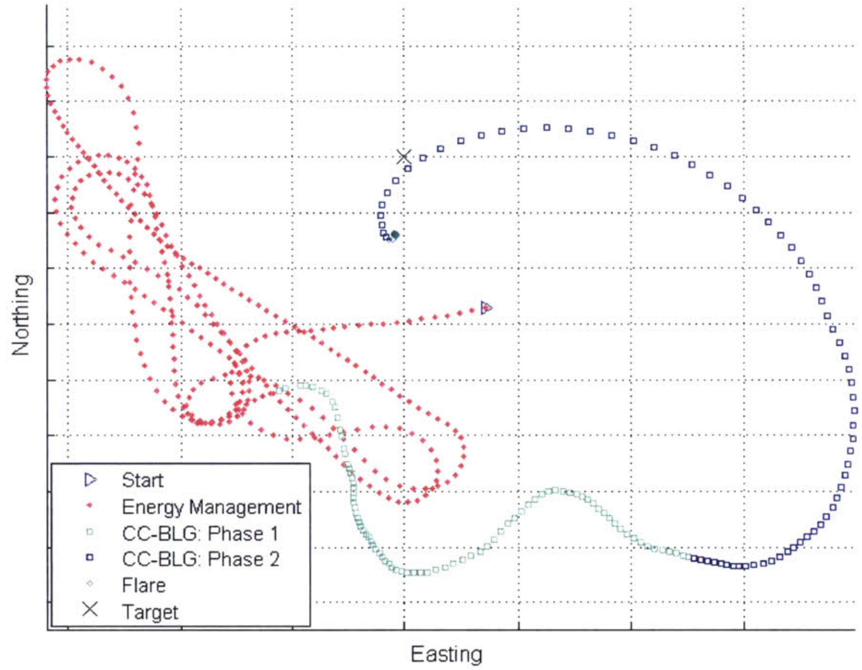
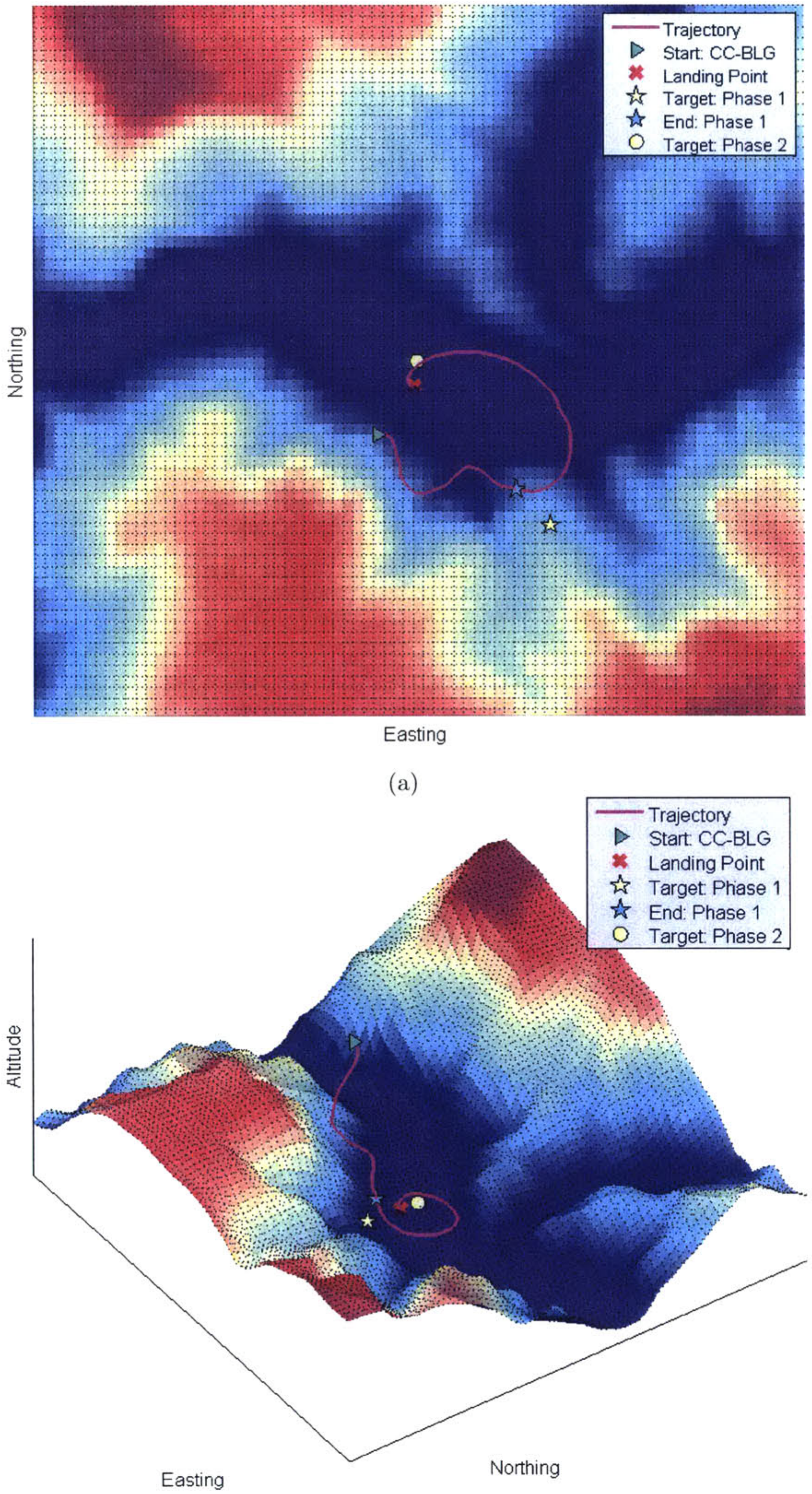


Figure 6-9: CC-BLG GPS ground track, Test 4

the yellow star) is located very close to the southern wall of the canyon-1 environment, creating a potentially dangerous planning scenario. With the help of the reachability set approximation, CC-BLG completes Phase 1 guidance with the parafoil positioned at a slight offset (blue star) and oriented parallel to the nearest terrain obstacle, as shown in Figure 6-10(b). This position and orientation increase the feasible planning space during the terminal guidance phase transition.

Figure 6-11 displays the wind measurements recorded onboard the *UltraFly* AGU from the Test 4 flight experiment. In this test, the west component of the wind increased by over 13 ft/s (7.7 knots) within the final 500 ft of descent above the target altitude. This caused the parafoil to accelerate rapidly towards the end of the Phase 2 trajectory segment (shown in Figures 6-9 and 6-10), while traveling downwind to the target location. Due to the wind uncertainty and environmental constraints, CC-BLG determined that it was unsafe to guide the parafoil further north in order to dissipate altitude before reaching the target. From the trajectory shown in Figure 6-10, this resulted in a slight target overshoot during the final turn maneuver into the wind. Figure 6-12 depicts the *UltraFly* parafoil system at the end of Test 4 after



(a)

(b)

Figure 6-10: CC-BLG trajectory in the canyon-1 terrain, Test 4

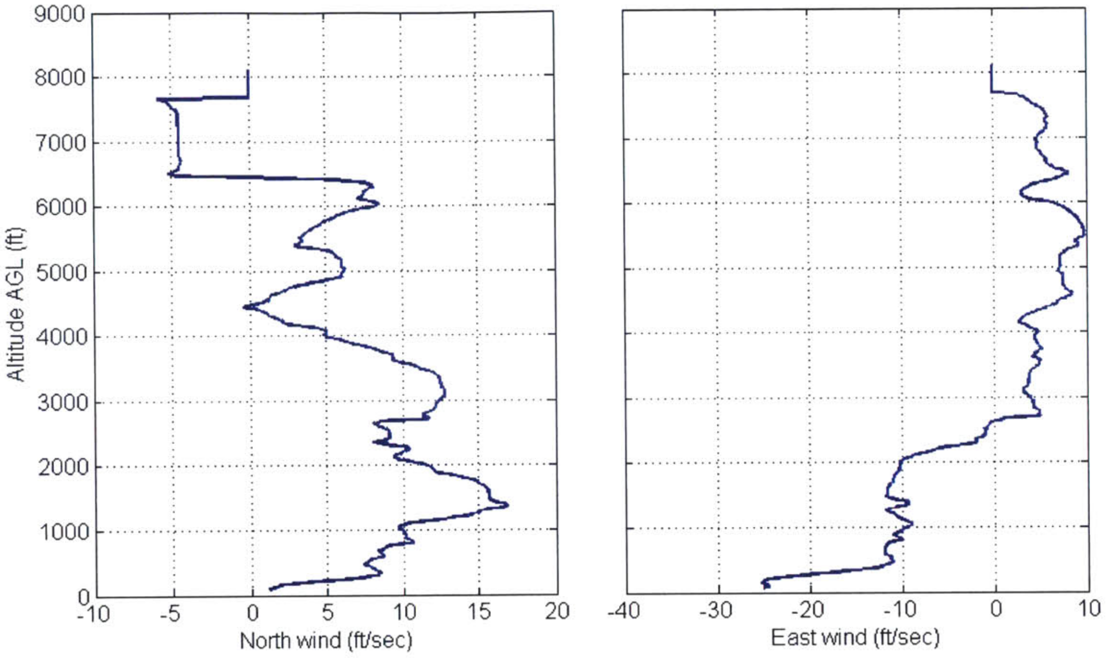


Figure 6-11: Onboard AGU wind measurements during parafoil descent, Test 4

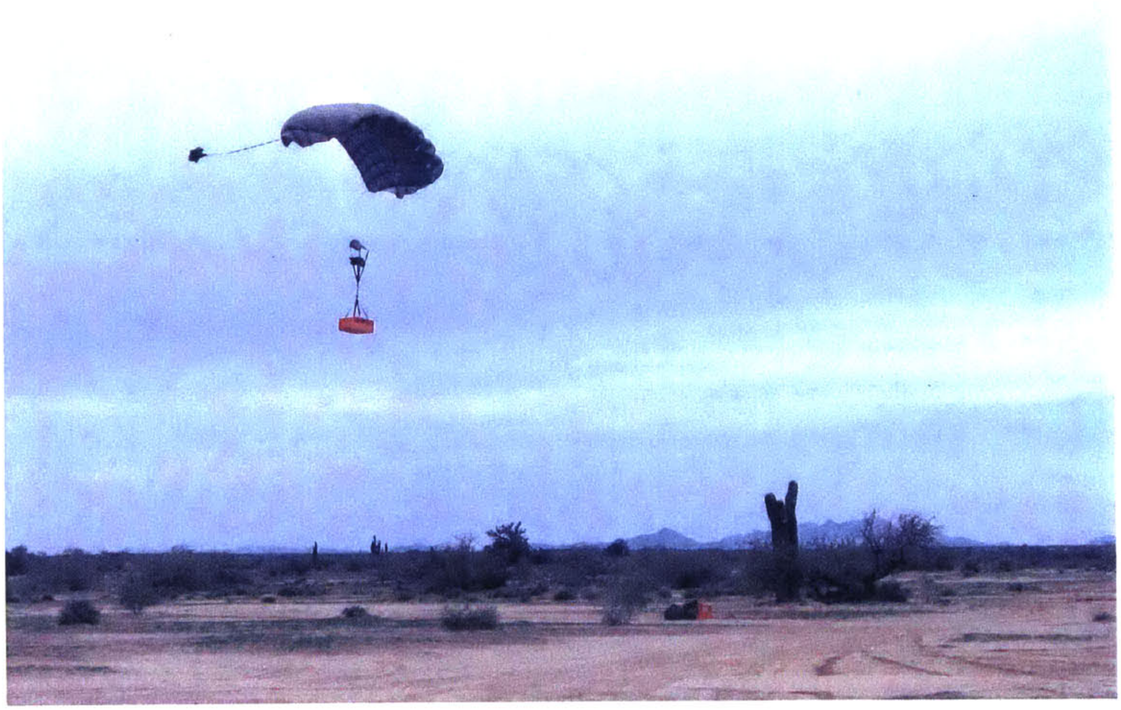


Figure 6-12: Parafoil system on final approach during CC-BLG Test 4 (author photo)

completing this final turn maneuver prior to flare initialization. Of the 4 experiments using the simulated canyon-1 terrain, this test produced the largest miss distance of 70.9 meters.

### Test 9: Analysis

Figures 6-13 and 6-14 provide an example of CC-BLG trajectory planning in the canyon-2 terrain using the Test 9 flight results. It can be observed in Figure 6-13 that the direction of the wind changed from north to south during the Test 9 energy management phase (red markers), causing the GN&C software to reposition the parafoil upwind of the target (black 'x') prior to the initialization of Phase 1 terminal guidance. The wind measurements recorded onboard the AGU are shown in Figure 6-15. In this example, the Phase 1 target point (yellow star) in Figure 6-14 is far enough away from the terrain that CC-BLG determines the parafoil can safely approach this upwind point at the start of Phase 2 guidance (blue star). Throughout Phase 2, CC-BLG designs and executes a trajectory plan that guides the parafoil toward the middle of the canyon so as to perform a final turn maneuver into the wind. As compared to Figure 6-11, the winds during this test displayed less variability throughout the final stages of CC-BLG terminal guidance, which helped to facilitate an accurate landing at the target location.

### Test 8: CC-BLG Trajectory Planning

Figure 6-16 displays the process of CC-BLG robust parafoil guidance over a series of four time points using the data recorded from Test 8 in the simulated canyon-1 terrain environment. The CC-BLG trajectory plan (green line) is shown in Figure 6-16(a) roughly three quarters of the way through Phase 1 terminal guidance, including the reachability states (green circles) and covariances samples (black dots). Figure 6-16(a) also displays collided covariance samples (red triangles) around several of the reachability states which incur a penalty within the $J_{CTG}$ component of the CC-BLG objective function (3.16). In this flight test experiment, the desired upwind target point (yellow star) for Phase 1 guidance was once again positioned in close proximity to the terrain due to the narrow canyon geometry constraints. As a result,

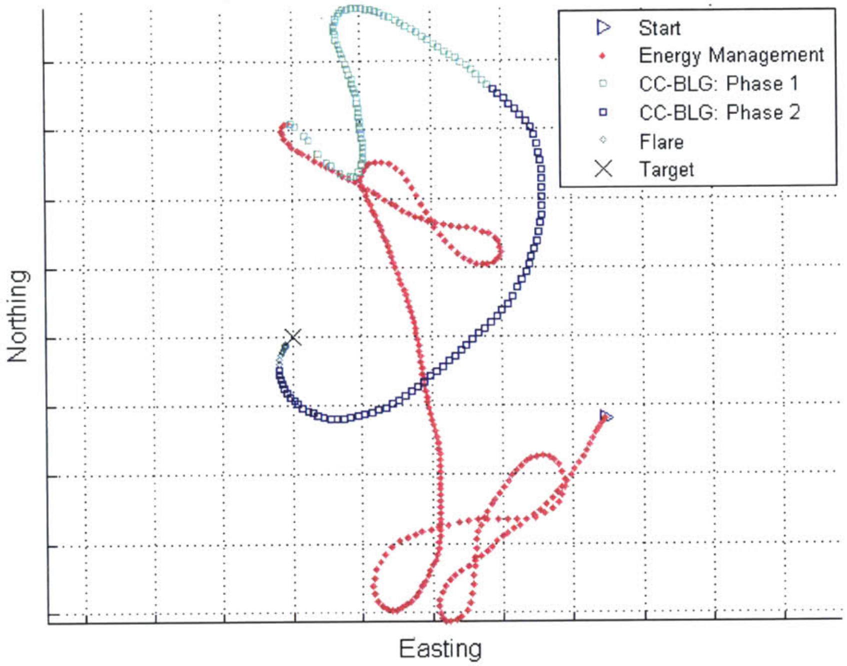
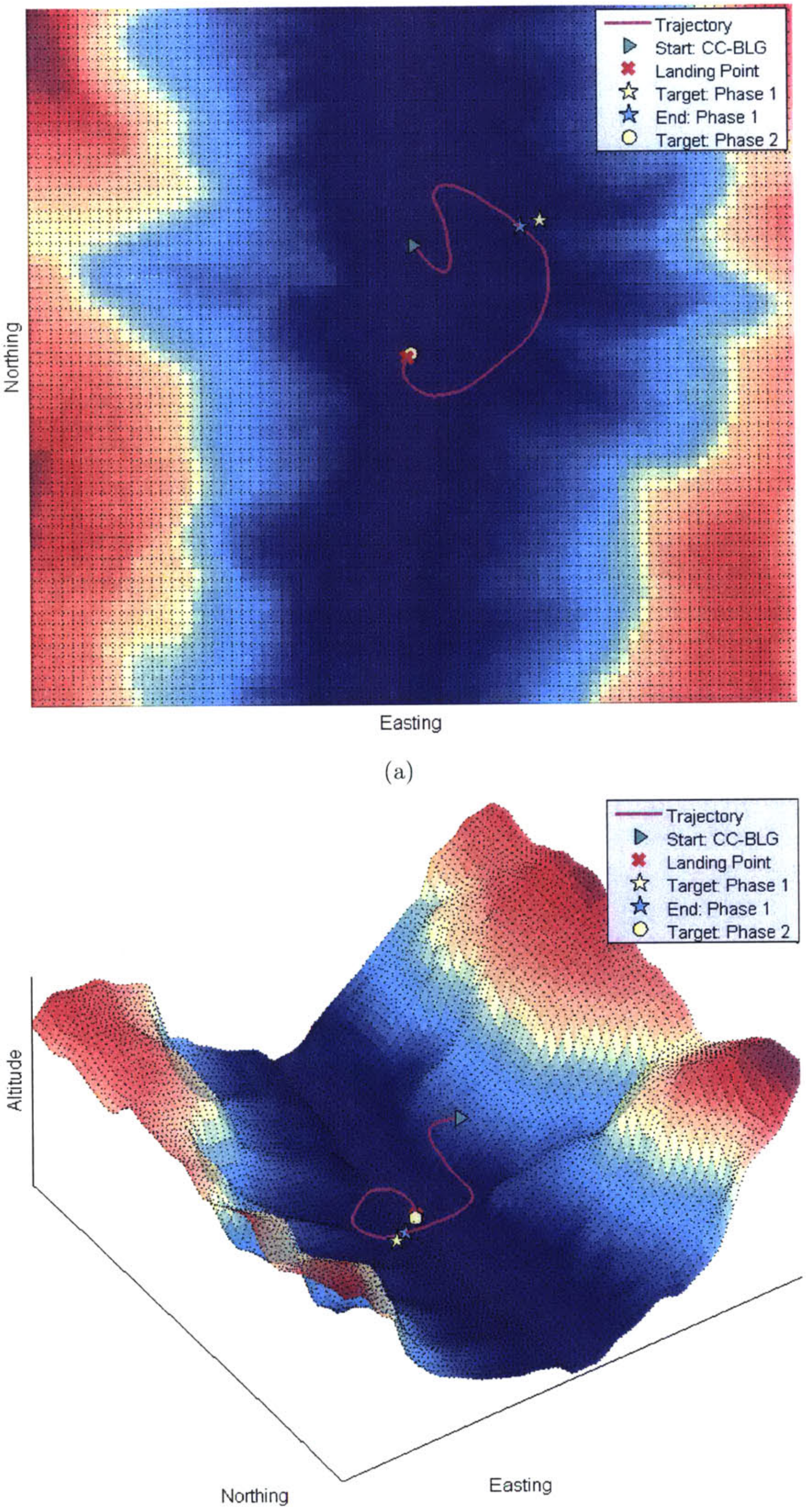


Figure 6-13: CC-BLG GPS ground track, Test 9

the reachability states in Figure 6-16(a) were used by CC-BLG to maintain a safe distance between the parafoil and the terrain, while orienting the parafoil's Phase 1 target approach to be parallel to the east-west contour of the canyon. From the set of risk-free reachability states shown in Figure 6-16(a), it can also be observed that a safe trajectory plan exists over the middle of the canyon if the parafoil turns to the right following the initialization of Phase 2 guidance.

The parafoil trajectory (magenta line) and CC-BLG plan (green line) are shown in Figure 6-16(b) after the first 3 seconds of Phase 2 terminal guidance. At this point in the flight, Figure 6-16(b) reveals that CC-BLG has successfully identified a trajectory plan to the final target location (yellow circle). As predicted from the reachability states in Figure 6-16(a), the Phase 2 CC-BLG plan turns to the right and avoids a collision with the terrain environment. However, because of the narrow canyon-1 bottleneck in the middle of the map, this trajectory plan causes the parafoil to pass within close proximity to the terrain along the path to the target location. In Figure 6-16(b), several collided covariance samples (red dots) are shown intersecting the canyon-1 terrain to the north and south, indicating regions of increased



(a)

(b)

Figure 6-14: CC-BLG trajectory in the canyon-2 terrain, Test 9

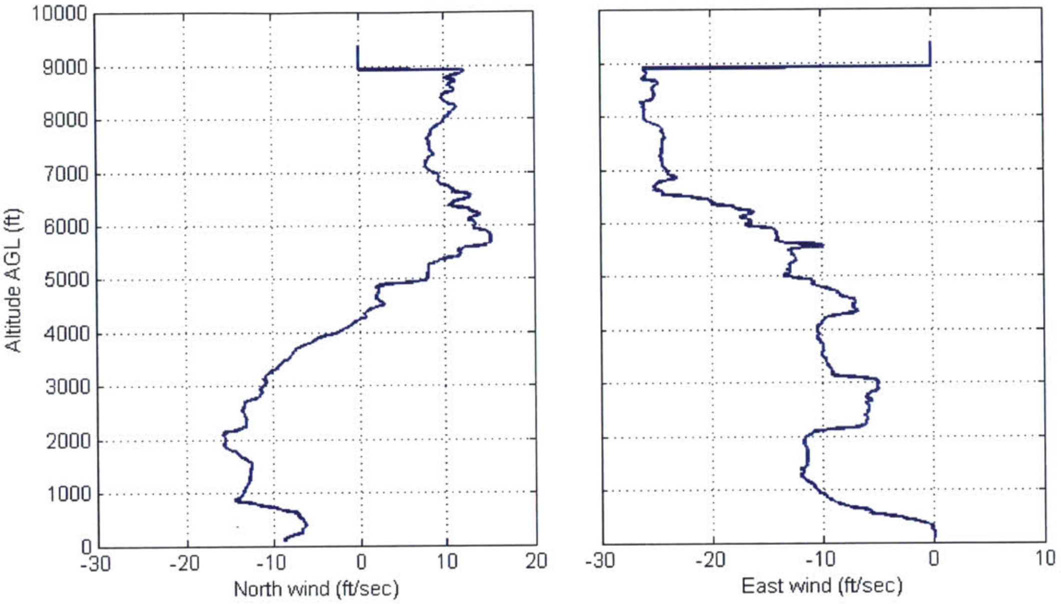
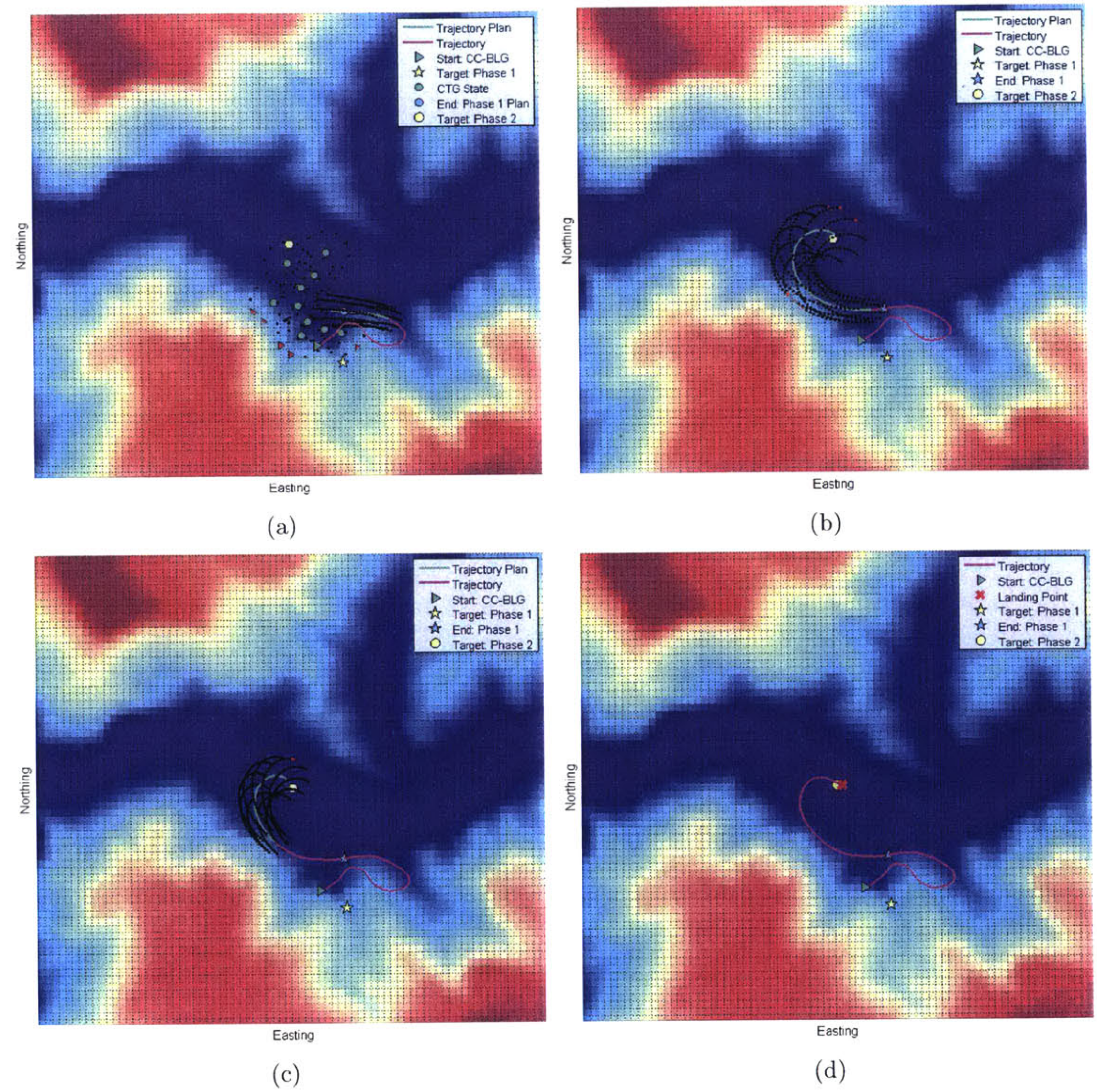


Figure 6-15: Onboard AGU wind measurements during parafoil descent, Test 9

risk. During this flight test, the CC-BLG optimization determined that the level of environmental risk was acceptable in order to minimize the landing distance from the target location. Figure 6-16(b) illustrates that with the information provided by the covariance samples, CC-BLG is able to generate an appropriate buffer of terrain separation for the nominal trajectory plan as a means to safeguard against possible wind disturbances throughout the remainder of the descent.

Figure 6-16(c) displays the Test 8 parafoil trajectory approximately halfway through the process of Phase 2 terminal guidance. Here it can be seen that as the parafoil passes alongside the southern canyon wall, CC-BLG has successfully minimized the level of environmental risk by eliminating all of the collided covariance samples within this region. After safely guiding the parafoil into the middle of the canyon, the final segment of the CC-BLG Phase 2 trajectory is shown in Figure 6-16(d), resulting in an upwind landing near the target location. The flight test data presented in these images demonstrates the robustness capabilities of CC-BLG in complex terrain environments, and reveals that the modified flight algorithm behaved largely as expected given the set of original CC-BLG simulation experiments conducted in Chapters 4–5.



(a) (b) (c) (d)

Figure 6-16: CC-BLG robust trajectory planning in canyon-1 terrain, Test 8

Figure 6-17 displays the time history for each of the three cost components of the CC-BLG risk-based objective function (3.16) throughout the Test 8 flight experiment shown in Figure 6-16. After a short transient during the initialization of Phase 1 terminal guidance, the most significant cost trade-off is shown to occur between the $J_{BLG}$ and $J_{CTG}$ objectives, as the optimizer attempts to minimize the distance from the upwind Phase 1 target. As demonstrated in Figure 6-16, this upwind target is positioned close to the canyon-1 terrain, causing the reachability states to incur an

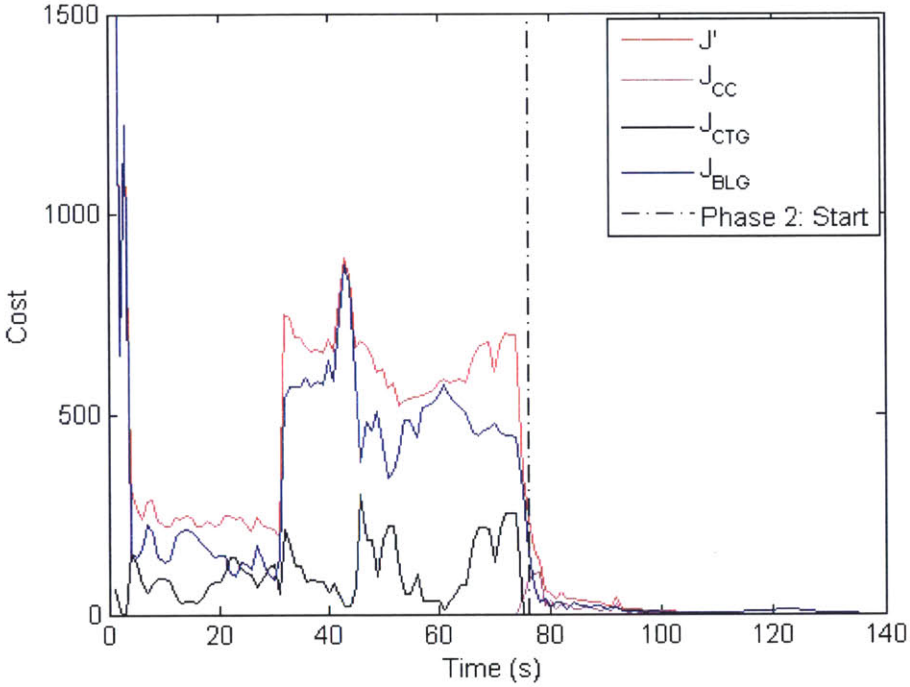


Figure 6-17: CC-BLG cost components during terminal guidance, Test 8

additional cost penalty by detecting the presence of environmental obstacles. Over the CC-BLG time period from 30 to 75 seconds, the total cost $J'$ produced by these competing objectives is shown to be in relative equilibrium. This suggests that the optimizer has converged around a parafoil approach orientation that simultaneously balances the level of risk with the distance from Phase 1 target location. After the transition to Phase 2 terminal guidance, CC-BLG quickly minimizes the total cost of the remaining $J_{CC}$ and $J_{BLG}$ objectives by converging to a robust trajectory solution that terminates at the final target location. As indicated in Figure 6-17, the $J_{CC}$ component for the risk incurred along the planned CC-BLG trajectory is visible for only a brief period during the start of Phase 2 guidance. This cost is incurred by a small number of collided covariance samples, as shown in Figure 6-16(b)-(c). The results from the Test 8 flight experiment indicate that CC-BLG is effective at minimizing the total level of environmental risk, such that the planned parafoil trajectory remains robust to future wind uncertainty.

## 6.5 Post-flight Simulation Analysis

This section provides a statistical comparison between the flight tested version of the CC-BLG algorithm, referred to hereafter as "CC-BLG Flight", and the nominal BLG algorithm using Draper Laboratory's high fidelity simulation environment [48]. Due to the limited resources available to extensively flight test both algorithms under identical drop conditions, this simulation environment is an effective tool for describing how both algorithms would have performed over repeated experiments by recreating the environmental conditions from each of the 9 flight tests described in Table 6.3. This was achieved by generating an altitude-dependent wind profile from each flight experiment using the measurement logs stored onboard the *UltraFly* AGU. Two of these wind profiles have been shown in Figure 6-11 and Figure 6-15 from the data recorded during Test 4 and Test 9, respectively.

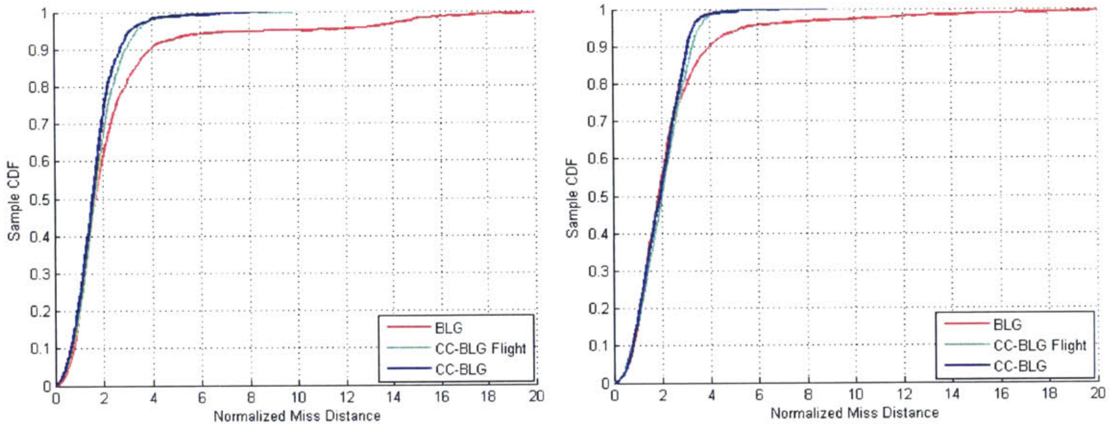
The wind profiles produced from each of the 9 flight test experiments were incorporated into the Draper Laboratory simulation as the "truth" wind acting on the nonlinear parafoil model [73]. Additionally, the adjusted map tiles of the canyon-1 and canyon-2 terrain environments used throughout flight testing (described in Section 6.2) were also implemented within the following Monte Carlo experiments in order to simulate the parafoil over the identical altitude range relative to the Eloy drop zone. Because the wind measurements observed by the GN&C software are instrumental in selecting both the upwind location for energy management, and the target point for Phase 1 terminal guidance, the wind-terrain interaction produced during each of the 9 flight tests within the canyon-1 and canyon-2 environments could be effectively recreated in simulation using the recorded wind data. This enabled the parafoil to be placed in a similar location at the initialization of Phase 1 terminal guidance. The Monte Carlo parameters for this set of simulation experiments were also adjusted to mimic both the flight test characteristics of the *UltraFly* parafoil, as well as the release distances and altitudes shown in Table 6.3. To reflect possible variations in the rigged weight of 355 lbs used during testing, the parafoil system weight was uniformly sampled in the range from 350 to 360 lbs for each simulation.

Likewise, the parafoil was simulated from release altitudes uniformly sampled over the range from 2,591 meters (8,500 ft) to 3,200 meters (10,500 ft) above the target, and release distances ranging from 100 to 2,000 meters. The sampling distributions for the remaining Monte Carlo variables of lift-to-drag ratio and turn rate bias were left the same as in Chapters 4-5 in order to represent imperfect knowledge of the parafoil system about the nominal canopy values.

From the analysis of the wind data recorded at Eloy, the set of 5 wind profiles from the January flight test displayed stronger average winds and higher variability than the 4 wind profiles from March—particularly in the final 2,000 ft above the target altitude. Within this altitude range used for parafoil terminal guidance, the wind profiles from the 5 tests in January had an average overall intensity of 14.78 ft/s (4.51 m/s), as compared to 9.91 ft/s (3.02 m/s) for the 4 tests in March. Although higher intensity wind gusts above 17.5 ft/s (5.33 m/s) were also more prevalent during the January flight tests, both months saw parafoil drops in which the wind gusts exceeded 24 ft/s (7.32 m/s) during the terminal guidance phase. In order to capture the differences between the winds observed from each test period, the following Monte Carlo experiments are divided into two simulation sets. The first set includes the 5 wind profiles from January, while the second set includes the 4 wind profiles from March. These sets of experiments will be referred to as "January-FT" for January flight test, and "March-FT" for March flight test, respectively.

### January-FT Simulation Analysis

Figure 6-18 and Table 6.4 display the results from 1500 Monte Carlo simulation trials of the BLG and CC-BLG Flight terminal guidance algorithms in both the canyon-1 and canyon-2 terrain environments. These tests were conducted with the January-FT simulation set, representing 300 Monte Carlo trials using each of the 5 wind profiles. For comparison, the results produced by the original CC-BLG algorithm with 40 covariance samples per state, and a maximum of 50 optimization iterations per planning cycle, are also shown in Table 6.4 and Figure 6-18 over an identical set of 1500 Monte Carlo trials in each terrain environment. Following the



(a) canyon-1

(b) canyon-2

Figure 6-18: Normalized miss distance CDF for January-FT simulation, 1500 trials

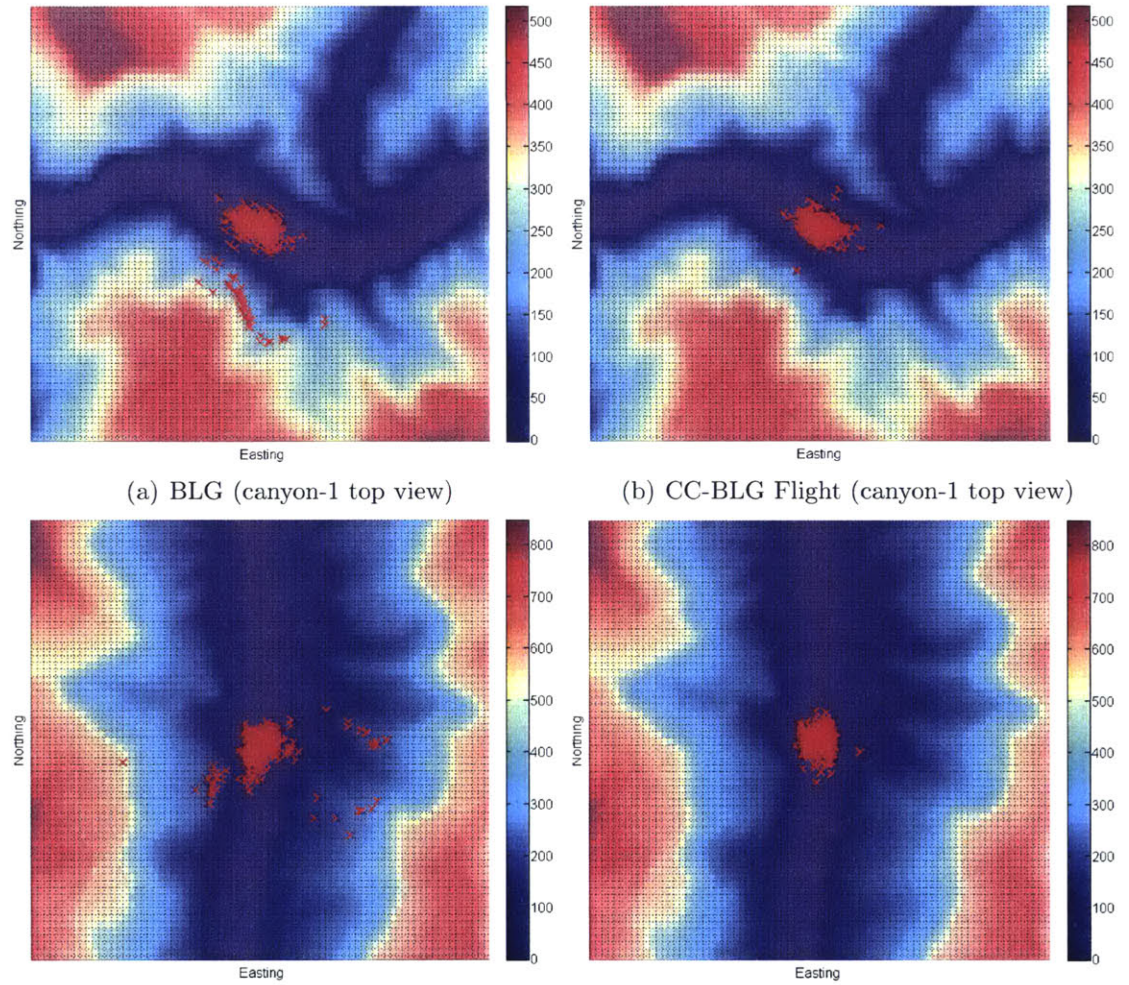
Table 6.4: Normalized miss distance data for January-FT simulation, 1500 trials

| **Terrain** | **Algorithm** | **Mean** | **StDev** | **50%** | **80%** | **90%** | **95%** | **98%** | **Max** |
|---|---|---|---|---|---|---|---|---|---|
| canyon-1 | BLG | 2.57 | 3.23 | 1.68 | 2.92 | 3.99 | 10.78 | 15.00 | 21.45 |
| | CC-BLG Flight | 1.79 | 1.00 | 1.66 | 2.41 | 2.93 | 3.42 | 4.04 | 10.31 |
| | CC-BLG | 1.68 | 0.94 | 1.58 | 2.20 | 2.73 | 3.13 | 3.94 | 8.75 |
| canyon-2 | BLG | 2.48 | 2.70 | 1.88 | 3.06 | 4.03 | 5.44 | 12.12 | 24.07 |
| | CC-BLG Flight | 2.04 | 0.98 | 2.01 | 2.90 | 3.28 | 3.59 | 3.95 | 7.57 |
| | CC-BLG | 1.94 | 0.94 | 1.93 | 2.77 | 3.08 | 3.30 | 3.72 | 8.91 |

procedure outlined in Chapters 4–5, these simulations were performed on a 3.2 GHz Intel i5-3470 CPU, such that computational limits were no longer a factor for the full CC-BLG algorithm. In this section, the miss distance data from all experiments is normalized using the median miss distance of the BLG algorithm on the *flat terrain* scenario recorded in Section 5.4. This is selected so as to compare the simulation performance of all 3 algorithms against the median BLG landing accuracy achieved in an obstacle free environment.

From the CDF plots in Figures 6-18(a) and 6-18(b), it can be observed that when the January-FT conditions are used for simulation, the CC-BLG and CC-BLG Flight algorithms provide improved landing accuracy over BLG throughout the upper tail of the distribution in both terrain environments. This improvement is evident for all trials above the 50th percentile in the canyon-1 scenario, and for all trials above the 78th percentile in the canyon-2 scenario. These results demonstrate that both variants of CC-BLG successfully reduce the number of worst-case collisions caused by the BLG algorithm, thereby eliminating a significant portion of large miss distances from each distribution. The statistics in Table 6.4 reveal that the average landing accuracy of the CC-BLG Flight algorithm is improved relative to BLG by more than 30% in the canyon-1 terrain, and by more than 17.5% in the canyon-2 terrain, while the worst-case miss distances are also reduced by more than a factor of 2 and 3, respectively. While the full CC-BLG algorithm produces slightly better accuracy than CC-BLG Flight across nearly all percentiles, the shapes of the distributions for the two CC-BLG variants in Figures 6-18(a)-(b) are very similar in both terrain scenarios. This indicates that even with only half the number of allowable optimization iterations and a quarter the number of covariance samples as the original algorithm, CC-BLG Flight is capable of providing improved terminal guidance performance over the state-of-the-art BLG algorithm in complex terrain environments due to the superior robustness properties of the analytic chance constraints.

The impact points produced by the BLG and CC-BLG Flight algorithms during each set of 1500 January-FT Monte Carlo trials are shown in Figure 6-19 for each terrain scenario. Note from Figures 6-19(b) and 6-19(d) that nearly all of the simulation trials of the CC-BLG Flight algorithm result in safe landings in the vicinity of the target at the bottom of the canyon-1 and canyon-2 environments. In contrast, the BLG algorithm in Figures 6-19(a) and 6-19(c) produces numerous terrain collisions during each set of simulations using the January-FT wind conditions. As illustrated in Figure 6-10, the direction of the wind during several of the January flight tests caused the Phase 1 target point to be placed in close proximity to the canyon-1 terrain in order to keep the parafoil upwind prior to the initialization of Phase 2 terminal guidance. This results in a particularly high concentration of BLG terrain collisions along the south wall of the canyon-1 terrain (shown in Figure 6-19(a)), suggesting the BLG algorithm is less robust to the wind-terrain interaction that was observed during these flight tests. BLG collisions can also be seen in Figure 6-19(c) along the protruding section of canyon-2 terrain to the southwest of the target, as well as scattered along the east canyon wall.



(a) BLG (canyon-1 top view)

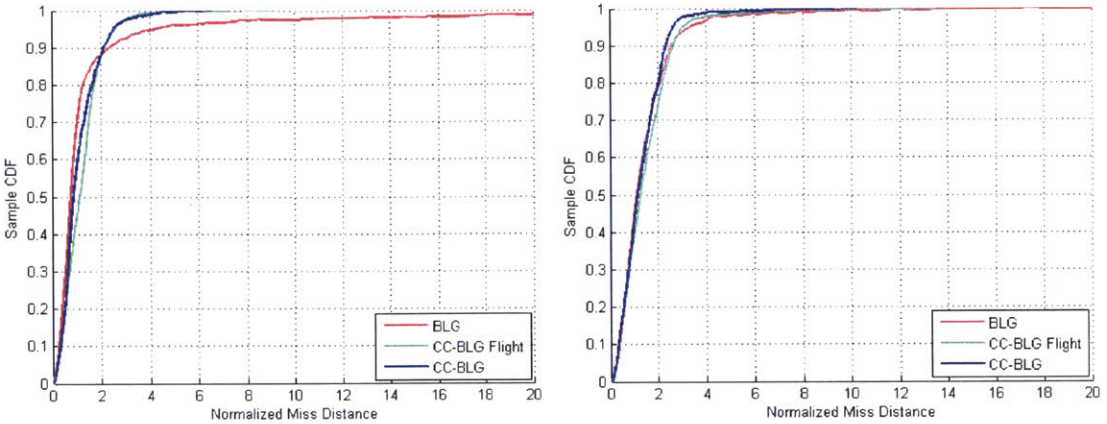
(b) CC-BLG Flight (canyon-1 top view)

(c) BLG (canyon-2 top view)

(d) CC-BLG Flight (canyon-2 top view)

Figure 6-19: BLG and CC-BLG impact points from January-FT simulation, 1500 trials

While the 5 wind profiles created from the January flight test are generally less severe than some of the more aggressive winds contained within the 25 training profiles used in Chapters 2, 4 and 5, the January-FT wind variability at lower altitudes poses a significant challenge for robust parafoil guidance in constrained terrain environments. The results in Figure 6-19 demonstrate the ability of the CC-BLG Flight algorithm to provide improved terminal guidance performance relative to BLG under these drop conditions.



(a) canyon-1

(b) canyon-2

Figure 6-20: Normalized miss distance CDF for March-FT simulation, 1200 trials

Table 6.5: Normalized miss distance data for March-FT simulation, 1200 trials

| Terrain | Algorithm | Mean | StDev | 50% | 80% | 90% | 95% | 98% | Max |
|---|---|---|---|---|---|---|---|---|---|
| canyon-1 | BLG | 1.48 | 3.29 | 0.75 | 1.26 | 2.35 | 4.33 | 13.14 | 30.17 |
| | CC-BLG Flight | 1.21 | 0.81 | 1.14 | 1.74 | 2.12 | 2.60 | 3.12 | 9.09 |
| | CC-BLG | 1.10 | 0.81 | 0.87 | 1.64 | 2.10 | 2.50 | 3.32 | 6.56 |
| canyon-2 | BLG | 1.49 | 1.74 | 1.11 | 2.09 | 2.59 | 3.39 | 5.06 | 27.29 |
| | CC-BLG Flight | 1.48 | 1.21 | 1.28 | 2.22 | 2.64 | 3.05 | 4.23 | 13.50 |
| | CC-BLG | 1.32 | 0.97 | 1.15 | 2.03 | 2.35 | 2.65 | 3.26 | 10.23 |

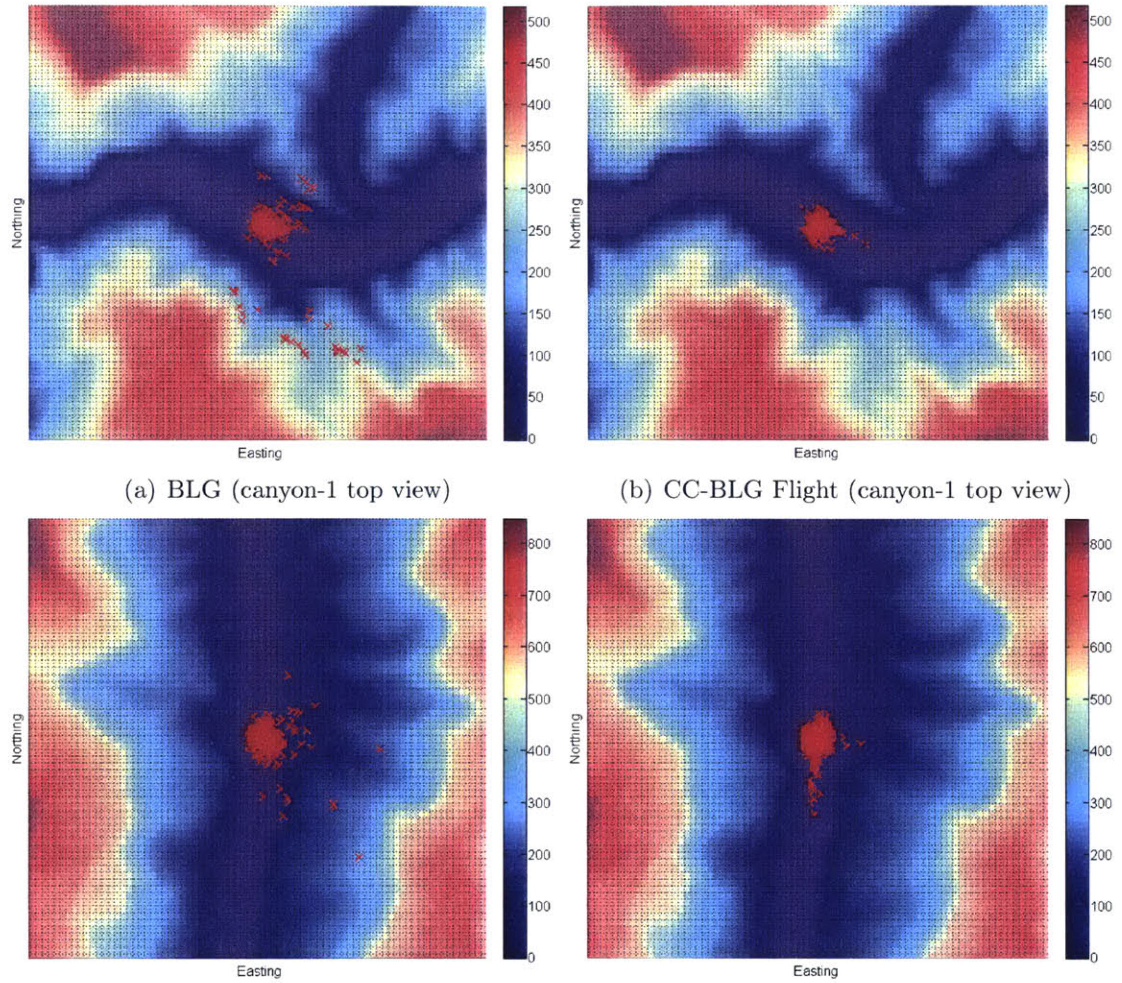
### March-FT Simulation Analysis

Figure 6-20 and Table 6.5 display the results from 1200 Monte Carlo simulation trials of the BLG, CC-BLG, and CC-BLG Flight terminal guidance algorithms using the March-FT wind conditions. As compared to the January-FT simulations, the results in Figure 6-20 and Table 6.5 reflect the decreased intensity and variability of the wind conditions recorded during the second set of March flight test experiments. The statistics in Table 6.5 indicate that the performance of the BLG algorithm has improved significantly in both the canyon-1 and canyon-2 terrain scenarios relative to the January-FT simulations shown previously in Table 6.4. In particular, the BLG mean and median miss distances have been reduced by 42% and 55% respectively in the canyon-1 scenario, and by 40% and 41% respectively in the canyon-2 scenario. This significant improvement in BLG landing accuracy can be attributed to the more passive March-FT wind conditions which lower the risk of off-nominal terrain collisions. Improvements in the mean and median landing accuracy of the CC-BLG and CC-BLG Flight algorithms are also apparent in Table 6.5, ranging from a 27.5% reduction in the mean miss distance of CC-BLG Flight in the canyon-2 terrain, to a 45% reduction in the median miss distance of CC-BLG in the canyon-1 terrain. However, the CDF plots shown in Figures 6-20(a) and 6-20(b) also demonstrate that the margin of improvement provided by each variant of CC-BLG relative to BLG is noticeably smaller than in the previous set of January-FT simulations. Figure 6-20(a) reveals that under the March-FT wind conditions, the reduced conservatism of the BLG algorithm results in improved landing accuracy over both CC-BLG and CC-BLG Flight in the canyon-1 terrain throughout most of the distribution below the 90th percentile. Although the planning conservatism used by the CC-BLG and CC-BLG Flight algorithms is adjusted dynamically via the online wind classifier (described in Section 2.3), the results in Figure 6-20(a) suggest that additional tuning may be required in order to prevent over-conservatism when the degree of environmental wind uncertainty is low.

Despite the less aggressive March-FT wind conditions, slight variations in the wind environment continue to produce several large BLG miss distances within the tail of each distribution shown in Figures 6-20(a) and 6-20(b). As a result, CC-BLG and CC-BLG Flight generate improved landing accuracy relative to BLG in the upper 10% of the distribution for the canyon-1 scenario (Figure 6-20(a)). Within the canyon-2 terrain scenario, the CDFs in Figure 6-20(b) demonstrate that all 3 terminal algorithms perform similarly up until roughly the 90th percentile of the distribution, with CC-BLG Flight producing slightly larger miss distances than either BLG or CC-BLG from the 40th to 90th percentiles. Here again we observe that both CC-BLG and CC-BLG Flight reduce the number of off-nominal terrain collisions in the tail of the distribution, and generate between a factor of 2-3 reduction in the BLG maximum miss distance. However, while CC-BLG demonstrates improved landing accuracy relative to BLG for all trials above the 80th percentile, CC-BLG Flight appears to improve only those trials above roughly the 94th percentile.

The miss distance results in Figure 6-20 and Table 6.5 reveal the effects of limiting the maximum number of optimization iterations for the CC-BLG Flight algorithm. In the March-FT wind conditions, the CDF plots in Figure 6-20 demonstrate that CC-BLG Flight does not achieve the same degree of accuracy over the interior of the distribution from the 40th to 80th percentiles, resulting in a median miss distance which is larger than BLG and CC-BLG in both the canyon-1 and canyon-2 terrain environments. When the wind conditions are calm, the results in Table 6.5 indicate that the optimizer is able to spend more time refining the trajectory solution, and less time compensating for the effects of the wind disturbances. As a result, with only half the allowable optimization iterations as the BLG and CC-BLG algorithms, CC-BLG Flight is unable to refine the trajectory solution to the same extent. For the set of March-FT simulations, this leads to slightly larger miss distances on average in the presence of complex environmental constraints. As demonstrated by the blue CDF curve of the CC-BLG algorithm shown in Figures 6-20(a) and 6-20(b), such limitations can largely be avoided with the use of additional computational resources.

The impacts points produced by the 1200 Monte Carlo trials of the BLG and CC-BLG Flight algorithms are shown in Figure 6-21 for the March-FT simulations. The BLG impact points in Figure 6-21(a) and 6-21(c) confirm the presence of large miss distances caused by collisions with the canyon-1 and canyon-2 terrain. Because BLG utilizes only a reactive replanning strategy in order to compensate for the effects of future wind disturbances, these results demonstrate that off-nominal terrain impacts can still occur even when the wind conditions are less volatile. As compared to the impact points shown previously in Figures 6-19(a) and 6-19(c), the fraction of total BLG simulation trials resulting in a terrain collision is noticeably smaller using the March-FT wind conditions, such that the majority of BLG impact points are positioned around the target location. The impact points for the CC-BLG Flight algorithm in Figures 6-21(b) and 6-21(d) continue to produce far fewer collisions than BLG in both terrain scenarios due to the added robustness of the analytic chance constraints. Although restricting the number of optimization iterations results in several large CC-BLG Flight miss distances, shown to the south of the canyon-2 target in Figure 6-21(d), these simulation trials land safely at the bottom of the



(a) BLG (canyon-1 top view)

(b) CC-BLG Flight (canyon-1 top view)

(c) BLG (canyon-2 top view)

(d) CC-BLG Flight (canyon-2 top view)

Figure 6-21: BLG and CC-BLG impact points from March-FT simulation, 1200 trials

canyon and avoid collisions with the terrain environment. Such results indicate that despite the modifications to reduce computation, CC-BLG Flight continues to remain robust to the environmental wind uncertainty.

## 6.6 Summary

This chapter presented the results from 9 flight tests of the modified CC-BLG terminal guidance algorithm conducted with a full-scale *UltraFly* parafoil system. In order to evaluate the robustness of the CC-BLG algorithm, the presence of complex terrain was simulated within the onboard GN&C software using the two Grand Canyon terrain scenarios introduced in Chapter 5. During all 9 flight test experiments, CC-BLG successfully guided the *UltraFly* parafoil to the target location while avoiding collisions with the simulated terrain environment. From the analysis of CC-BLG in-flight trajectory planning, this chapter demonstrates that the analytic chance constraints and augmented CC-BLG objective function developed in Chapter 3 effectively maintained parafoil safety, and mitigated the level of environmental risk throughout the terminal guidance phase. These results confirm that the modified flight test algorithm operates as expected given the set of CC-BLG simulations conducted previously in Chapters 4–5.

With the aid of Draper Laboratory's high fidelity simulator [48], this chapter also compared the statistical performance of the BLG, CC-BLG, and CC-BLG Flight algorithms by recreating the set of wind and drop conditions from the 9 flight test experiments. These simulations demonstrate that CC-BLG and CC-BLG Flight provide improved planning robustness relative to BLG in each terrain scenario over a series of repeated experiments using the flight test wind conditions. This improvement in performance is more pronounced when the intensity and variability of the flight test winds lead to increased planning uncertainty during the terminal guidance phase. Despite the computational limitations imposed by the available flight hardware, this chapter reveals that the robustness properties of CC-BLG Flight remain largely consistent with the full CC-BLG algorithm. When additional computational resources are available, simulation results demonstrate that CC-BLG offers a slight advantage in landing accuracy over CC-BLG Flight.

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

This thesis has presented two novel parafoil terminal guidance algorithms for robust motion planning in challenging wind and terrain environments. Through explicit modeling of the wind uncertainty, these algorithms address the limitations of current state-of-the-art parafoil guidance strategies by directly considering the possibility of future wind and terrain interaction during the trajectory planning process. The Analytic CC-RRT and CC-BLG algorithms developed in this thesis are capable of handling arbitrary initial altitudes, approach geometries, and terrain surfaces, while providing robustness to highly dynamic wind conditions that cannot be achieved through replanning alone. The effectiveness of each algorithm has been thoroughly demonstrated in simulation over a series of Monte Carlo experiments in complex terrain scenarios. Simulation results have shown that Analytic CC-RRT and CC-BLG achieve significant improvements in both mean and worst-case landing accuracy relative to the state-of-the-art Band-Limited Guidance (BLG) algorithm by substantially reducing the number of off-nominal terrain collisions caused by wind disturbances. Flight test experiments with a full-scale *UltraFly* parafoil system confirm that the optimized CC-BLG algorithm can robustly execute collision avoidance and precision landings in simulated Grand Canyon terrain by applying the method of analytic chance constraints presented in this work.

In Analytic CC-RRT, a unique multi-class wind uncertainty model is introduced to anticipate the effect of future wind disturbances. This model has been shown to accurately represent true wind behavior, and to enable online adaptation of planning conservatism to reflect the intensity and variability of the observed wind conditions. Leveraging the assumed form of the wind uncertainty model, the analytic *a priori* uncertainty distribution is derived over future parafoil trajectories. Through analytic sampling of the state uncertainty distribution, the risk of constraint violation is efficiently evaluated against arbitrary, non-convex terrain environments to ensure probabilistic feasibility with guaranteed, user-specified bounds. Simulation results have demonstrated that Analytic CC-RRT can operate successfully from a wide range of initial conditions, including altitudes up to 2000 meters, while the addition of speed penalties within the cost-to-go function has been shown to promote upwind parafoil landings with minimal effect on target accuracy.

The CC-BLG algorithm incorporates the wind uncertainty model and analytic sampling technique into the optimized BLG trajectory planning framework. A novel risk-based objective function is proposed in order to directly minimize the probability of constraint violation during trajectory optimization. This objective function utilizes several heuristics, tuned to create the desired balance between risk aversion and goal seeking behavior throughout the terminal guidance phase. By applying the principles of discrete reachability set approximation, obstacle detection and avoidance is effectively maintained during finite-horizon CC-BLG path planning in constrained terrain environments, allowing for the initialization of terminal guidance beginning from high initial altitudes, and significant target offsets. Extensive Monte Carlo simulation analysis has demonstrated the effectiveness of each of these components, and that the resulting CC-BLG algorithm is capable of generating improvements in both mean and worst-case landing accuracy relative to BLG of more than 32% and 66%, respectively, in complex terrain scenarios. The results of nine flight test experiments demonstrate that CC-BLG performed as expected given prior simulations, and is readily implementable for online trajectory planning using commercially available flight hardware.

## 7.2 Future Work

### 7.2.1 Planning Conservatism

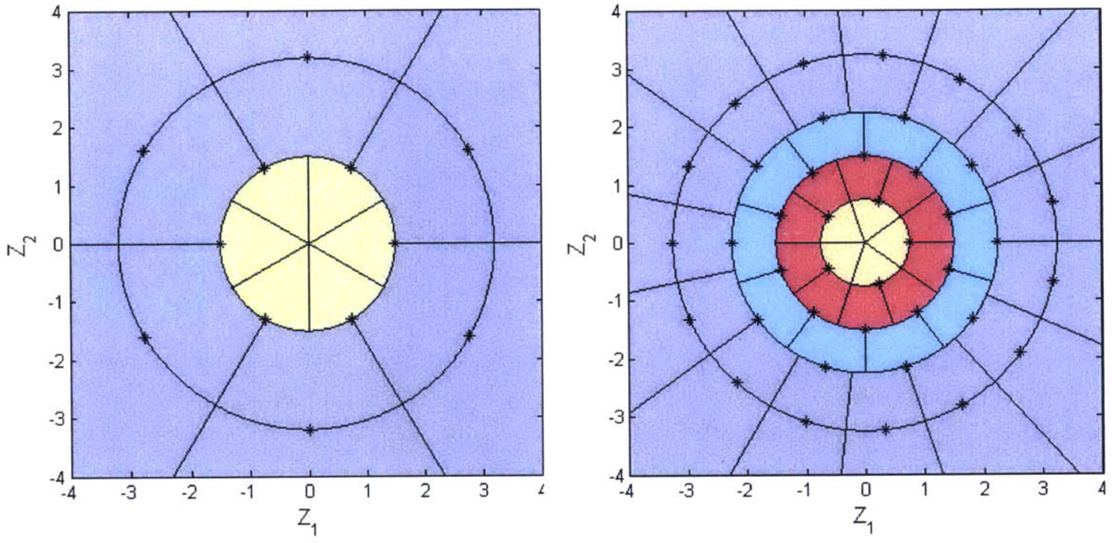
Planning conservatism can be a valuable quality for parafoil terminal guidance algorithms by promoting robustness to future wind disturbances in the presence of environmental obstacles. However, over-conservatism in the planner also has the potential to degrade performance when wind conditions are less dynamic, and the risk to parafoil safety is minimal. Although the Analytic CC-RRT and CC-BLG algorithms were shown to handle this dichotomy well for a wide variety of wind and terrain conditions, the effect of over-conservatism was observed in Section 6.5 during Monte Carlo simulation trials of CC-BLG and CC-BLG Flight using the less aggressive March-FT wind conditions. Further refinements to the wind model, classifier, and analytic uncertainty sampling technique may help to prevent over-conservatism by improving the fidelity of the wind prediction.

#### Wind Model Training Set

First, the training set of wind profiles should be expanded using additional real-world drop data in order to capture an increased variety of possible wind behaviors. The DP-means clustering algorithm in Section 2.3.2 can then be used to generate a new set of $N_C$ wind classes, including those with less aggressive characteristics. In the event that DP-means identifies $N_C \geq 5$ clusters within the expanded set of training wind profiles, Error-Correcting Output Coding (ECOC) could be incorporated as a low bias and low variance classification method for supervised learning problems involving $k \gg 2$ classes [75, 76]. These modifications may allow the classifier to better distinguish between various levels of environmental risk, without significantly effecting model complexity.

#### Wind Model Feature Vector

To construct the feature vector (2.23) for each wind profile, future work should also consider replacing the maximum value of wind magnitude, rate of change of wind



(a) $N_S = 12, N_E = 2$ (b) $N_S = 40, N_E = 4$

Figure 7-1: Alternative arrangements of weighted covariance samples

magnitude, and rate of change of wind direction with the standard deviation of each feature, respectively. Characterizing the effect of wind disturbances using the maximum function has the potential to create bias during both offline clustering, and online classification, due to possible sensor noise and measurement outliers. If a single large outlier is observed, this information will be retained within the feature vector throughout all future time steps, and may bias classification toward greater conservative. On the other hand, the standard deviation represents another useful metric for describing wind variability, while reducing the risk of classification bias.

### Weighted Covariance Sampling

Another possible modification to prevent over-conservatism is an alternative arrangement of the weighted covariance samples. For the CC-BLG Flight algorithm presented in Chapter 6, sampling the uncertainty distribution along a single covariance ellipse with standard deviation $\sigma > 1.5$ can encourage planner conservatism, particularly when the total number of covariance samples, $N_S$, is small. If only a single covariance ellipse is used for sampling, all samples carry a probability weight of $P(S) = 1/N_S$ regardless of the value of $\sigma$. Sampling the distribution at large values of $\sigma$ will therefore penalize sample collisions at locations far from the nominal trajectory state where the parafoil is unlikely to be. An alternative approach is to arrange covariance samples across multiple $\sigma$-ellipses, even for small $N_S$, in order to better distribute the sample probability weights. To compensate for reduced sampling coverage along each ellipse, equi-spaced covariance samples can be arranged at rotated angular offsets between consecutive $\sigma$-ellipses, as shown in Figure 7-1. Figure 7-1(a) demonstrates a possible arrangement of $N_S = 12$ samples over $N_E = 2$ covariance ellipses for $\{\sigma_1, \sigma_2\} = \{1.5, 3.2\}$ and $\{N_{S_1}, N_{S_2}\} = \{6, 6\}$, respectively. The yellow area in Figure 7-1(a) represents the region of the probability distribution assigned to samples along the $\sigma_1$-ellipse, while the gray area represents the remaining region of the probability distribution assigned to samples along the $\sigma_2$-ellipse. This arrangement with $N_S = 12$ samples could potentially be used for CC-BLG Flight. For the case of $N_S = 40$, Figure 7-1(b) provides an example arrangement of covariance samples using $N_E = 4$ ellipses for $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\} = \{0.75, 1.5, 2.25, 3.25\}$ and $\{N_{S_1}, N_{S_2}, N_{S_3}, N_{S_4}\} = \{5, 10, 10, 15\}$, respectively. Future work should evaluate the benefit of these alternative covariance sample arrangements for improving planning conservatism while maintaining robustness to the wind uncertainty.

### 7.2.2 Analytic Uncertainty in 3-Dimensions

Extending the wind uncertainty model to 3-Dimensions to incorporate the effects of vertical wind disturbances, such as updrafts and downdrafts, could provide an additional degree of robustness for parafoil guidance in complex terrain environments. Such an extension was proposed in Ref.[47] and is worth considering for future Analytic CC-RRT and CC-BLG development. Under the consistent assumption that the wind uncertainty along the $z$-axis is uncorrelated with uncertainty in the $xy$-plane, the colored noise parameters for the wind model (2.31) could be re-tuned in 3D by repeating the procedure outlined in Section 2.3.2. In order to isolate each component, this approach would require tuning first the $x/y$ wind uncertainty by neglecting vertical wind disturbances, and then the $z$ wind uncertainty by neglecting the $x/y$ wind disturbances [47]. The difference in impact time between the propagated parafoil dynamics subject to true vertical wind, and subject to the estimated mean vertical wind, could be used to derive a similar *vertical* distance metric and empirical CDF to the one developed in Section 2.3.2 for $x/y$, so as to represent the cumulative effect of updrafts and downdrafts on parafoil altitude. After re-deriving the analytic uncertainty distribution following the approach in Section 2.4, a method of 3D uncertainty sampling over covariance ellipsoids could be developed using a similar approach to the one described in Ref.[77].

Although 3D analytic uncertainty sampling would require additional computation in comparison to the 2D sampling approach described in this work, 3D sampling may result in valuable robustness properties—particularly for parafoil guidance scenarios such as canyons, valleys, and urban environments, where updraft and downdrafts can have a significant impact on the parafoil dynamics. To further evaluate Analytic CC-RRT and CC-BLG performance, wind profile data should be recorded within these types of complex terrain scenarios and incorporated into future simulation analysis. This would enable more consistent modeling of the challenging wind-terrain interaction associated with a particular target environment, while providing useful training data for wind classification.

# Bibliography

[1] 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.

[2] R. Benney, J. Barber, J. McGrath, J. McHugh, G. Noetscher, and S. Tavan, "The new military applications of precision airdrop systems," in *AIAA Infotech@Aerospace Conference*, (Arlington, VA), (AIAA-2005-7069), September 2005.

[3] 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.

[4] O. Yakimenko and N. Slegers, "Using direct methods for terminal guidance of autonomous aerial delivery systems," in *European Control Conference*, European Control Association, August 2009.

[5] Y. de Lassat de Pressigny, R. Benney, M. Henry, R. Bechet, and J. Wintgens, "Pacd2008: Operational requirements fulfilled," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2009.

[6] D. Jalbert, "Multi-cell wing type aerial device," November 1966. US Patent 3,285,546.

[7] M. Keller, "Supply drop in afghanistan." http://www.march.afrc.af.mil/shared /media/photodb/photos/090827-F-8155K-003.JPG, April 2015.

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

[9] C. Knapp and W. Barton, "Controlled recovery of payloads at large glide distances, using the para-foil," *Journal of Aircraft*, vol. 5, pp. 112–118, March 1968.

[10] S. Tavan, "Status and context of high altitude precision aerial delivery systems," in *In Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[11] P. Hattis, B. Appleby, T. Fill, and R. Benney, "Precision guided airdrop system flight test results," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, June 1997.

[12] D. Campbell, T. Fill, P. Hattis, and S. Tavan, "An on-board mission planning system to facilitate precision airdrop," in *Infotech@Aerospace Conferences*, American Institute of Aeronautics and Astronautics, September 2005.

[13] B. J. Rademacher, P. Lu, A. L. Strahan, and C. J. Cerimele, "In-flight trajectory planning and guidance for autonomous parafoils," in *Journal of Guidance, Control, and Dynamics*, vol. 32, pp. 1697–1712, American Institute of Aeronautics and Astronautics, 2009.

[14] M. Ward and M. Costello, "Adaptive glide slope control for parafoil and payload aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 36, pp. 1019–1034, July 2013.

[15] J. Murray, A. Sim, D. Neufeld, P. Rennich, S. Norris, and W. Hughes, "Further development and flight test of an autonomous precision landing system using a parafoil," tech. rep., NASA, 1994.

[16] 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.

[17] J. McHugh, R. Benney, J. Miletti, and P. Mortaloni, "Planning, execution, and results of the precision airdrop technology conference and demonstration (2003)," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2005.

[18] R. Benney, M. Henry, K. Lafond, A. Meloni, and S. Patel, "Dod new jpads programs and nato activities," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2009.

[19] P. Hattis, K. Angermueller, T. Fill, R. Wright, R. Benney, D. LeMoine, and D. King, "In-flight precision airdrop planner follow-on development program," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2003.

[20] K. Bergeron, G. Noetscher, M. Shurtliff, S. Tavan, and F. Deazley, "Longitudinal control for ultra-light weight guided parachute systems," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[21] "Ultrafly precision guided airdrop system." http://www.wamore.com/Products/Military-Products.aspx, 2015.

[22] D. Carter, S. 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.

[23] I. Kaminer and O. Yakimenko, "Development of control algorithms for the autonomous gliding delivery system," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2003.

[24] T. Jann, "Advanced features for autonomous parafoil guidance, navigation and control," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2005.

[25] N. Slegers and O. Yakimenko, "Terminal guidance of autonomous parafoils in high wind to airspeed ratios," in *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, pp. 336–346, 2011.

[26] 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.

[27] 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.

[28] K. Bergeron, S. 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.

[29] M. Ward, A. Gavrilovski, and M. Costello, "Flight test results for glide slope control of parafoil canopies of various aspect ratios," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2011.

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

[31] J. Cleminson, "Path planning for guided parafoils: An alternative dynamic programming formulation," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2013.

[32] D. Carter, L. Singh, L. Wholey, S. Rasmussen, T. Barrows, S. George, M. McConley, C. Gibson, S. Tavan, and B. Bagdonovich, "Band-limited guidance and control of large parafoils," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2009 (AIAA 2009-2981).

[33] O. Yakimenko and N. Slegers, "Optimization of the ads final turn maneuver in 2d and 3d," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, May 2011.

[34] N. Slegers, A. Brown, and J. Rogers, "Experimental investigation of stochastic parafoil guidance using a graphics processing unit," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[35] B. Klein and J. Rogers, "A probabilistic approach to unguided airdrop," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[36] H. Niederreiter, "Random number generation and quasi-monte carlo methods," in *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics, 1992.

[37] L. Fowler and J. Rogers, "Bezier curve path planning for parafoil terminal guidance," in *Journal of Aerospace Information Systems*, vol. 11, pp. 300–315, American Institute of Aeronautics and Astronautics, 2014.

[38] M. Ward, M. Cacan, E. Scheuermann, M. Costello, K. Bergeron, G. Noetscher, and M. Shurtliff, "Flight test results of recent advances in precision airdrop guidance, navigation, and control logic," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[39] M. Cacan, E. Scheuermann, M. Ward, and M. Costello, "Use of ground-based wind measurements for improved guided airdrop accuracy," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[40] T. Herrmann, M. Ward, M. Costello, and N. Slegers, "Utilizing ground-based lidar for autonomous airdrop," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2013.

[41] B. Chiel and C. Dever, "High wind autonomous parafoil guidance," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[42] M. Ward, A. Gavrilovski, and M. Costello, "Glide slope control authority for parafoil canopies with variable incidence angle," *Journal of Aircraft*, vol. 50, pp. 1504–1513, September 2013.

[43] M. Ward, S. Culpepper, and M. Costello, "Parafoil control using payload weight shift," *Journal of Aircraft*, vol. 51, pp. 204–215, January 2014.

[44] E. Scheuermann, M. Ward, M. Cacan, M. Costello, and K. Bergeron, "Flight testing of autonomous parafoils using upper surface bleed air spoilers," in *Aerodynamic Decelerator Systems Technology Conferences*, American Institute of Aeronautics and Astronautics, March 2015.

[45] 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).

[46] B. Luders, I. Sugel, and J. P. How, "Robust trajectory planning for autonomous parafoils under wind uncertainty," in *AIAA Infotech@Aerospace Conference*, (Boston, MA), August 2013.

[47] I. Sugel, "Robust planning for autonomous parafoil," Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, September 2013.

[48] 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.

[49] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep. 98-11, Iowa State University, October 1998.

[50] 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.

[51] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *American Control Conference (ACC)*, 2006.

[52] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, July 1957.

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

[54] 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.

[55] 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.

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

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

[58] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 3rd Edition.* Prentice-Hall, 2010.

[59] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via Bayesian nonparametrics," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, (Edinburgh, Scotland), 2012.

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

[61] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, 1st ed., 2007.

[62] 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.

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

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

[65] 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.

[66] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, pp. 846–894, June 2011.

[67] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, (Boston, MA), August 2013.

[68] B. Luders, *Robust Sampling-based Motion Planning for Autonomous Vehicles in Uncertain Environments.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, May 2014.

[69] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Robotics: Science and Systems (RSS)*, June 2010.

[70] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, pp. 846–894, June 2011.

[71] W. Hoover, "Algorithms for confidence circles and ellipses," tech. rep., NOAA, 1984.

[72] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proc. American Control Conference the 2004*, vol. 6, pp. 5576–5581, 30 June–2 July 2004.

[73] M. Stoeckle, "Fault detection, isolation, and recovery for autonomous parafoils," Master's thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 2014.

[74] T. Barrows, "Apparent mass of parafoils with spanwise camber," *Journal of Aircraft*, vol. 39, pp. 445–451, May 2002.

[75] 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.

[76] E. Kong and T. Dietterich, "Error-correcting output coding corrects bias and variance," in *In Proceedings of the 12th International Conference on Machine Learning*, pp. 313–321, Morgan Kaufmann, 1995.

[77] B. D. Luders, "Robust trajectory planning for unmanned aerial vehicles in uncertain environments," Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, September 2008.