Control and Optimization for Aerospace Systems with Stochastic Disturbances, Uncertainties, and Constraints

by

Andrew W. Berning Jr.

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Anouck R. Girard, Co-Chair
Professor Ilya V. Kolmanovsky, Co-Chair
Stefan R. Bieniawski, Blue Origin
Professor Aaron J. Ridley

Andrew W. Berning Jr.

awbe@umich.edu

ORCID iD: 0000-0001-6388-4200

For my parents, Michelle and Andy

ACKNOWLEDGMENTS

No man is an island, least of all myself while working on this PhD. During my time in Ann Arbor, I was aided and inspired by some of the most exceptional people it has ever been my pleasure to meet, and I would like to thank a few of them here.

Firstly, I would like to thank Professor Ilya Kolmanovsky and Professor Anouck Girard. I could not have possibly asked for a more complimentary and effective pair of advisors. Your broad theoretical and technical knowledge greatly improved the quality of my work and has made me a better engineer. Additionally, your professional and personal mentorship has put me on the path towards a meaningful career in engineering research for which I am extremely grateful. Thank you both for the faith you placed in me, and for the guidance you continue to show me.

Secondly, I would like to thank the other members of my committee, Professor Aaron Ridley and Stefan Bieniawski. Professor Ridley, your comments and feedback coming from the realm of space science helped me look at my research with fresh eyes and understand how it fits into the greater body of literature. Stefan, your technical and professional guidance greatly improved the applicability of my research and kept me grounded on real engineering problems. I look forward to tackling many more exciting projects with you at Blue Origin. Thank you both.

Thank you also to Sarah D'Souza at NASA Ames Research Center, Chris Petersen at The Air Force Research Laboratory, and Fred Leve at The Air Force Office of Scientific Research for providing valuable technical guidance at various stages of this PhD.

I would also like to thank my fellow labmates and friends who helped make FXB an enjoyable and exciting workplace. Thank you Dominic, Nan, Will, Greg, and Huayi. You all were absolutely critical in my survival of the coursework and qualifying exam portions of this degree, and your friendship and technical discussions made the office genuinely fun. Dominic and Nan, writing papers with you two has been one of the great pleasures of this PhD, and I cannot overstate how much I have benefited from your technical abilities.

I would like to thank my advisor and mentors during my time at the University of Texas who helped lay the foundation for my PhD. Thank you Professors Behçet Açikmeşe and Matthew Harris for introducing me to the world of optimal control and challenging me to further develop my mathematical skill sets. Thank you Miki for the friendship, puns, and for being a wonderful role model for an aspiring GNC researcher.

I would like to thank Joel, Grace, Gautam, Michelle, Sam, and all of the wonderful people of Blue Origin who made my internship there so profoundly enjoyable and formative. Heather and Sarah, it is difficult to imagine a more challenging or reward internship program than yours. Thank you Andrew for being a wonderful mentor, co-author, and manager who knew when to let me explore interesting problems and when to rein me back in.

It is no great secret that I place a large emphasis on my social health, and I am eternally grateful for all of my friends who made Ann Arbor home: Mitch, Tojo, Shad, Ben, Anna, Hannah, Ali, Troy, Matt D., Matt S., Malcolm, and all the other members of the Michigan Triathlon team. These friendships provided much-needed respites from the stresses of graduate school that allowed me to resume research with a renewed vigor. To Ryan P., Jimmy, Alli, Ryan C., Shyam, Ishan, Cale, Zack, Hallie, Karsten, Laurence, Sam, Luke, Mark, Viet and all my other TAMS, PWB, and out-of-state friends: thank you for reminding me of the world outside academia and for providing ample opportunities to leave Michigan in the winters.

To my grandparents, aunts, uncles, and cousins, thank you for your support and interest in my research which must seem terribly boring to anyone other than myself. Thank you for the periodic check-ins, reminders to graduate, and for your understanding when I am many states away during family gatherings.

To my oldest and closest friend Matthew, thank you for the unconditional love and camaraderie. I am lucky to have a brother like you, with whom to discuss the day's news, celebrate our successes, and commiserate after failures. I look forward to watching you achieve great things and I look forward to the end of your "when will you graduate?" jokes.

Thank you Sam for the love, friendship, and support necessary get through this last and most challenging year of the degree. Your intelligence, compassion, and insuppressible levity serve as daily sources of inspiration. I love you and look forward to exploring Seattle with you and Hurley.

Most of all, thank you to my parents Michelle and Andy for the love, support, and confidence you have always had in me, even at times when my academic goals appeared to be certainly out of reach. Your sacrifices and hard work provided me with endless opportunities, and any success I may enjoy is a reflection on you. In life, friendship, and love, you two are the idyllic role models; watching you has taught me who I want to be. I cannot possibly repay you, but hopefully this work and my subsequent removal from your cell phone plan are a start.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

sUAS small Unmanned Aircraft Systems

UTM UAS Traffic Management

UQ Uncertainty Quantification

FAA Federal Aviation Administration

PSD Power Spectral Density

MC Monte Carlo

RRT Rapidly-exploring Random Trees

ODE Ordinary Differential Equation

LC Linear Covariance

QP Quadratic Programming

SRMP Spacecraft Relative Motion Planning

MPC Model Predictive Control

NMPC Nonlinear Model Predictive Control

CW Clohessy-Wiltshire

DoF Day of Flight

PDF Probability Distribution Function

LEO Low Earth Orbit

NRHO Near Rectilinear Halo Orbit

CR3BP Circular Restricted Three Body Problem

ER3BP Elliptic Restricted Three Body Problem

TDO Time Distributed Optimization

SQP Sequential Quadratic Programming

OCP Optimal Control Problem

NASA National Aeronautics and Space Administration

ABSTRACT

The topic of this dissertation is the control and optimization of aerospace systems under the influence of stochastic disturbances, uncertainties, and subject to chance constraints. This problem is motivated by the uncertain operating environments of many aerospace systems, and the ever-present push to extract greater performance from these systems while maintaining safety. Explicitly accounting for the stochastic disturbances and uncertainties in the constrained control design confers the ability to assign the probability of constraint satisfaction depending on the level of risk that is deemed acceptable and allows for the possibility of theoretical constraint satisfaction guarantees.

Along these lines, this dissertation presents novel contributions addressing four different problems: 1) chance-constrained path planning for small unmanned aerial vehicles in urban environments, 2) chance-constrained spacecraft relative motion planning in low-Earth orbit, 3) stochastic optimization of suborbital launch operations, and 4) nonlinear model predictive control for tracking near rectilinear halo orbits and a proposed stochastic extension. For the first problem, existing dynamic and informed rapidly-expanding random trees algorithms are combined with a novel quadratic programming-based collision detection algorithm to enable computationally efficient, chance-constrained path planning. For the second problem, a previously proposed constrained relative motion approach based on chained positively invariant sets is extended in this dissertation to the case where the spacecraft dynamics are controlled using output feedback on noisy measurements and are subject to stochastic disturbances. Connectivity between nodes is determined through the use of chance-constrained admissible sets, guaranteeing that constraints are met with a specified

probability. For the third problem, a novel approach to suborbital launch operations is presented. It utilizes linear covariance propagation and stochastic clustering optimization to create an effective software-only method for decreasing the probability of a dangerous landing with no physical changes to the vehicle and only minimal changes to its flight controls software. For the fourth problem, the use of suboptimal nonlinear model predictive control (NMPC) coupled with low-thrust actuators is considered for station-keeping on near rectilinear halo orbits. The nonlinear optimization problems in NMPC are solved with time-distributed sequential quadratic programming techniques utilizing the FBstab algorithm. A stochastic extension for this problem is also proposed. The results are illustrated using detailed numerical simulations.

CHAPTER 1

Introduction

The global aerospace industry is diverse and economically influential, spanning applications from small Unmanned Aircraft Systems (sUAS) to interplanetary spacecraft. The 2020 Federal Aviation Administration (FAA) forecast estimates a total number of commercial sUAS in the U.S. to be 385,000 as of December 2019, a 40% growth over 2018 [1]. This large growth rate is driven in part by commercial applications such as agricultural surveillance [2], disaster management [3], and delivery [4]. Additionally, the forecast calls for U.S. carrier domestic airlines' passenger growth over the next 20 years to average 2.0 percent per year. In 2018, the aerospace and defense industries contributed $374 billion (or 1.8%) to the U.S. gross domestic product (GDP), driven largely by the aircraft and space systems sectors [5]. Figure 1.1 show the recovery of the global orbital space launch cadence since the early 2000's lull [6]. This growth has been driven in large part by non-government launch providers such as SpaceX, ULA, and Rocketlab, and programs such as the National Aeronautics and Space Administration (NASA) Commercial Resupply Services and Commercial Crew Program.

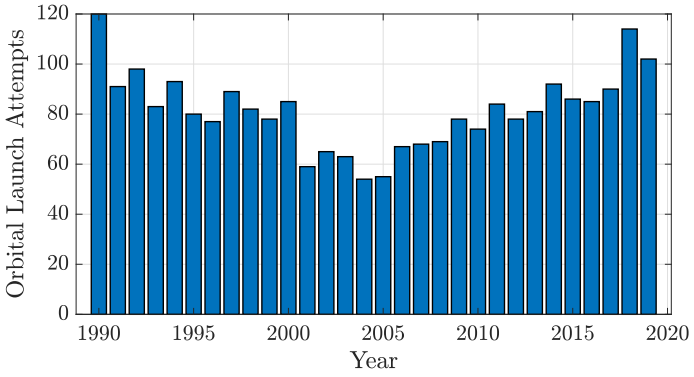

Figure 1.1: Global orbital launch attempts by year.

The industry spans a broad range of diverse applications, but commonalities include

tight safety factors, congested operating environments, fuel consumption considerations, and the ever-present push to extract greater performance from these systems. Another important commonality is the difficulty in modeling these complex systems, resulting in significant differences between the engineering models' predicted behavior, and the physical systems' actual behavior. Sources of this uncertainty are numerous, including imperfect vehicle mass properties information, wind gusts, navigational uncertainty, the oblateness of the Earth, solar radiation pressure, thruster mis-alignments, etc.

A 1976 paper by George Box popularized the saying "All models are wrong, but some are useful." [7], which has a particular relevance to the fields of model-based control and optimization. A controls engineer cannot hope to perfectly model every minute complexity of a physical system and so the real world application of any controls engineering effort will inevitably behave differently than the model upon which the control system is based. Of course, some models are sufficiently representative of the real-world system as to be useful, and for constraint satisfaction purposes, the differences between the two can often be accounted for with conservative margins or safety factors. The resulting control or optimization scheme can then be validated with some combination of simulation and experimentation, to ensure that the margins are sufficient for safe operation.

This thesis considers control methodologies that differ from the traditional approach by directly addressing modeling uncertainty, with applications to several different aerospace systems including multirotor sUAS, fixed-wing sUAS, and spacecraft in the atmospheric, Low Earth Orbit (LEO), and cis-lunar flight regimes.

Consider an example scenario in which we are attempting to find the shortest safe flight path for a sUAS through an urban environment. The initial state and desired final state are both fixed, and the constraint of interest is avoiding collisions with buildings. If we assume a deterministic model of sUAS flight, satisfying this constraint is a well-defined problem – plan a flight trajectory that does not intersect the buildings.

If instead we assume that the model is inaccurate or uncertain, then the future behavior of the system cannot be predicted exactly. Consider an exogenous disturbance of stochastic wind and the propagation of this uncertainty along the vehicle's trajectory, in which we characterize the vehicle's state at each time instant with a Probability Distribution Function (PDF) which does not have compact support. In this case, the constraints in this path planning problem must be posed as a chance constraint, e.g., ensure that there is a β probability that the sUAS does not collide with the buildings.

Using techniques that are detailed in Chapter 2, this sample problem is solved for two different values of β and the results are depicted in Figures 1.2 and 1.3. The tubes plotted in both sub-figures represent the set inside which the vehicle has a β-probability of being

for all time along the flight trajectory. Accordingly, this tube is larger for $\beta = 0.9999$ than it is for $\beta = 0.99$. The probability tube starts as single point since the initial state of the vehicle is known deterministically, grows with time due to the wind disturbance, and eventually stops growing as the vehicle's feedback control law makes large deviations from the nominal flight path unlikely. Thus, satisfaction of the chance constraint is achieved when the probability tube does not intersect the obstacles. The buildings with which we are trying to avoid a collision are represented below as red cuboids. In Figure 1.2, the collision avoidance constraint is satisfied for $\beta = 0.99$ (or 99% probability) and the resulting probability tube is small enough to fit in the gap between the two buildings without collision. For Figure 1.3, where $\beta = 0.9999$, the resulting probability tube is too large to fit within the gap, so the optimal solution is a route that goes around the building, resulting in a longer flight path.

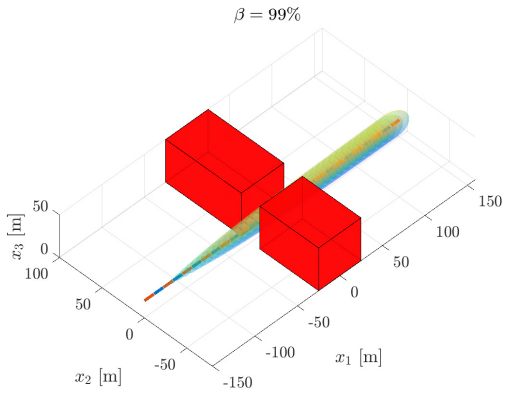

Figure 1.2: Stochastic path planning for $\beta = 0.99$ or 99%.

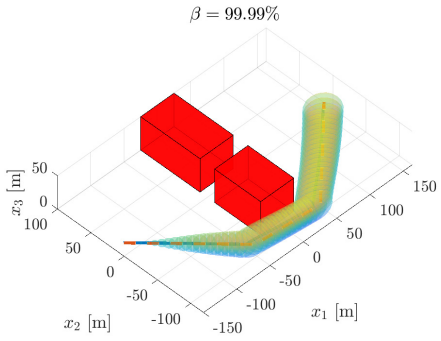

Figure 1.3: Stochastic path planning for $\beta = 0.9999$ or 99.99%.

This sample problem illustrates a significant advantage of addressing uncertainty directly in the control of a stochastic system, when compared to a deterministic control scheme with safety margins: the ability to directly assign the probability of constraint satisfaction, which can sometimes result in dramatically different control actions (and therefore costs) depending on the level of risk that is deemed acceptable. Additionally, stochastic control strategies allow for the possibility of theoretical guarantees, such as the constraint satisfaction guarantees presented in Chapter 3.

Figures 1.2 and 1.3 provide an illustration of these principles for fixed-wing atmospheric flight, but similar considerations are valid for many other aerospace systems, as is shown in the four subsequent chapters of this thesis (chapters 2-5). Common considerations for these problems include control constraints (such as thrust saturation) and obstacle

avoidance constraints. The latter could involve not only avoiding collisions with vehicles, buildings, or other terrain features, but also reflects practical requirements dictated by flight keep-out zones or sensor protection. Additional considerations include the fuel efficiency of the resulting trajectories and the computational efficiency of the control algorithms.

Considering uncertainty directly in control schemes is not a new development. Existing strategies for stochastic path planning include convex programming [8], Markov Decision Processes [9], Graph Search [10], and Rapidly-exploring Random Trees (RRT) [11, 12]. The field of stochastic Model Predictive Control (MPC) is similarly well established [13–16], and the closely related problem of propagating uncertainty has been addressed with Monte Carlo (MC) simulations [17,18], Linear Covariance (LC) Uncertainty Quantification (UQ) [19, 20], unscented transformation [21, 22], polynomial chaos expansions [23], and Gaussian mixture models [24–26].

Chapters 2-5 are based on manuscripts that have been published in the form of either a peer-reviewed journal article or conference proceedings and each chapter includes more detailed literature review and problem statement. A summary of our technical contributions and the layout of this dissertation is as follows:

Chapter 2: Rapid Uncertainty Propagation and Chance-Constrained Path Planning for Small Unmanned Aerial Vehicles

Chapter 2 is based on material in [27] and [28] and details the development of a computationally efficient stochastic path planning algorithm for multirotor and fixed-wing sUAS. In this chapter, we use linear covariance propagation in combination with a quadratic programming-based collision detection algorithm to rapidly validate declared flight plans. These algorithms are combined with a Dynamic, Informed RRT* algorithm and detailed numerical examples for both fixed-wing and quadrotor sUAS models are presented. The main contribution is the amalgamation of existing Dynamic RRT* and Informed RRT* algorithms, and the addition of an obstacle buffer resizing technique, resulting in a single algorithm that allows computationally efficient, chance-constrained path planning for sUAS. In particular, this contribution addresses a dilemma in chance-constrained path planning under uncertainty: trajectory re-planning changes the outcome of the covariance propagation, which, when obstacles or exclusion zones are involved, may require further re-planning.

Chapter 3: Spacecraft Relative Motion Planning Using Chained Chance-Constrained Admissible Sets

Chapter 3 is based on material in [29] and presents a graph search-based stochastic path planning technique for spacecraft in LEO.

This chapter extends a previously proposed constrained relative motion approach based on chained positively invariant sets to the case where the spacecraft dynamics are controlled using output feedback on noisy measurements and are subject to stochastic disturbances. The methodology consists of constructing a connected graph (referred to as the virtual net) of static equilibria nodes in the Clohessy-Wiltshire-Hill frame. Connectivity between nodes is determined through the use of chance-constrained admissible sets, guaranteeing that constraints are met with a specified probability. Unlike [30], where positive invariant sets being chained are sublevel sets of Lyapunov functions, here we exploit chance constrained admissible sets to determine connectivity of forced equilibria. These sets are near maximal and hence allow more vertices in the graph to be connected. Furthermore, our approach is able to accommodate stochastic measurement and process noise, as opposed to bounded disturbances as in [30]. We also propose a novel approach of handling non-convex constraints by exploiting their inner approximation with a union of convex constraints and we extend the connectivity conditions to this case. Further we demonstrate that the chance constraint used as a requirement in the construction of chance constrained admissible sets holds for the closed-loop trajectories with switching between forced equilibria.

Chapter 4: Suborbital Reentry Uncertainty Quantification and Stochastic Optimization

Chapter 4 is based off material in [31] and focuses on two separate but related problems: rapid uncertainty propagation for a vehicle reentering the atmosphere and stochastic optimization of ascent trajectories. For the former, we develop an algorithm that takes the vehicle's dynamics, physical parameters, initial state, and atmospheric conditions as inputs, and propagates the uncertainty associated with the vehicle's state trajectory over the specified flight horizon. For the latter, we develop an optimization method that, for a finite number of ascent reference trajectories and assuming the ability to choose the best on the day of flight, minimizes the probability of a landing zone constraint violation. Through a combination of linear covariance uncertainty quantification and clustering optimization, we solve for a small, finite set of optimal ascent reference trajectories for the launch vehicle to follow during the boost mission phase. The result is a control scheme that 'shapes' the landing position probability density function to avoid dangerous landing zones, increasing

the probability of a successful mission with no physical changes to the space vehicle and only minimal changes to its flight control software.

Chapter 5: Suboptimal Nonlinear Model Predictive Control Strategies for Tracking Near Rectilinear Halo Orbits

This chapter investigates the use of Nonlinear Model Predictive Control (NMPC) coupled with low-thrust actuators for station-keeping on a Near Rectilinear Halo Orbit (NRHO), a subclass of halo orbits in cis-lunar space. It is demonstrated through numerical simulations that NMPC is able to stabilize a spacecraft to a reference orbit and handle control constraints. Further, it is shown that the computational burden of NMPC can be managed using new optimization algorithms [32] and Time Distributed Optimization (TDO) [33], a kind of suboptimal MPC, to demonstrate that it is possible to meet closed-loop performance requirements using little computational power. Our approach contrasts with the existing literature that typically considers impulsive thrusters and uses control methodologies based on linearized models. The chapter concludes with a discussion of a stochastic extension to the problem.

CHAPTER 2

Rapid Uncertainty Propagation and Chance-Constrained Path Planning for Small Unmanned Aerial Vehicles

2.1 Introduction

2.1.1 Motivation

There were 110,604 registered sUAS in the United States at the end of 2017, and that number is expected to quadruple by 2022 [34]. Most of these are of fixed-wing or multirotor design, though other hybrid designs have garnered recent interest [35]. There has been great interest, accordingly, in a UAS Traffic Management (UTM) system to handle the demands of this rapidly growing aviation sector [36, 37]. In at least one potential design of such a system, real-time communication channels exist between a central computation platform and the sUAS. This would provide the vehicles access to valuable computation resources and information about nearby vehicles, terrain, and atmospheric conditions, allowing for safe and efficient route planning.

One of the responsibilities of such a UTM system will likely be to apply a risk-based approach where geographical needs and use cases determine the airspace performance requirements [38]. Addressing this will require a sUAS trajectory prediction model that validates vehicle flight performance and allows for UTM to determine whether or not the vehicle can operate in the airspace given real-time information about wind, other vehicles, and/or obstacles. Given a large number of vehicles predicted to be in operation, this trajectory prediction model must accommodate multiple vehicle types and airspace environments (wind, terrain, etc.). Another challenge in this setting is that the trajectory could depend on proprietary information such as control systems, methods and gain tuning specific to a particular vehicle. Implementing a system that is reliant on such information could be

prohibitive due to constant modifications that would be required to accommodate diverse and evolving control laws and due to the potential legal barriers to acquiring proprietary information from all operators.

In the context of UTM operations, where varied vehicle types and uncertainties are expected, there is a need for a model that represents expected vehicle performance while accounting for the uncertainty in the context of operational environments. The motivation for the work detailed in this chapter is to develop computationally efficient trajectory validation and planning algorithms that utilize UQ and propagation techniques to provide an assessment of the vehicle performance and risk of violating constraints.

2.1.2 Problem Statement

The work described in this chapter focuses on two separate but related problems: rapid uncertainty propagation for trajectory validation and chance-constrained path planning.

For the former, we seek an algorithm that takes the vehicle's dynamics, initial state, parameters, and desired trajectory as inputs, and outputs some quantification of the uncertainty associated with the vehicle's state trajectory over the specified flight horizon, as well as an assessment of whether the probability of trajectories violating constraints is sufficiently low at any given time instant. For the latter, we are interested in the ability to re-plan or repair the trajectory if it is found that the vehicle's probabilistic trajectory tube intersects with a keep-out zone. Such a tube and constraint are illustrated in Figure 2.1. In this example scenario, the vehicle's state is known deterministically initially, and the uncertainty tube grows as the vehicle attempts to follow the desired trajectory (solid line) subject to disturbances such as wind gusts.

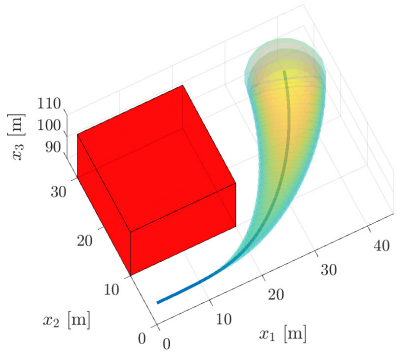

Figure 2.1: Nominal trajectory (solid line), probabilistic trajectory tube in the presence of uncertainties and an obstacle (red).

2.1.3 Literature Review

The source of uncertainty considered in this work is atmospheric turbulence and stochastic wind gusts, which affect the response [39] and safety [40, 41] of aircraft. These effects are particularly pronounced in relatively lightweight sUAS. Hoblit [42] provides a detailed description of discrete and continuous gust models and Richardson [43] provides a thorough review of the modeling techniques involved. The Dryden and Von Kármán models are the two most common gust/turbulence models [44–48], which are used in both FAA and military specifications. Both consider the linear and angular velocity components of the gusts to be varying stochastic processes and then specify those components' Power Spectral Density (PSD). The Dryden model utilizes rational PSDs, while the Von Kármán model uses irrational PSDs. The latter matches experimental gust observations more closely than the former, but its use of irrational spectral densities prevents its spectral factorization from being exactly expressed.

We also consider the problem of propagating the vehicle's state uncertainty subject to the aforementioned wind gust disturbances. Given an initial probability distribution of a state, the objective of a UQ algorithm is to obtain a characterization of the state's probability distribution at a future instant in time. For this work, we require the algorithm to compute the uncertainty for all time instants between the initial and final times. The conceptually simple solution is through a MC simulation, but the computational intensity of this method limits its usefulness for our application [17,18]. Linearization techniques suffer from diminished accuracy for highly nonlinear systems or for long time horizons, but their simplicity and high computational efficiency make them well-suited for real-time trajectory optimization or path planning [19, 20]. Other nonlinear UQ methods include unscented transformation [21, 22], polynomial chaos expansions [23], and Gaussian mixture models [24–26]. A thorough survey of many other UQ methods is provided by Luo [49], though none of these were deemed appropriate for our use due to our models' large number of states, the presence of stochastic inputs, and the need for rapid computations. For that reason, the linear covariance propagation method is an attractive choice for our problem. The ease of incorporating exogenous disturbances and computing a complete time history of the covariance further distinguish it from the other UQ methods.

There is a wide array of solution strategies for planning a vehicle's path subject to uncertainty, including convex programming [8], mixed integer linear programming [50], graph search [10], fast marching trees [51], and probabilistic roadmaps [9]. One of the more popular approaches [52–56] utilizes a class of stochastic search algorithms called RRT [11]. These algorithms are well suited for real-time implementation [53] and are sampling-based, so they scale well with problem size, but only offer a probabilistic guarantee of

completeness. A comparison of sequential quadratic programming, genetic algorithms, and RRT is provided by Borowski and Frew [56]. Extensions to RRT such as RRT* [57], Informed RRT* [58], and Dynamic RRT* [12], improve optimality of the solution, increase sampling efficiency, and allow for dynamically changing constraints, respectively. These three RRT extensions are utilized in this work.

2.1.4 Original Contributions

The main contribution of this chapter is the amalgamation of existing Dynamic RRT* and Informed RRT* algorithms, and the addition of an obstacle buffer resizing technique, resulting in a single algorithm that allows computationally efficient, chance-constrained path planning for sUAS. In particular, this contribution addresses a dilemma in chance-constrained path planning under uncertainty: trajectory re-planning changes the outcome of the covariance propagation, which, when obstacles or exclusion zones are involved, may require further re-planning.

This chapter builds upon previous work [28] that considered rapid uncertainty propagation, collision detection, and trajectory optimization for fixed-wing, 2D longitudinal flight dynamics. The numerical examples presented here utilize both fixed-wing and quadrotor sUAS in full 3D flight in a non-static atmosphere with inner loop and outer loop controllers in closed form. Other contributions in support of the path planning algorithm are linear covariance propagation of the uncertainty in initial states and exogenous disturbances for three dimensional vehicle motion, and a quadratic programming-based approach to 3D collision detection.

2.2 Modeling

The purpose of this chapter is to present and illustrate with simulations a procedure for rapid uncertainty propagation and chance constrained path planning that is broadly applicable in the sUAS setting in terms of models and uncertainties assumed. This section introduces the two systems used for numerical examples in this work: a quadrotor sUAS model and a fixed-wing sUAS model.

The quadrotor sUAS is modeled as a double integrator with quadratic aerodynamic drag, constant drag coefficient, dynamic extension controller, and a Dryden-based wind disturbance. The full model can be found in Appendix A. The fixed-wing sUAS model is based on a 6-state model for longitudinal and lateral flight in a moving atmosphere, with inner and outer-loop controllers added and a Dryden wind gust model. The full model can

be found in Appendix B.

The above models given by Eqs. (A.2) – (A.11) for the quadrotor or Eqs. (B.27) – (B.32) for the fixed-wing aircraft can be viewed as a single system of differential equations of the form:

$$\dot{X} = f\big(X, X_{des}(t), n(t), \theta\big), \tag{2.1}$$

where θ is the set of vehicle, environmental, and control parameters, n is the white noise input, and the state for the quadrotor model or and for the fixed-wing model, respectively, is:

$$X_{quadrotor} = [r, V_0, \eta_{x1}, \eta_{x2}, \eta_{x3}]^\top \tag{2.2}$$

$$X_{fixed-wing} = [x, y, h, V, \psi, \gamma, T, V_{des}, \psi_{des}, \eta_u, \eta_w, \eta_v]^\top \tag{2.3}$$

2.3 Technical Approach

2.3.1 Covariance Propagation

For the uncertainty propagation analysis, we linearize the sUAS model in Eq. (2.1). The linearized model has the following form:

$$\delta\dot{X} = A(t)\delta X(t) + B_n(t)n(t), \tag{2.4}$$

$$A(t) = \left.\frac{\partial f}{\partial X}\right|_{\substack{X=\bar{X}(t)\\ n=0\\ X_{des}(t)}}, \tag{2.5}$$

$$B_n(t) = \left.\frac{\partial f}{\partial n}\right|_{\substack{X=\bar{X}(t)\\ n=0\\ X_{des}(t)}}. \tag{2.6}$$

Here, $\bar{X}(t)$ is the nominal trajectory obtained by propagating Eq. (2.1) subject to $n = 0$ and specified time history of X_{des}. Physically, this corresponds to the nominal trajectory that the sUAS would fly under no-wind conditions.

Under the assumption that $n(t)$ is a standard white noise process, the state covariance

matrix, P, satisfies the Lyapunov differential equation [59]:

$$\dot{P}(t) = A(t)P(t) + P(t)A^{\top}(t) + B_n(t)B_n^{\top}(t). \tag{2.7}$$

Note that (2.7) can be solved using any standard Ordinary Differential Equation (ODE) solver. The initial condition for P is the zero matrix if the vehicle's initial state is known deterministically, or may be chosen as a diagonal matrix of state covariances, otherwise.

Now, we can use the nominal trajectory $\bar{X}(t)$, covariance matrix $P(t)$, and a Gaussian distribution density function to estimate the probability that the vehicle is contained in a specified set at time t. For the quadrotor model: given the mean $\bar{r}(t) = [\bar{x}_1(t), \bar{x}_2(t), \bar{x}_3(t)]^{\top}$ and the block of the covariance matrix corresponding to these states, $\Sigma_{\omega}(t) = P_{(1:3,1:3)}(t)$, we can build a set from the definition of a probability ellipsoid [60]:

$$Prob\left[\omega \in \mathbb{R}^3 : (\omega - \bar{r}(t))^{\top}\Sigma_{\omega}^{-1}(t)(\omega - \bar{r}(t)) \leq c^2\right] = \beta, \tag{2.8}$$

where $\beta \in [0, 1]$ is a prescribed probability level and c is solved for using the three degree-of-freedom chi-squared distribution [61]. Thus the vehicle has a β probability of being within the set (**??**) at time t.

For verification purposes, Eq. (2.1) is also solved numerically in a series of MC simulations. For these computations, the white Gaussian noise is computed using MATLAB's $randn()$ function, scaled by the inverse square root of the integration time step [62]. A comparison between the linear covariance propagation and the covariance matrix estimated from the MC runs is shown in Section 2.4.

2.3.2 Collision Detection

For trajectory validation purposes, it is desirable to have a collision detection algorithm that can quickly determine whether or not the trajectory tube formed by the set of probability ellipsoids defined in Eqn. (**??**) intersects with an obstacle, represented in this work as a cuboid, as illustrated in Figure 2.5.

The ellipsoid-cuboid intersection at time instant t is formulated as a Quadratic Programming (QP) problem that can be solved by a standard QP solver:

$$\begin{aligned} &\min_{z} \ (z - \bar{r}(t))^{\top}\Sigma_{\omega}^{-1}(t)(z - \bar{r}(t)) \\ &s.t. \\ &A_O z \leq b_O. \end{aligned} \tag{2.9}$$

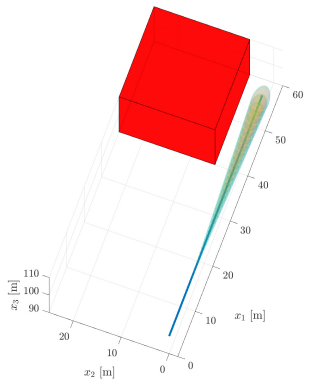

Figure 2.2: $\beta = 0.5$

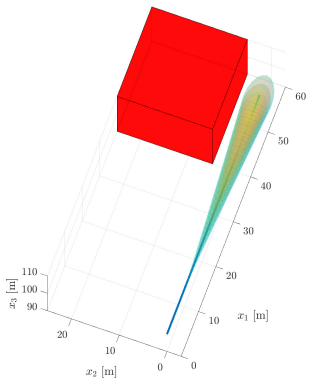

Figure 2.3: $\beta = 0.9$

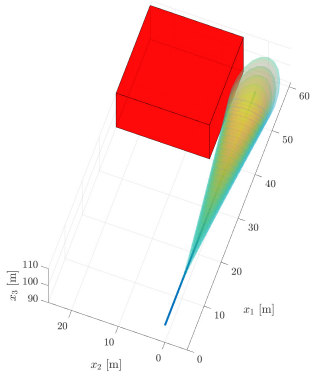

Figure 2.4: $\beta = 0.999$

Figure 2.5: Collision scenario for varying values of β.

Here, z, the optimization parameter, is a concatenation of x_1, x_2 and x_3 coordinates: $z = [x_1, x_2, x_3]^\top$ and $A_O z \leq b_O$ are the linear inequalities that represent the cuboid obstacle. Thus we are solving for the point z that belongs to the smallest ellipsoid centered at $\bar{r}(t)$ that still touches the cuboid obstacle defined by $A_O z \leq b_O$. Let z^* denote the solution to this QP and let $c^{*2}(t) = (z^* - \bar{r}(t))^\top \Sigma_\omega^{-1}(t)(z^* - \bar{r}(t))$. If $c^{*2}(t) \geq c^2$, then there is no intersection between the ellipsoid and cuboid at time t, and the chance constraint is not violated.

For the numerical implementation of this collision detection scheme, the frequency at which collisions are checked is an algorithm tuning parameter, and (5.14) is only solved after it has been determined that the overbounding spheres of the ellipsoid and cuboid intersect. The latter check is computationally very simple.

2.3.3 Path Planning

In our prototypical UTM scenario mentioned in Section 2.1, the covariance propagation and tube generation algorithms discussed in Section 2.3.1 inform the functionality to validate aircraft trajectories. If a desired flight plan is found to be unsafe due to possible collisions with obstacles, the UTM system should return a safe flight path for the UAS to fly. Here, the UAS is operating in a densely populated urban environment with many no-fly zones or obstacles. For scenarios such as this, with a large number of nonconvex obstacle constraints, an RRT algorithm is useful.

This section lays out the modifications made to the standard 2D RRT algorithm [11] for the purpose of generating a desired constant-altitude flight path X_{des}. Note that the path planning is handled in two dimensions, under the assumption that the UAS operates in an urban environment at a constant desired altitude, but all tube generation and collision checks are three-dimensional.

2.3.3.1 Informed Subset

By their nature, RRT* algorithms find optimal paths between the initial state and every state in the search space, regardless of what final state the user is actually interested in. This is an inefficient use of computational resources that the Informed variant [58] attempts to address. By only sampling within an ellipsoidal subset and excluding regions of the sample space that cannot possibly improve the solution, the algorithm makes a better-informed decision about where to look for optimal paths.

This informed subset is defined as follows:

$$Q_{\hat{f}} = \left\{ q \in Q \;\middle|\; \|x_{start} - q\| + \|q - x_{goal}\| \leq c_{best} \right\} \tag{2.10}$$

where c_{best} is the current best solution cost, Q is the global sample space, x_{start} is the initial state and x_{goal} is the desired final state and $\|\cdot\|$ denotes the euclidean distance.

This Informed variant is outlined in Algorithm 2.1 from [58], where $\mathcal{T}$ is the current set of nodes, r_w is the rewiring radius, and N is the total number of iterations. Note that the subroutine $SampleEllipse(x_{start}, x_{goal}, c_{best})$ returns a randomly sampled point that is contained within the ellipse defined in (2.10), and lines 6–17 in Algorithm 2.2 include the rewiring steps essential to the RRT* algorithm variant.

Algorithm 2.1 $\mathcal{T}$ = Informed RRT*(x_0, x_f)

1: $\mathcal{T}_0 \leftarrow x_0$
2: $c_{best} \leftarrow \infty$
3: **for** $k = 1 \ldots N$ **do**
4: $\quad \mathcal{T}_{k+1} = AddNode(\mathcal{T}_k)$
5: **end for**
6: **return** $\mathcal{T}_N$

Algorithm 2.2 $\mathcal{T}_{k+1} = AddNode(\mathcal{T}_k)$

$x_{rand} \leftarrow SampleEllipse(x_{start}, x_{goal}, c_{best})$; % c_{best} is cost of current best solution
$x_{nearest} \leftarrow Nearest(\mathcal{T}, x_{rand})$;
$x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$;
if $NoCollision(x_{nearest}, x_{new})$ **then**
    $\mathcal{T} \leftarrow \mathcal{T} + x_{new}$;
    $X_{near} \leftarrow Near(\mathcal{T}, x_{new}, r_w)$;
    $x_{min} \leftarrow x_{nearest}$;
    $c_{min} \leftarrow Cost(x_{min}) + ||coordinates(x_{nearest}) - coordinates(x_{new})||$
    for $\forall x_{near} \in X_{nearest}$ **do**
        $c_{new} \leftarrow Cost(x_{near}) + ||coordinates(x_{near}) - coordinates(x_{new})||$
        if $c_{new} < c_{min}$ **then**
            if $NoCollision(x_{nearest}, x_{new})$ **then**
                $x_{min} \leftarrow x_{near}$
                $c_{min} \leftarrow c_{new}$
            end if
        end if
    end for
    $Parent(x_{new}) \leftarrow x_{min}$
end if
return $\mathcal{T}$

2.3.3.2 Chance Constraints and Dynamic Extension

If the solution of Algorithm 2.1 was passed to the tube generation algorithm as X_{des}, there would be no guarantee that the generated tube would not intersect the obstacles. The solution to a path planning problem with obstacles often passes very closely to the obstacle, where any nonzero uncertainty in the state would produce a tube that would intersect the obstacle. Thus a buffer region is added to the obstacles and adjusted periodically in a manner informed by full covariance propagation.

This buffer can be seen in Figure 2.6, which represents a single outer-loop iteration in which the buffer is held constant. Here, the green triangle is the starting point x_{start}, the blue circle is the end point x_{goal}, the green line is the current best solution, the dashed red ellipse is the current informed subset as defined in Eqn. (2.10), and the black lines represent the total set of sampled nodes and connecting vertices. The solid red rectangle is the actual obstacle, while the dashed red rectangle is the buffered obstacle that is actually passed to Algorithm 2.1 and used in the subroutine $NoCollision(x_{nearest}, x_{new})$. The bottom subfigure is the same as the top, but with the covariance tube as defined by (**??**) overlaid, illustrating how the buffer prevents the tube from intersecting the true obstacle.

The dynamic extension to this algorithm involves intermittently halting the standard

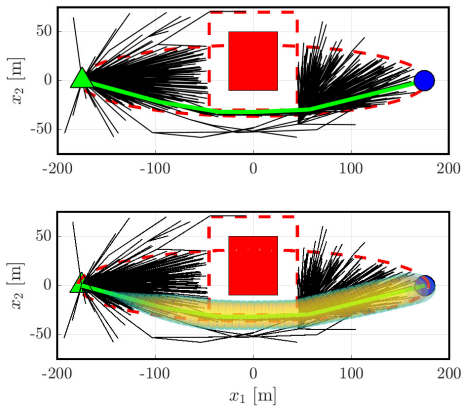

Figure 2.6: Results of Algorithm 2.1 with a buffered obstacle.

RRT algorithm, propagating the nominal vehicle trajectory and covariance, and adjusting the obstacle buffer sizes accordingly. This requires an algorithm that informs the buffer size adjustments, answering the question: "By how much should we grow or shrink the buffer such that the covariance tube just touches the obstacle?". To answer this, we start with the easier question of "What size ellipsoid just touches the obstacle?" This is illustrated in Figure 2.7 and can be expressed as a QP:

$$\min_{z}(z-\bar{r}(t))^{\top}\Sigma_r^{-1}(t)(z-\bar{r}(t)) \tag{2.11}$$

$$s.t.$$

$$A_O z \leq b_O. \tag{2.12}$$

We alter (2.12) to take into account the size of the buffer as follows:

$$A_O z \leq b_O + ld \tag{2.13}$$

where l is a vector of ones and d is a scalar representing the size of the buffer. We can then vary d until $c^{2*} = c^2$, which solves for the buffer size such that the smallest ellipsoid that intersects the buffer is the same size as the actual covariance ellipsoid. This solution is used as the buffer size for the next iteration. Note that this new buffer size is only an approximation of the actual solution since the size of the covariance tube is dependent on the solution to the inner loop, X_{des}, which changes at every iteration.

Letting l be a vector of all ones has the effect of buffering the obstacle by an equal

amount in all directions, as opposed to extending the buffer only in certain directions. Equal buffering on all sides of the obstacle may result in less optimal trajectories in a situation where the vehicle's trajectory interacts with multiple edges of an obstacle and this is viewed as the trade-off for not including the computationally nontrivial logic for a more discriminating buffer re-sizing. Additionally, expanding the buffer in all directions reduces the RRT search space more than merely expanding in one direction, which aids convergence.

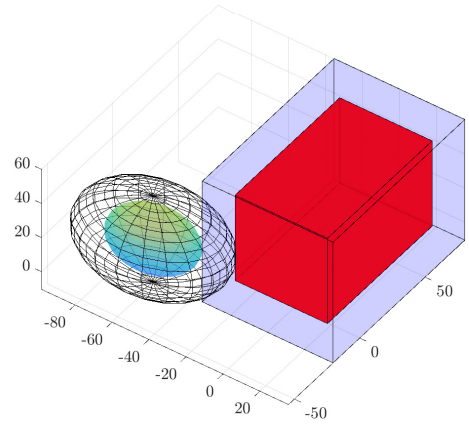

Figure 2.7: Cuboid buffer expansion illustration. The shaded ellipsoid and red cuboid represent the original confidence ellipsoid and obstacle, respectively. The wire mesh ellipsoid and blue cuboid represent the expanded sets that just touch the original obstacle and confidence ellipsoid, respectively.

Because the size of the probability ellipsoids vary spatially, the optimal buffer size is not necessarily the same for every obstacle. Thus, buffer changes happen independently for every obstacle. There are two cases to be considered: the buffer shrinks or stays the same, and the buffer grows. If a buffer is shrunken or left unchanged, no RRT nodes are changed, and the algorithm continues as normal. However, if a buffer grows, the nodes and connecting vertices that are now contained within the newly-sized obstacle are no longer valid. They are thus deleted, leaving only the parent trees and the now-stranded branches. Then, the Informed RRT* algorithm is used to grow branches from the parent tree until any part of the stranded branch is re-connected. At this point, any nodes left without a parent are removed from the set of nodes, and the Informed RRT* algorithm continues to run according to Algorithm 2.1 until the next covariance propagation and buffer resize. This is laid out in more detail in Algorithm 2.3.

Algorithm 2.3 $\mathcal{T} =$ Dynamic Informed RRT*(x_0, x_f)

$\mathcal{T}_0 \leftarrow x_0$
$c_{best} \leftarrow \infty$
$\delta \leftarrow \infty$
for $m = 1 \dots M$ **do**
    $k = 0$
    while $k < N_{max}$ AND $\delta > tol$ **do**
        $k = k + 1$
        $\mathcal{T}_k = AddNode(\mathcal{T}_{k-1})$
        $cost_i = FindBestPath(\mathcal{T}_i)$
        if $i > N_{conv}$ **then**
            $\delta = |\frac{cost_i - cost_{i-N_{conv}}}{cost_{i-N_{conv}}}|$
        end if
    end while
    if $m \neq M$ **then**
        $d = CompObsDist(\mathcal{T}_k, O)$ % Compute distance between obstacle and covariance tube
        for $j = 1 \dots N_{obs}$ **do**
            if $d_j \geq 0$ **then**
                $b_j = b_j - d_j$
            else if $d_j < 0$ **then**
                $b_j = b_j - d_j$
                $\mathcal{T} = CleanupNodes(\mathcal{T}, O_j)$ % remove nodes that are no longer valid
                $\mathcal{T} = Regrow(\mathcal{T}, O_j, params)$ % regrow until orphaned nodes are reconnected or pruned
            end if
        end for
    end if
end for
return $\mathcal{T}$

2.4 Results

2.4.1 Trajectory Validation

The first result of interest is a validation of the LC propagation described in Section 2.3.1 versus the results of the MC simulations. For both the fixed-wing and quadrotor models, a simple desired trajectory was prescribed, and the state covariance computed from a 10,000 run MC simulation is compared to the LC results computed from Eqn. (2.7). For the quadrotor model, the desired trajectory is a three phase ascent-cruise-descent mission profile. For the fixed-wing model, it is a constant-altitude trajectory with a sinusoid shape in the lateral direction.

The comparison results are shown in Figures 2.8 – 2.11. Here, $\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$, $\bar{x}$, $\bar{y}$, and $\bar{h}$ represent either the mean position, or the nominal trajectory $\bar{X}(t)$, from the MC and LC simulations, respectively. All plots show good agreement between MC and LC, with the largest deviations being $\approx 15\%$ in the σ^2_{x2} channel in Figure 2.8. This justifies our use of LC propagation in path planning under uncertainty. Note that the deviations in $\bar{x}_2$ appear large in the plot, but the vertical axis range is quite small. Also note that the nominal trajectory for the fixed-wing example does not track the desired trajectory exactly, but that is a function of the model's controller, which is not the focus of this work, as explained in Section 3.2.

All computations are performed in the MATLAB environment running on a dual-core Intel Core i5 at 2.7 GHz, and all confidence ellipsoid tubes use $\beta = 0.999$. This value was chosen for illustration purposes, though some UTM applications may call for smaller or larger values. For the quadrotor model, the LC propagation took $0.062s$ of computation time for $40s$ of simulated flight time. For the fixed-wing model, it took $0.18s$ of computation time for $35s$ of simulated flight time.

2.4.2 Path Planning

Now that we have shown that the LC propagation scheme is sufficiently fast and accurate when compared to MC simulations of the nonlinear system, we can use the LC propagation algorithm to enable chance-constrained path planning as laid out in Section 2.3.3. For brevity, only the quadrotor model is utilized in this section.

Figures 2.12 and 2.13 show the results of Algorithm 2.3 for a three obstacle scenario. A solution is found that is very close to the minimum path length between x_{start} and x_{goal}, while still ensuring that the β confidence ellipsoid tube does not intersect any of the obstacles. Total computation time for this example was $30.5s$.

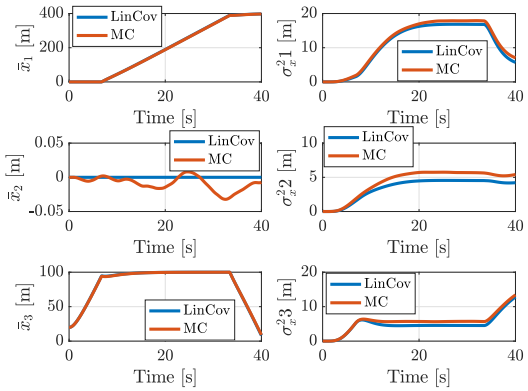

Figure 2.8: State and covariance time histories for prescribed quadrotor trajectory.

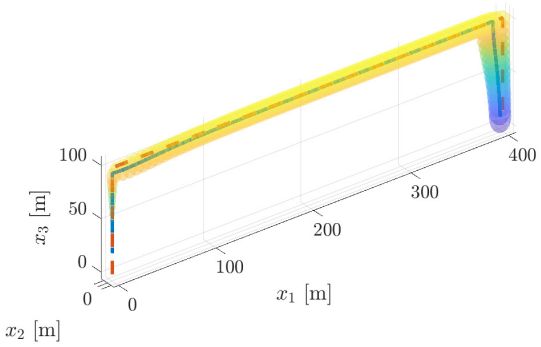

Figure 2.9: Quadrotor 3D trajectory. The covariance tube is constructed from a set of confidence ellipsoids defined by (**??**), the dashed line is the desired trajectory, and the solid line is the nominal trajectory $\bar{X}(t)$.

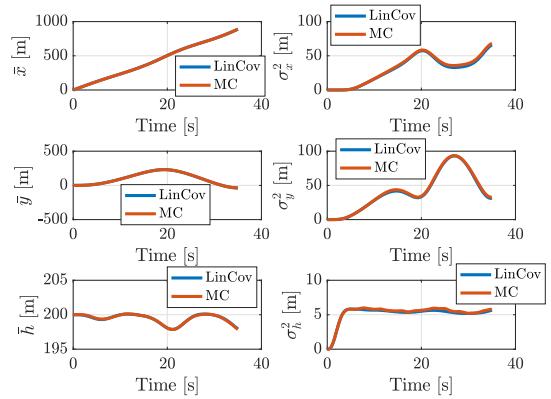

Figure 2.10: State and covariance time histories for prescribed fixed-wing trajectory.

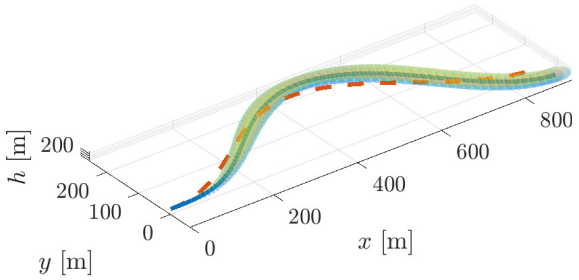

Figure 2.11: Fixed-wing sUAS 3D trajectory. The covariance tube is constructed from a set of confidence ellipsoids defined by (**??**), the dashed line is the desired trajectory, and the solid line is the nominal trajectory $\bar{X}(t)$.

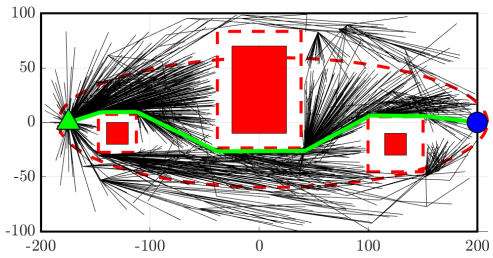

Figure 2.12: Chance-constrained Informed Dynamic RRT* solution for quadrotor model with three obstacles.

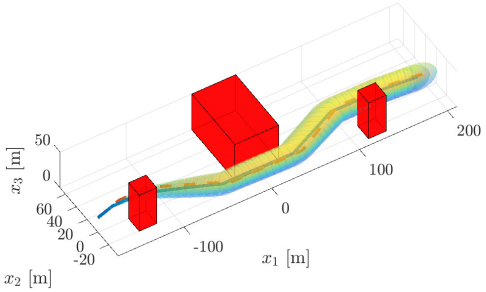

Figure 2.13: 3D plot of obstacles and flight path with β confidence tube overlaid.

Note that the buffer re-sizing algorithm converges, but only to a region, the size of which is dictated by tol. There is no guarantee that the algorithm will exit without a constraint violation, but, based on extensive simulation studies, the magnitude of the violation will be sufficiently small for sufficiently large M and small tol.

2.5 Conclusions

Future UTM systems will need to rely on rapid UQ for trajectory validation and path planning in order to account for the effects of wind turbulence/gusts and other uncertainties and disturbances. This work combines existing Dynamic RRT* and Informed RRT* algorithms, and adds an obstacle buffer resizing technique to solve a challenge of chance-constrained path planning: trajectory re-planning changes the outcome of the covariance propagation. The results presented here shows that this RRT*-based algorithm, in combination with QP-based collision checking for trajectory validation, successfully solves the aforementioned problem, resulting in a computationally efficient chance-constrained path planner. Trajectory validation examples were presented for both quadrotor and fixed-wing models in 3D flight in a non-static atmosphere. Path planning examples were presented for the quadrotor model, showing near-optimal navigation around 3 obstacles while enforcing chance state constraints.

CHAPTER 3

Spacecraft Relative Motion Planning Using Chained Chance-Constrained Admissible Sets

3.1 Introduction

As of April 2019, the U.S. Space Surveillance Network was tracking 19,404 pieces of orbital debris [63]. Two major contributors to this debris are the 2007 Chinese anti-satellite missile test and the 2009 Iridium-Kosmos satellite collision, though other minor collisions contribute to the debris count yearly [64]. The need to operate satellites safely in the presence of this orbital debris, as well as other operational satellites, motivates the development of relative motion planning algorithms that include nonconvex obstacle avoidance constraints. Other mission considerations include handling modeling uncertainties and measurement uncertainties while relying on the limited computational capabilities of many spacecraft.

Spacecraft Relative Motion Planning (SRMP) is concerned with the design of orbital maneuvers with respect to a reference point on a nominal orbit. To handle nonconvex obstacle avoidance constraints, one approach involves solving a sequence of convexified problems that eventually recovers the original optimal solution [65]. Richards et al. proposes a framework in which the fuel-optimal spacecraft trajectory optimization problem subject to avoidance constraints is expressed as a mixed-integer linear program [66]. A MPC approach for rendezvous and proximity operation is presented in [67], while [68] approaches the SRMP problem with a computationally efficient, sampling-based algorithm. A comprehensive survey of spacecraft formation flying can be found in [69].

A graph search framework for SRMP proposed in [30, 70, 71] benefits from the computational efficiency and simplicity of algorithms such as Dijkstra's [72] and A^* [73] search. The approach involves building a connectivity graph for a set of forced equilibria or natural motion trajectories and the use of safe, positively invariant sets to determine connectiv-

ity between graph vertices. The resulting motion planning framework can accommodate obstacles and bounded disturbances.

The developments in [30] are based on assumptions of full state measurement and set bounded disturbances. Under these conditions, positively invariant sets are constructed around forced equilibria, which guarantee that the closed-loop response satisfies the constraints for any initial condition in this set when the selected equilibrium reference command is the one corresponding to this set. A safe transition between two forced equilibria can be accomplished if the first equilibrium is in the interior of the positively invariant set for the second equilibrium. Such forced equilibria are then treated as vertices in a directed graph (a virtual net in the terminology of [30]) and are connected by an edge. Based on the family of such equilibria, spacecraft motion planning reduces to a graph search for the sequence of the equilibria to hop between to arrive at the target equilibrium while minimizing suitably constructed cumulative transition cost. This approach is extended in [70] to include periodic natural motion trajectories, in [74] to include non-periodic trajectories, and in [71] to handle set bounded disturbances and minimum thrust constraints. Related ideas have been explored for the development of motion planners for self-driving cars in [75, 76].

In this chapter, we consider the case when the system model is linear, the full state measurement is not available and the measured outputs are contaminated by random Gaussian measurement noise. In addition, the system is affected by random Gaussian disturbances which could represent the effect of unknown forces such as thrust errors or other perturbations acting on the spacecraft. In this setting, an observer is introduced to estimate the state, and state constraints are imposed as chance (probabilistic) constraints. Note that hard constraints cannot be enforced for all times as disturbances and measurement noise values are not compactly supported. For this setting, chance constrained admissible sets are defined as sets of initial state estimates such that the chance constraints hold for all future times for the given constant reference. As the chance constraints are dependent on the initial estimation error covariance matrix, a simplifying assumption is made–the observer has reached steady-state and hence this matrix is equal to the steady-state error covariance matrix. Unlike [30], where positive invariant sets being chained are sublevel sets of Lyapunov functions, here we exploit chance constrained admissible sets to determine connectivity of forced equilibria. These sets are near maximal and hence allow more vertices in the graph to be connected. Furthermore, our approach is able to accommodate stochastic measurement and process noise, as opposed to bounded disturbances as in [30].

We also propose a novel approach of handling non-convex constraints by exploiting their inner approximation with a union of convex constraints and we extend the connectivity conditions to this case. Further we demonstrate that the chance constraint used as a

requirement in the construction of chance constrained admissible sets holds for the closed-loop trajectories with switching between forced equilibria.

This chapter is organized as follows. Section 3.2 describes the relative motion dynamics and disturbance model. Section 3.3 describes the chance constrained admissible sets and how to construct them. Section 3.4 describes the construction of the virtual net and its extension for obstacle avoidance, while Section 3.5 presents some numerical examples.

3.2 Modeling

Consider a system with a linear discrete-time model given by

$$x_{k+1} = Ax_k + Bu_k + \Gamma w_k, \tag{3.1}$$

$$y_k = Cx_k + Fv_k, \tag{3.2}$$

where x_k is an n_x-vector state, u_k is an n_u-vector control input, w_k is an n_w-vector disturbance, y_k is an n_y-vector measured output, and v_k is an n_v-vector measurement noise. We make the following assumptions:

1. The disturbance and measurement noise sequences are independent and identically distributed Gaussian processes with zero mean and unit covariance matrix, i.e.,

$$w_k \sim \mathcal{N}(0, \mathbb{I}), \qquad v_k \sim \mathcal{N}(0, \mathbb{I}). \tag{3.3}$$

2. The variables $x_0, \{w_k\}_{k\in\mathbb{Z}_{\geq 0}}$, and $\{v_k\}_{k\in\mathbb{Z}_{\geq 0}}$ are independent.

In this chapter focused on SRMP, the nominal dynamics are represented by the linearized Clohessy-Wiltshire (CW) equations [77], which describe the motion of a chase spacecraft relative to a target spacecraft orbiting a central body in a circular orbit. The continuous-time CW equations are given by:

$$\dot{x} = A_{ct}x + B_{ct}u, \tag{3.4}$$

$$A_{ct} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \tag{3.5}$$

$$B_{ct} = \begin{bmatrix} 0_{3,3} \\ \mathbb{I}_3 \end{bmatrix},\ n = 0.00116\frac{\text{rad}}{\text{s}}, \tag{3.6}$$

where $x = \begin{bmatrix} x_1 & x_2 & x_3 & \dot{x}_1 & \dot{x}_2 & \dot{x}_3 \end{bmatrix}^\intercal$, and the value for n used here corresponds to an orbital period of 90 minutes, or LEO. The x_1 axis is along the direction from the central body to the target spacecraft, the x_3 axis is along its angular momentum vector, and the x_2 axis completes the right-handed reference frame. Discretizing these equations with a zero order hold and sampling period of ΔT results in (3.1)-(3.2) with:

$$A = e^{A_{ct}\Delta T},\ B = \int_0^{\Delta T} e^{A_{ct}(\Delta T - \tau)} d\tau B_{ct}, \tag{3.7}$$

and with the process noise and sensor noise matrices, Γ and F, and the output matrix C defined as:

$$\Gamma = \frac{1}{100}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ F = \frac{1}{100}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.8}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.9}$$

For state estimation, a Luenberger observer of the following form is augmented:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(\hat{y}_k - y_k), \tag{3.10}$$

$$= A\hat{x}_k + Bu_k + L(C\hat{x}_k - Cx_k - Fv_k). \tag{3.11}$$

We also consider a nominal feedback control law of the form:

$$u_k = K\hat{x}_k + Gr, \tag{3.12}$$

where r is the set-point. Here, L and K are any stabilizing gain matrices and G is computed as

$$G = \left(C(\mathbb{I} - A - BK)^{-1}B\right)^{-1}, \tag{3.13}$$

so that $y = r$ in steady-state in the disturbance free case.

3.3 Chance Constrained Admissible Sets

3.3.1 Covariance Computation

Define the estimation error as

$$e_k = x_k - \hat{x}_k. \tag{3.14}$$

Then the estimation error dynamics are represented by the following equations,

$$e_{k+1} = Ae_k + \Gamma w_k + LCe_k + LFv_k \tag{3.15}$$

$$= A_o e_k + B_o \begin{bmatrix} w_k \\ v_k \end{bmatrix}, \tag{3.16}$$

where

$$A_o = (A + LC), \quad B_o = \begin{bmatrix} \Gamma & LF \end{bmatrix}. \tag{3.17}$$

The matrix A_o is assumed to be Schur (all eigenvalues are inside the unit disk).

The evolution of the state estimate, $\hat{x}_k$, is determined from (3.11), (3.12) and (3.14) by

$$\hat{x}_{k+1} = A_c\hat{x}_k + B_c r + \Gamma_c \begin{bmatrix} e_k \\ v_k \end{bmatrix}, \tag{3.18}$$

where

$$A_c = (A + BK), \tag{3.19}$$

$$B_c = BG, \tag{3.20}$$

$$\Gamma_c = \begin{bmatrix} -LC & -LF \end{bmatrix}. \tag{3.21}$$

The control gain K is assumed to be stabilizing so that the matrix A_c is Schur.

Let

$$\tilde{x}_k = \hat{x}_k - (\mathbb{I} - A_c)^{-1} B_c r. \tag{3.22}$$

Then

$$\tilde{x}_{k+1} = A_c \tilde{x}_k + \Gamma_c \begin{bmatrix} e_k \\ v_k \end{bmatrix}, \tag{3.23}$$

$$x_k = \hat{x}_k + e_k = \tilde{x}_k + e_k + (\mathbb{I} - A_c)^{-1} B_c r. \tag{3.24}$$

If the closed-loop system (including the plant and the observer) has operated for a sufficiently long period of time, the observer error dynamics (3.15) can be assumed to be in steady-state and the error to be normally distributed with zero mean and steady-state covariance matrix, $P_\infty \succeq 0$, satisfying

$$P_\infty = A_o P_\infty A_o^\intercal + B_o B_o^\intercal. \tag{3.25}$$

That is, $e_k \sim \mathcal{N}(0, P_\infty)$ for all k. Note that since A_o is Schur, the error covariance matrix at time instant k, $P_k \to P_\infty$ as $k \to \infty$ [78].

3.3.2 Chance Constraints

Consider now enforcing a state constraint of the form

$$Hx_k \leq h, \forall k, \tag{3.26}$$

where h is an n_h-vector. This constraint can be written using (3.14) and (3.22) as:

$$H\tilde{x}_k + He_k \leq h - H(\mathbb{I} - A_c)^{-1} B_c r. \tag{3.27}$$

We now consider approaches to enforce the constraint (3.26) with probabilistic guarantees based on the model (3.15) and (3.23). Re-stating the model and the constraint for

convenience here, we have

$$\begin{bmatrix} \tilde{x}_{k+1} \\ e_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_c & -LC \\ 0 & A_o \end{bmatrix}}_{=:A_{\text{aug}}} \begin{bmatrix} \tilde{x}_k \\ e_k \end{bmatrix} + \underbrace{\begin{bmatrix} -LF \\ LF \end{bmatrix}}_{=:B_{\text{aug}}} v_k + \underbrace{\begin{bmatrix} 0 \\ \Gamma \end{bmatrix}}_{=:\Gamma_{\text{aug}}} w_k, \tag{3.28}$$

$$H\tilde{x}_k + He_k \leq h - H(\mathbb{I} - A_c)^{-1}B_c r, \tag{3.29}$$

where

$$v_k \sim \mathcal{N}(0, \mathbb{I}), \qquad w_k \sim \mathcal{N}(0, \mathbb{I}). \tag{3.30}$$

Let k be the current time instant, and consider $t \geq 0$ to be running time over the prediction horizon. Denote by $\tilde{x}_{t|k}$ the predicted value of $\tilde{x}_{k+t}$ and by $e_{t|k}$ the predicted value of e_{k+t}. The dynamics of $\left[\tilde{x}_{t|k},\, e_{t|k}\right]^\intercal$ are given as

$$\begin{bmatrix} \tilde{x}_{t+1|k} \\ e_{t+1|k} \end{bmatrix} = A_{\text{aug}} \begin{bmatrix} \tilde{x}_{t|k} \\ e_{t|k} \end{bmatrix} + B_{\text{aug}} v_{k+t} + \Gamma_{\text{aug}} w_{k+t}. \tag{3.31}$$

We can predict the time-varying covariance matrix of $\left[\tilde{x}_{t|k},\, e_{t|k}\right]^\intercal$ using

$$\tilde{P}_{t+1|k} = A_{\text{aug}}\tilde{P}_{t|k}A_{\text{aug}}^\intercal + B_{\text{aug}}B_{\text{aug}}^\intercal + \underbrace{\Gamma_{\text{aug}}\Gamma_{\text{aug}}^\intercal}_{=\begin{bmatrix} 0 & 0 \\ 0 & \Gamma\Gamma^\intercal \end{bmatrix}}, \tag{3.32}$$

where

$$\tilde{P}_{0|k} = \begin{bmatrix} 0 & 0 \\ 0 & P_\infty \end{bmatrix}. \tag{3.33}$$

Note that $\tilde{x}_{0|k} = \tilde{x}_k$, as the observer output, is measured, thus, $\text{cov}(\tilde{x}_{0|k}, \tilde{x}_{0|k}) = \text{cov}(\tilde{x}_{0|k}, e_{0|k}) = 0$. We assume that $\text{cov}(e_{0|k}, e_{0|k}) = P_\infty$ for all k, where P_∞ is defined in (3.25), which is a reasonable approximation if the closed-loop system including the observer has operated for a sufficiently long period of time.

Then,

$$\begin{bmatrix} \tilde{x}_{t|k} \\ e_{t|k} \end{bmatrix} \sim \mathcal{N}\left(A_{\text{aug}}^t \begin{bmatrix} \tilde{x}_k \\ 0 \end{bmatrix}, \tilde{P}_{t|k} \right), \tag{3.34}$$

so

$$\underbrace{\begin{bmatrix} H & H \end{bmatrix}}_{=:H_{\text{aug}}} \begin{bmatrix} \tilde{x}_{t|k} \\ e_{t|k} \end{bmatrix} \sim \mathcal{N}\Bigg(H_{\text{aug}} A_{\text{aug}}^t \begin{bmatrix} \tilde{x}_k \\ 0 \end{bmatrix}, \underbrace{H_{\text{aug}} \tilde{P}_{t|k} H_{\text{aug}}^\intercal}_{=:\Sigma_{t|k}} \Bigg). \tag{3.35}$$

Due to the fact that the disturbance and noise signals are unbounded, it is in general not possible to enforce the constraint (3.26) for all possible realizations of disturbance and noise sequences. Therefore, we instead consider a chance constraint imposed over the prediction horizon of the form,

$$\text{Prob}\{Hx_{t|k} \leq h\} \geq 1 - \alpha, \quad t \geq 0, \tag{3.36}$$

where $0 < \alpha < 1$. This chance constraint can be re-stated as

$$\text{Prob}\left\{ H_{\text{aug}} \begin{bmatrix} \tilde{x}_{t|k} \\ e_{t|k} \end{bmatrix} \leq h - H(\mathbb{I} - A_c)^{-1} B_c r \right\} \geq 1 - \alpha, \quad t \geq 0. \tag{3.37}$$

Assume for the moment that the constraint is scalar, $n_h = 1$ (this assumption will be relaxed using a risk allocation approach later in this section). In this case, the constraint (3.37) can be re-stated as

$$H_{\text{aug}} A_{\text{aug}}^t \begin{bmatrix} \tilde{x}_k \\ 0 \end{bmatrix} \leq h - H(\mathbb{I} - A_c)^{-1} B_c r - \sqrt{2\Sigma_{t|k}}\, \text{erf}^{-1}(1 - 2\alpha), \tag{3.38}$$

where erf is the error function [79],

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{3.39}$$

Motivated by the above considerations, define the chance constrained admissible set,

$\tilde{O}_\infty(r)$, as

$$\tilde{O}_\infty(r) = \Bigg\{ \tilde{x}_0 : H_{\text{aug}} A_{\text{aug}}^t \begin{bmatrix} \tilde{x}_0 \\ 0 \end{bmatrix} \leq h(r) - \sqrt{2\Sigma_{t|0}}\, \text{erf}^{-1}(1-2\alpha),\ t \geq 0 \Bigg\}, \tag{3.40}$$

where

$$h(r) = h - H(\mathbb{I} - A_c)^{-1} B_c r. \tag{3.41}$$

For numerical implementation, $\tilde{O}_\infty(r)$ is constructed as described in [80]. For $n_h \geq 1$, let H_i denote the ith row of H and h_i denote the ith entry of h. Based on Boole's inequality, the chance constraint (3.36) can be satisfied by enforcing the following set of constraints:

$$\text{Prob}\{H_i x_{t|k} \leq h_i\} \geq 1 - \alpha', \tag{3.42}$$

for all $i = 1, ..., n_h$, where $\alpha' = \frac{\alpha}{n_h}$. Then, we define $\tilde{O}_\infty(r)$ as

$$\tilde{O}_\infty(r) = \bigcap_{i=1}^{n_h} \tilde{O}_{\infty,i}(r), \tag{3.43}$$

where $\tilde{O}_{\infty,i}(r)$ is defined using (3.40) with H, h, and α replaced by, respectively, H_i, h_i, and α'.

3.4 Virtual Net

3.4.1 Graph Construction

In its simplest form, the virtual net constrained motion planning framework exploits a discrete set of set-points, $\mathcal{R} = \{r^1,\ r^2, \cdots, r^{n_r}\}$, and reduces the trajectory design problem to an online graph search for the path in this set of set-points. Once the path is determined through the graph search, an online switching logic is used to execute the path whereby a switch from one set-point to the next is effected when suitable switching conditions are satisfied.

The virtual net guarantees safety (constraint enforcement) by declaring that a connection (edge) between set-points r^i and r^j exists if

$$(\mathbb{I} - A_c)^{-1} B_c (r^i - r^j) \in int\big(\tilde{O}_\infty(r^j)\big), \tag{3.44}$$

i.e., a connection between set-points r^i and r^j exists if the vector from the disturbance-free equilibrium of r^j to the disturbance-free equilibrium of r^i lies within the chance-constrained admissible set of r^j. This ensures that there will necessarily be some k for which a safe switch between set-points r^i and r^j is possible. Specifically, suppose the system has been operating with the set-point r^i for a while. Then the dynamics of $\tilde{x}_k$ have been evolving according to (3.23) with $r = r^i$. Under the assumptions made, $\hat{x}_k$ will enter an arbitrary small neighborhood of the disturbance-free equilibrium, $(\mathbb{I} - A_c)^{-1}B_c r^i$, for some k. In particular, the condition,

$$\hat{x}_k - (\mathbb{I} - A_c)^{-1}B_c r^j \in \tilde{O}_\infty(r^j) \tag{3.45}$$

is guaranteed to hold (with probability 1) for some k. If this condition holds, then the switch of the set-point r^i to r^j can be effected at the time instant k while guaranteeing that if the set-point r_k is maintained at r^j for all the subsequent time instants, the constraint (3.38) and hence the chance constraint (3.36) will be satisfied. These properties of the framework will be formally presented as Propositions 1 and 2.

The online switching controller monitors the state estimate, $\hat{x}_k$, and checks whether for the next set-point in the path, r^+, the switching condition,

$$\hat{x}_k - (\mathbb{I} - A_c)^{-1}B_c r^+ \in \tilde{O}_\infty(r^+), \tag{3.46}$$

is satisfied. Once (3.46) holds, the switch $r_k \leftarrow r^+$ is made.

With the goal of generating fuel-efficient trajectories, the weight of the edge from node r^i to node r^j, $\mathcal{G}(i,j)$ is defined as the approximate amount of fuel needed for the spacecraft to travel from node r^i to node r^j. This amount of fuel is estimated based on (3.1) with control (3.12) propagated subject to zero disturbance and perfect observations, i.e., $\Gamma = 0$ and $\hat{x} = x$. The initial condition is set as $x_0 = \begin{bmatrix} r^i \\ 0 \end{bmatrix}$ and the reference is set as $r = r^j$. The simulations are run until

$$\left|(\mathbb{I} - A_c)^{-1}B_c r^j - x_k\right|_2 \leq 0.05 \left|(\mathbb{I} - A_c)^{-1}B_c (r^j - r^i)\right|_2 \tag{3.47}$$

and the edge weighting is set as

$$\mathcal{G}(i,j) = \sum_{l=0}^{k} |u_l|_2 \,. \tag{3.48}$$

Proposition 1: Suppose that the initial pair of state estimate and set-point $(\hat{x}_0, r_0)$ satisfies $\hat{x}_0 - (\mathbb{I} - A_c)^{-1}B_c r_0 \in \tilde{O}_\infty(r_0)$, and $w_k, v_k \sim \mathcal{N}(0, \mathbb{I})$, $e_{0|k} \sim \mathcal{N}(0, P_\infty)$ for all $k \geq 0$. Further, suppose that all of the set-point switches are made when the switching condition (45) is satisfied. Then, the probability of satisfying the constraint (24) is no less than $1 - \alpha$, i.e., Prob$(Hx_k \leq h) \geq 1 - \alpha$, for all $k \geq 0$.

Proof: For any $k \geq 0$, let $k' = \max\big(\{t \,|\, 1 \leq t \leq k, r_t \neq r_{t-1}\} \cup \{0\}\big)$. Note that k' is a random variable. According to the definition of k' and the set-point switching condition (3.46), for any realization of k', the corresponding set of realizations of state estimate and set-point trajectory $\{(\hat{x}_t, r_t)\}_{t=0}^{k}$ must all satisfy $\hat{x}_{k'} - (\mathbb{I} - A_c)^{-1}B_c r_{k'} \in \tilde{O}_\infty(r_{k'})$ and $r_t = r_{k'}$ for all $t = k', \ldots, k$. Then, by the definition of $\tilde{O}_\infty(r_{k'})$, the conditional probability measure of the subset of trajectories satisfying $Hx_k \leq h$ must be no less than $1 - \alpha$, i.e., Prob$(Hx_k \leq h \,|\, k') \geq 1 - \alpha$, where Prob$(\cdot \,|\, k')$ denotes the probability measure conditioned on the realized k'. Then, using the formula of total probability, we obtain Prob$(Hx_k \leq h) = \sum_{k'=0}^{k}$ Prob$(Hx_k \leq h \,|\, k')$Prob$(k') \geq (1-\alpha)\sum_{k'=0}^{k}$ Prob$(k') = 1-\alpha$, since $\sum_{k'=0}^{k}$ Prob$(k') = 1$. ■

Proposition 2: For a path determined by the graph search algorithm, as a sequence of set-points $\{r^0, r^1, ..., r^f\}$ satisfying

$$(\mathbb{I} - A_c)^{-1}B_c(r^{i-1} - r^i) \in int\big(\tilde{O}_\infty(r^i)\big)$$

for all $i = 1, ..., f$, suppose that $r_0 = r^0$ and set-point switches $r_k \leftarrow r^+$ are made when the switching condition (3.46) is satisfied. Then, there almost surely exists $k^f \in \mathbb{N}$ such that $r_k = r^f$ for all $k \geq k^f$, i.e., the terminal set-point of the path r^f, as the reference point for the spacecraft to track, is reached by r_k in finite time.

Proof: For any $i = 1, ..., f$, since $(\mathbb{I} - A_c)^{-1}B_c(r^{i-1} - r^i) \in int\big(\tilde{O}_\infty(r^i)\big)$, there exists an open set U containing 0 such that $U + (\mathbb{I} - A_c)^{-1}B_c r^{i-1} \subset \tilde{O}_\infty(r^i) + (\mathbb{I} - A_c)^{-1}B_c r^i$. Considering the system (3.23), by the fact that A_c is strictly Schur, for any initial condition $\tilde{x}_0$, there almost surely exists $k' \in \mathbb{N}$ such that $\tilde{x}_{k'} \in U$ [80]. This implies that if $r_k = r^{i-1}$ for a sufficiently long period of time, there almost surely exists $k' \in \mathbb{N}$ such that $\hat{x}_{k'} \in U + (\mathbb{I} - A_c)^{-1}B_c r^{i-1} \subset \tilde{O}_\infty(r^i) + (\mathbb{I} - A_c)^{-1}B_c r^i$, where the set-point switching condition (3.46) is satisfied and thus $r_{k'} \leftarrow r^i$. Then, the statement of Proposition 2 follows from the fact that the above result holds for all $i = 1, ..., f$. ■

3.4.2 Obstacle Avoidance

The framework presented in Section 3.4.1 enables the spacecraft to satisfy the chance constraints of the form (3.36), however obstacle avoidance necessitates handling non-convex

constraints that cannot be expressed in the form $Hx_k \leq h$. In this work, we consider a scenario in which the spacecraft's motion is constrained to be inside a set defined by $Hx_k \leq h$ and outside of the obstacle defined by $Qx_k \leq q$.

This problem is solved by an inner approximation of the non-convex set $\mathbb{C}_{nc} = \{ \xi \mid H\xi \leq h \} \setminus \{ \xi \mid Q\xi \leq q \}$ by a union of N_s convex sets $\mathbb{C}_c = \bigcup_{i=1}^{N_s} \{ \xi \mid H_i\xi \leq h_i \}$ such that $\mathbb{C}_c \subseteq \mathbb{C}_{nc}$. This is illustrated in Figure 3.1, which depicts an example with a cube obstacle inside outer box constraints. For each of these new sets $\mathbb{C}_{c,i} = \{ \xi \mid H_i\xi \leq h_i \}, i = 1, \ldots, N_s$, a chance-constrained admissible set may be defined as in (3.40) and (3.41),

$$\tilde{O}_{\infty,i}(r) = \Bigg\{ \tilde{x}_0 : \begin{bmatrix} H_i & H_i \end{bmatrix} A_{\text{aug}}^t \begin{bmatrix} \tilde{x}_0 \\ 0 \end{bmatrix} \leq h_i(r) - \sqrt{2\Sigma_{t|0}}\, \text{erf}^{-1}(1-2\alpha),\ t \geq 0 \Bigg\}, \tag{3.49}$$

where

$$h_i(r) = h_i - H_i(\mathbb{I} - A_c)^{-1} B_c r. \tag{3.50}$$

Thus a connection between set-points r^i and r^j exists if

$$(\mathbb{I} - A_c)^{-1} B_c (r^i - r^j) \in int\big(\tilde{O}_{\infty,i}(r^j)\big). \tag{3.51}$$

for any $i = 1, \ldots, N_s$.

Corollary 1: Suppose that $\hat{x}_0 - (\mathbb{I} - A_c)^{-1} B_c r_0 \in \tilde{O}_{\infty,i}(r_0)$ for some $i = 1, ..., N_s$, and $w_k, v_k \sim \mathcal{N}(0, \mathbb{I})$, $e_{0|k} \sim \mathcal{N}(0, P_\infty)$ for all $k \geq 0$. And suppose that all of the set-point switches $r_k \leftarrow r^+$ are made when the switching condition $\hat{x}_k - (\mathbb{I} - A_c)^{-1} B_c r^+ \in \tilde{O}_{\infty,i}(r^+)$ is satisfied for some $i = 1, ..., N_s$. Then, the probability of staying in the safety set $\mathbb{C}_{nc}$ is higher than $1 - \alpha$, i.e., Prob$(x_k \in \mathbb{C}_{nc}) \geq 1 - \alpha$, for all $k \geq 0$.

Proof: By a similar proof as that for Proposition 1, it can be shown Prob$(x_k \in \mathbb{C}_c) \geq 1 - \alpha$ for all $k \geq 0$. Then, the statement Prob$(x_k \in \mathbb{C}_{nc}) \geq 1 - \alpha$ for all $k \geq 0$ follows from the fact that $\mathbb{C}_c \subseteq \mathbb{C}_{nc}$. ■

Remark 1: The approach we presented assumes a static obstacle/debris, however, the extension to moving obstacles/debris, including at high relative velocity, is straightforward following [30]. In this case, $Qx_k \leq q$ defines a polyhedral set which overbounds the obstacle path (i.e., union of obstacle positions over time), and has to be avoided by the spacecraft trajectory, while the set $\{\zeta | H\zeta \leq h\}$ represents the other constraints the spacecraft motion has to satisfy.

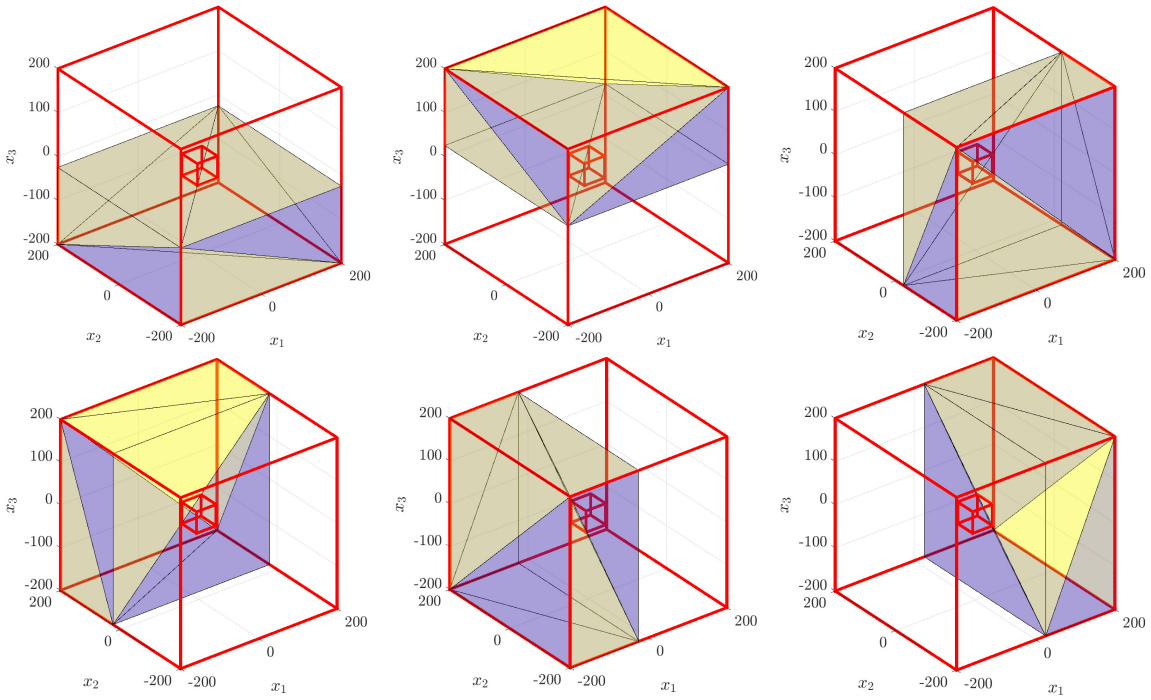

Figure 3.1: Inner approximation of a non-convex set $\mathbb{C}_{nc}$ by a union $\mathbb{C}_c$ of convex sets.

3.5 Simulations

The simulation case studies presented here use the model described in Section 3.2. The discrete-time sampling period and chance constraint probability used for simulation are $\Delta T = 10s$ and $\beta = 1 - \alpha = 0.9$, respectively. The gain matrices K and L are computed by solving the discrete-time algebraic Riccati equation with the corresponding weighting matrices $Q_K = 10^{-7}\mathbb{I}_6$, $R_K = 10\mathbb{I}_3$, $Q_L = 10^{-7}\mathbb{I}_6$, and $R_L = \mathbb{I}_3$.

This section presents the solution for two different scenarios. In the first scenario, the obstacle used is a 9-sided pyramid emanating from the origin, which is intended to represent a sensor-based keep-out zone. In this scenario, it is envisioned that our chaser spacecraft has an optical sensor that is constantly pointing at the target spacecraft at the origin, and the pyramid approximates the cone representing the region in which the sensor would be damaged by the Sun. The second scenario's obstacle is a polyhedral set which overbounds the path of a piece of orbital debris passing through the relative motion frame, as described in Section 3.4.2, Remark 1.

Figure 3.2 depicts an example of one particular chance-constrained admissible set, $\tilde{O}_\infty([97,\ 0,\ 0]^\top)$. Note that this set does not extend all the way to box constraints and that it is non-symmetric about the x_1 and x_2 axes. Figure 3.5 gives an intuitive explanation of these features. The same $\tilde{O}_\infty$ set is shown projected onto the $x_1 - x_2$ plane and the trajec-

tory tubes from two separate simulations are overlaid: one where $x_0 \in \tilde{O}_\infty$ and one where $x_0 \notin \tilde{O}_\infty$. The former shows that the box chance constraints are satisfied while in the latter simulation they are not, illustrating how the asymmetries in the relative orbital dynamics manifest in the asymmetric $\tilde{O}_\infty$ set.

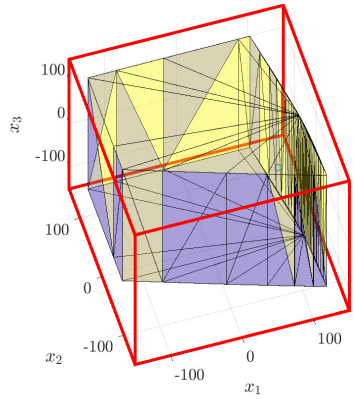

Figure 3.2: Visualization of $\tilde{O}_\infty([97,\ 0,\ 0]^\top)$ for the cuboid constraints shown in red.

After the graph $\mathcal{G}$ is constructed, the standard Dijkstra's algorithm is used for path planning and the solution trajectory is propagated for a 1,000 run Monte Carlo simulation. Figures 3.6, 3.7, and 3.8 show the resulting β-probability trajectory tube of the solution to scenario 1. This tube is defined as the union of β-probability ellipsoids for different k,

$$\{\, \omega \in \mathbb{R} \mid (\omega - x_{1:3,k})^\intercal P_{1:3,k}^{-1} (\omega - x_{1:3,k}) \leq c^2 \,\}, \tag{3.52}$$

where c is solved for using the three degree-of-freedom chi-squared distribution [61]. Figures 3.9 and 3.10 show the solution to scenario 2.

In these simulations, the spacecraft successfully navigates from x_0 to the final reference point r_f while satisfying the chance state constraints and avoiding the obstacle. Additionally, it is shown that the experimentally computed covariance of e matches the theoretical value of P_∞ from (3.25).

3.6 Conclusions

In this chapter, the constrained SRMP framework based on chained positively invariant constraint admissible sets in [30] was extended to the setting of stochastic disturbances and output measurement with stochastic measurement noise. With the proposed approach, chance constraints are considered and maximal chance-constrained admissible sets are exploited

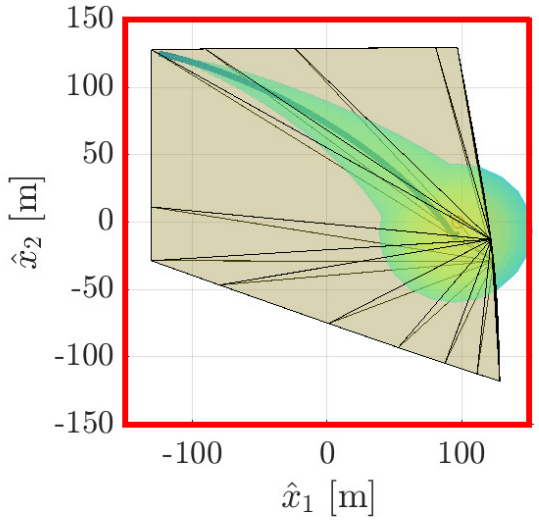

Figure 3.3: Example trajectory where $x_0 \in \tilde{O}_\infty$ and the β chance constraints are satisfied, as illustrated by the β-probability trajectory tube residing fully within the box constraint.

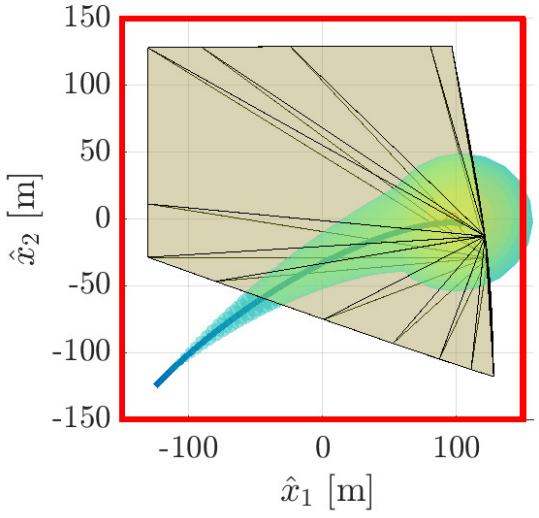

Figure 3.4: Example trajectory where $x_0 \notin \tilde{O}_\infty$ and the β chance constraints are *not* satisfied, as illustrated by the β-probability trajectory tube extending past the box constraint.

Figure 3.5: Comparison of safe and non-safe trajectories, as shown by the $\tilde{O}_\infty$ set.

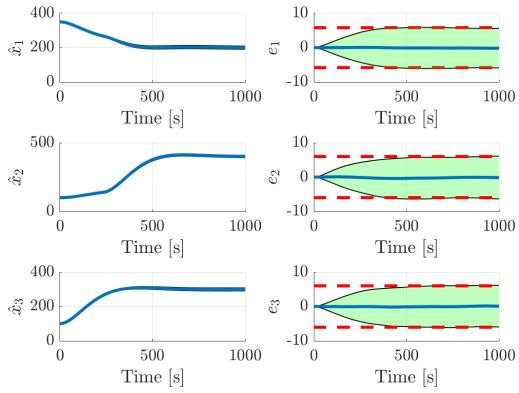

Figure 3.6: Position estimated states $\hat{x}$ and position error states e from Monte Carlo simulations. Green shaded region is the computed β-probability distribution and the dashed red line is the β-probability distribution predicted by (3.25). Initial errors are assumed to be zero.

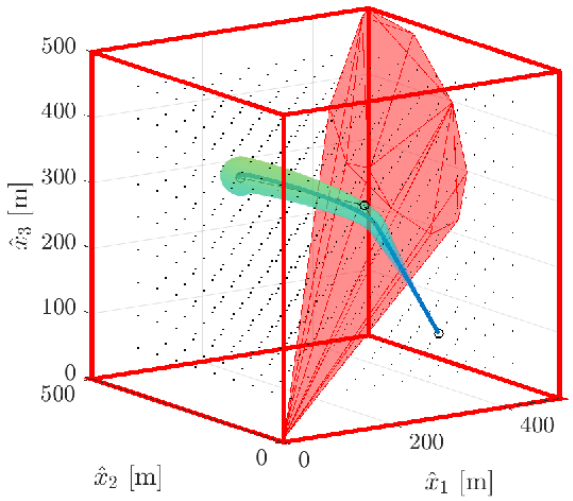

Figure 3.7: Three-dimensional trajectory tube defined by (3.52), tracking $\mathcal{R}$ (circular markers), while avoiding the red conical obstacle. The dot markers are the nodes of the virtual net.

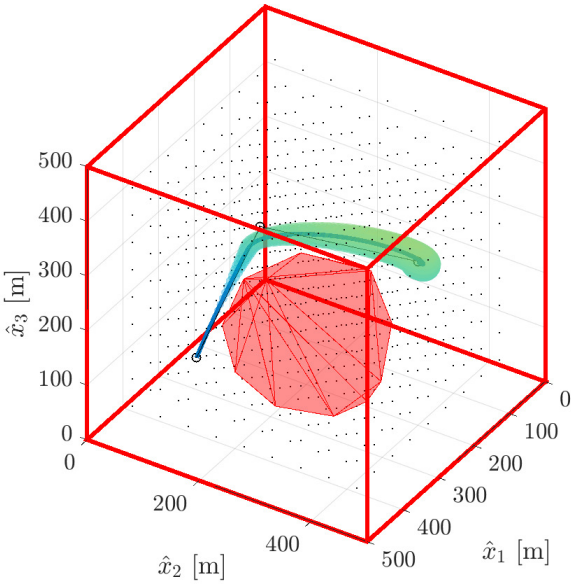

Figure 3.8: Second view of Figure 3.7.

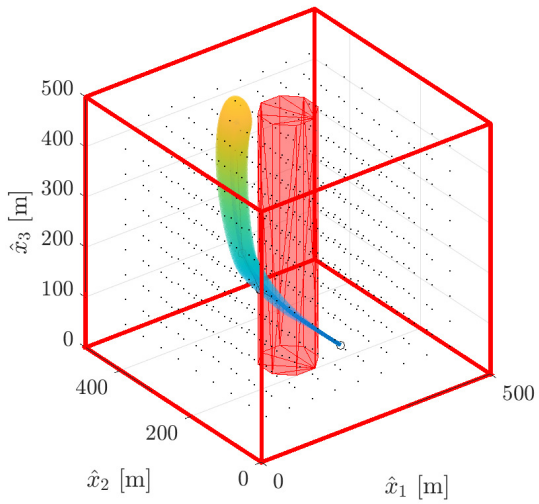

Figure 3.9: Solution to chance-constrained motion planning problem for second scenario, representing orbital debris obstacle avoidance.

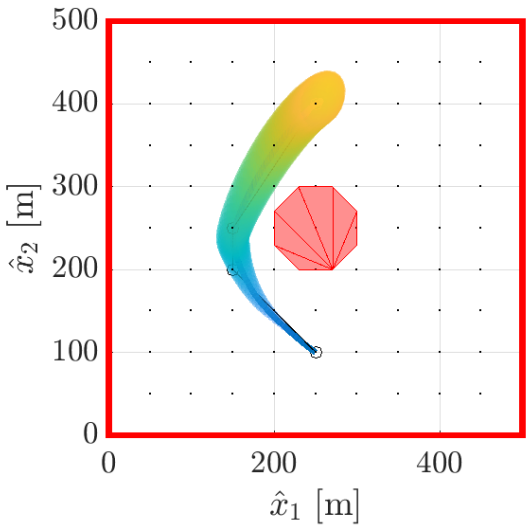

Figure 3.10: Second view of Figure 3.9.

in defining connectivity and possibility of safe transitions between forced equilibria. The relative motion planning problem reduces to the graph search for a path between connected equilibria. As in [30], extensions to the case of multiple control gains appear possible as well as the use of multiple observer gains; details are left to future work. Connectivity is determined via chance-constrained admissible sets, which allows the consideration of output feedback and Gaussian process and measurement noise while still enforcing obstacle avoidance constraints. The resulting graph was searched using Dijkstra's algorithm, resulting in a fuel-efficient trajectory that satisfied the chance constraints with specified probability.

CHAPTER 4

Suborbital Reentry Uncertainty Quantification and Stochastic Optimization

4.1 Introduction

4.1.1 Motivation

Suborbital space launch vehicles are a crucial component of the space industry. They allow access to a region of the atmosphere inaccessible to weather balloons, provide a cheaper platform than orbital launches for scientific experiments and sub-scale demonstrations, and their use as space tourism vehicles is of increasing interest. Between the fiscal years 2008 - 2018, NASA alone launched 178 sounding rockets from over a dozen different launch sites, for a variety of scientific purposes [81]. The private spaceflight company Blue Origin has flown their suborbital New Shepard vehicle 12 times as of January 2020 [82], with the ultimate goal of being a platform for suborbital tourism and scientific payloads.

There are a wide range of suborbital vehicles with mission profiles that are equally as varied, but one important commonality is that all will reenter the atmosphere and land, either ballistically or under chutes. Other vehicles that fall under this category are expended booster stages from orbital vehicles (some of which fall under chutes) and manned space capsules after a ballistic abort.

The landing zones for these vehicles are of great interest from a safety perspective. Whether in an attempt to protect payloads or to protect assets on the ground, it is common for the region of potential landings to include keep-out zones. Examples include spent boosters reentering the atmosphere and colliding with valuable ground support equipment or human-occupied buildings. Conversely, there may be some areas of the landing zone where a valuable payload or human-occupied capsule cannot land safely, such as heavily forested or rocky areas. It is also common for the re-entering vehicle to not have active control over its downrange or crossrange position, such as in the case of a ballistic reentry,

or reentry under symmetric parachutes.

Though the reentry portion of the mission profile may be uncontrolled, it is often the case that a vehicle's ascent trajectory is controlled, and may be biased in the crossrange or downrange directions, affecting the capsule or booster's state at separation and effectively exerting control over the landing location. One solution might be to have a large number of ascent reference trajectories prepared a priori and to pick the best option on launch day based on current wind conditions, but this may be infeasible for software qualification reasons. Each additional ascent trajectory that is maintained in the vehicle's software requires more MC simulations to be conducted to prove it is safe to fly, which is labor and time intensive. It is desirable, therefore, to maintain a small number of ascent reference trajectories to choose from on the Day of Flight (DoF), and that is the motivation for this work.

4.1.2 Problem Statement

The work described in this chapter focuses on two separate but related problems: rapid uncertainty propagation for a vehicle reentering the atmosphere and stochastic optimization of ascent trajectories.

For the former, we seek an algorithm that takes the vehicle's dynamics, physical parameters, initial state, and atmospheric conditions as inputs, and outputs some quantification of the uncertainty associated with the vehicle's state trajectory over the specified flight horizon. For the latter, we seek a method of trajectory optimization that, for a finite number of ascent reference trajectories and the ability to choose the best on DoF, minimizes the probability of a landing zone constraint violation.

It is assumed that the optimized variable in this optimization problem is the vehicle's state at engine cut-off or separation, which is effected via design of the ascent reference trajectory. This choice is justified by another assumption that small differences to the vehicle's downrange or crossrange positions at separation affect downrange or crossrange positions at landing but do not have an appreciable effect on payload delivery, and thus are acceptable optimization parameters from a system level.

4.1.3 Literature Review

A major source of uncertainty considered in this work is stochastic wind gusts. For aircraft, Hoblit [42] provides a detailed description of discrete and continuous gust models and Richardson [43] provides a thorough review of the modeling techniques involved. The Dryden and Von Kármán models are the two most common models [44–48], for which

there are both Federal Aviation Administration and military specifications. Both consider the linear and angular velocity components of the continuous gusts to be varying stochastic processes and then specify those components' PSD. The Dryden model utilizes rational PSDs, while the Von Kármán model uses irrational PSDs. The latter matches experimental gust observations more closely than the former, but its use of irrational spectral densities prevents its spectral factorization from being exactly expressed. The development of wind gust models for suborbital vehicles is an open area of research; other works [83] have adapted a Dryden model to other applications, and we attempt the same here, under an assumption that such a model can be tuned to match higher fidelity wind gust models representative of suborbital vehicles. We also believe that the key steps in treating the problem (which we exemplify below) can still be adopted even if the model of atmospheric disturbances used is very different.

We also consider the problem of propagating the vehicle's state uncertainty subject to the aforementioned continuous wind gust disturbances. Given an initial probability distribution of a state, the objective of a UQ algorithm is to obtain a characterization of the state's probability distribution at a future instant in time. One potential solution is a MC simulation, but the computational intensity of this method limits its usefulness for the stochastic optimization techniques with which we hope to use the UQ algorithm [17, 18]. Linearization techniques suffer from diminished accuracy for highly nonlinear systems or for long time horizons, but their simplicity and high computational efficiency make them well-suited for computationally sensitive applications [19, 20]. Other nonlinear UQ methods include unscented transformation [21, 22], polynomial chaos expansions [23], and Gaussian mixture models [24–26]. A thorough survey of many other UQ methods for problems in orbital mechanics is provided by Luo [49]. The LC method has been chosen for UQ in our problem due to its computational speed and its ability to sufficiently quantify uncertainty along the vehicle's trajectory. In particular, LC UQ methods excel at handling white noise inputs (an important feature of the system considered in this work) when compared to methods such as polynomial chaos or Gaussian mixture models, despite being less capable of capturing strongly nonlinear dynamics. The LC UQ used here is validated against MC UQ in Section 4.4.

The optimization problem considered in this work belongs to a special class called clustering optimization [84] which involves minimizing the cluster function of the following form,

$$F(z_1, \ldots, z_k) = \sum_{a \in \mathbb{A}} \min\big(\phi(z_1, a), \phi(z_2, a), \ldots, \phi(z_k, a)\big)$$

where $\mathbb{A}$ is a finite subset of a finite dimensional space [85]. The optimization of this 'sum of minima' function with respect to $z_1, \ldots, z_k$ corresponds to our problem statement of optimizing a finite number of ascent reference trajectories, from which we have the capability of choosing the lowest cost on DoF. For this application, the cardinality of $\mathbb{A}$ is large, thus F has a large number of shallow local minima. Such problems are nonsmooth, nonconvex, and are best solved using gradient-free optimization methods [86].

4.2 Modeling

4.2.1 Vehicle

As in our previous work [28], the model used is relatively simple, to make the uncertainty quantification more computationally tractable, and has key physical parameters that may be tuned to better match a real-world vehicle. In particular, we consider a model of a suborbital vehicle based on a double integrator with quadratic drag. Modeling fidelity is improved with the inclusion of Coriolis and centripetal accelerations, and by having many parameters depend on altitude and vehicle speed:

$$r = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \; V_0 = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \; V = V_0 - \big(w_n(x_3) + w_g\big), \tag{4.1}$$

$$\Omega = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}^{\times}, \; g = \begin{bmatrix} 0 \\ 0 \\ g_0(x_3) \end{bmatrix}, \; s = \|V\| \tag{4.2}$$

$$\dot{V}_0 = g - \frac{1}{2m}\rho(x_3)S(x_3)C_D(s, x_3)sV - \Omega^2 r - 2\Omega V_0, \tag{4.3}$$

where x_1, x_2, and x_3 are the vehicle's coordinates in an earth-centered, earth-fixed (ECEF) reference frame, V is its velocity with respect to the atmosphere, $w_n \in \mathbb{R}^3$ is the velocity of the nominal local atmosphere with respect to the ground, w_g is the velocity of wind gusts, ω is the rotation rate of the earth, and m is the vehicle mass. Local gravitational acceleration g_0, mean atmospheric wind w_n, atmospheric density ρ, surface reference area S, and coefficient of drag C_D are all indexed by the 'up' vehicle coordinate x_3. Additionally, C_D is indexed by vehicle speed s.

These non-constant parameters, g_0, w_n, ρ, S, and C_D are modeled to vary with altitude to account for the wide-ranging flight regimes a spacecraft experiences during reentry. Ad-

ditionally, this allows us to account for parachute opening dynamics, as one could vary S and C_D to emulate the deployment of chutes at a specified altitude.

4.2.2 Mean Atmospheric Conditions

The ultimate goal of this work is to minimize the overall probability of a landing zone constraint violation, and one way to achieve this is to optimize over a finite but representative set of atmospheric conditions. For illustrative purposes, all atmospheric data used in this work are randomly generated, using a random walk for mean wind velocities w_n and an exponential decay law for atmospheric density, as shown in Figures 4.1 and 4.2. In a flight vehicle implementation of these algorithms, it is anticipated that the atmospheric profiles used for simulation and optimization would come from sources more specific to the launch site, such as outputs from NASA's Earth-GRAM software [87], or measurements from radiosonde and LIDAR equipment. The end result would thus be a set of N_A vectors for $\rho(x_3)$, $w_n(x_3)$, and g_0, indexed by altitude.

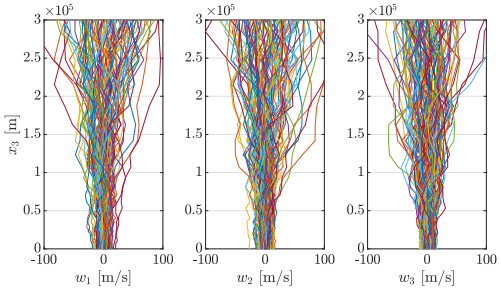

Figure 4.1: Wind velocities w as a function of altitude, for N_A sets of atmospheric conditions.

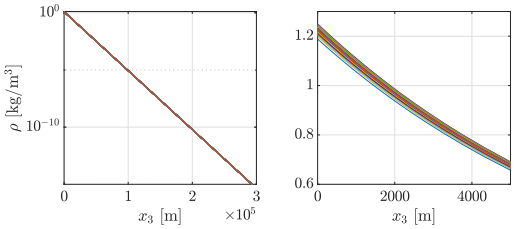

Figure 4.2: Atmospheric density ρ for N_A sets of atmospheric conditions across all relevant altitudes to this problem (left plot) and magnified (right plot).

4.2.3 Wind Gusts

Finally, we consider the model for our wind gust disturbance w_g. A Dryden wind model specifies the PSD for the body-fixed longitudinal, lateral, and vertical directions of a fixed-wing aircraft, but here we have replicated the longitudinal channel in the x_1, x_2, and x_3 directions of the inertial frame. The resulting Dryden-like wind model is summarized as

$$A_i = \frac{-s}{L_i}, \tag{4.4}$$

$$B_i = 1, \tag{4.5}$$

$$C_i = \sqrt{2}s\sigma_i\sqrt{\frac{L_i}{s}}\frac{1}{L_i\sqrt{\pi}}, \tag{4.6}$$

$$\dot{\eta}_i = A_i\eta_i + B_i n_i, \tag{4.7}$$

$$w_i = C_i\eta_i, \tag{4.8}$$

for $i = \{1, 2, 3\}$. Here, A_i, B_i, and C_i are the state space realization of the Dryden model coloring filter, σ_i is the gust intensity parameter, L_i is the characteristic length, and n is the Gaussian white noise input [42]. The output of this model, $w_g = [w_{g1}, w_{g2}, w_{g3}]^\top$, is the wind gust velocity vector in (4.1).

4.2.4 Modeling Summary

We summarize the preceding equations of motion as:

$$\dot{X} = f\big(X, n(t), \theta\big), \tag{4.9}$$

where the state is $X = [r, V_0, \eta_1, \eta_2, \eta_3]^\top$, θ is the set of vehicle and environmental parameters, and n is the Gaussian white noise input.

4.3 Technical Approach

4.3.1 Uncertainty Quantification

The linearized model corresponding to (4.9) has the following form:

$$\delta\dot{X} = A(t)\delta X(t) + B_n(t)n(t), \tag{4.10}$$

$$A(t) = \left.\frac{\partial f}{\partial X}\right|_{\substack{X=\bar{X}(t)\\ n=\bar{n}}}, \tag{4.11}$$

$$B_n(t) = \left.\frac{\partial f}{\partial n}\right|_{\substack{X=\bar{X}(t)\\ n=\bar{n}}}. \tag{4.12}$$

Here, $\bar{X}(t)$ is the nominal trajectory obtained by propagating Eq. (4.9) subject to $n = 0$. Physically, this corresponds to the nominal trajectory that the vehicle would fly under no-gust conditions.

Under the assumption that $n(t)$ is a standard white noise process, the state covariance matrix, P, satisfies the Lyapunov equation [59]:

$$\dot{P}(t) = A(t)P(t) + P(t)A^{\top}(t) + B_n(t)B_n^{\top}(t). \tag{4.13}$$

Note that (4.13) can be solved using any standard ODE solver and the initial condition for P is the zero matrix if the vehicle's initial state is known deterministically.

Now, we can use the nominal trajectory $\bar{X}(t)$, covariance matrix $P(t)$, and a Gaussian distribution density function to estimate the probability that the vehicle is contained in a specified set at time t. Given the mean $\bar{r}(t) = [\bar{x}_1(t), \bar{x}_2(t), \bar{x}_3(t)]^{\top}$ and the block of P corresponding to these states, $\Sigma_{\omega}(t) = P_{(1:3,1:3)}(t)$, we can determine an ellipsoid corresponding to a given probability level, β:

$$Prob\left[\omega \in \mathbb{R}^3 : (\omega - \bar{r}(t))^{\top}\Sigma_{\omega}^{-1}(t)(\omega - \bar{r}(t)) \leq c^2\right] = \beta, \tag{4.14}$$

where c is solved for using the three degree-of-freedom chi-squared distribution [61].

4.3.2 Clustering Optimization

When considering the design objective of minimizing the overall probability of a landing zone constraint violation, a crucial component of the DoF operations is the ability to choose the optimal separation condition from the set of k separation conditions, given current

atmospheric conditions. This knowledge is assumed to be gathered via a weather balloon, atmospheric sensing LIDAR system, or likewise.

Thus if we are optimizing over N_A sets of atmospheric conditions and are able to pick the lowest cost of the k separation conditions, we are attempting to minimize the sum of minima. This gives way to a clustering optimization problem [85] with:

$$F(z_1, \ldots, z_k) = \sum_{a \in \mathbb{A}} \min\big(\phi(z_1, a), \ldots, \phi(z_k, a)\big) \tag{4.15}$$

where each $z_1, \ldots, z_k$ is the set of separation conditions to choose between on DoF, $z = \begin{bmatrix} x_1(0) & x_2(0) \end{bmatrix}^\top$ where time 0 is the time of stage separation. These vectors $z_1, \ldots, z_k$ are the parameters over which we will be optimizing subject to additional constraints such as on maximum deviation ("max divert distance".) The variable a represents a single set of atmospheric conditions belonging to the larger set of N_A atmospheric conditions $\mathbb{A}$. The sub-cost function $\phi(z, a)$ reflects the liklihood of a landing zone constraint violation for a specific set of separation and atmospheric conditions. The preferred option for this sub-cost $\phi(z, a)$ would be to propagate the vehicle's state following the UQ procedure laid out in Section 4.3.1 and compute the actual probability of landing zone constraint violation, which may be possible by integrating vehicle probability density function (PDF) over the set excluded by landing zone constraints. However, this would be prohibitively computationally intensive and so a heuristic sub-cost function is adopted instead. This function $\phi(z, a)$ takes a large value in regions of a landing zone constraint violation, and smoothly transitions to lower values elsewhere, as shown in Figure 4.3. Thus the calculation of this heuristic subcost is based only on the simulated mean of the vehicle's landing position while the vehicle's landing PDF is utilized only for validation purposes in Section 4.4.

The subcost $\phi(z, a)$ is smooth, but the cost function F is nonsmooth. For such functions, non-gradient-based solvers have proven to be more effective in searching for a solution. The optimization algorithm used in this work is MATLAB's implementation of a pattern search algorithm [88], `patternsearch()`. Straightforward extensions of the approach with the cost function (4.15) modified to average over the a priori PDF of atmospheric conditions (if known) are possible but are not pursued in this dissertation.

4.4 Results

The results of a single propagation of Equations (4.9) and (4.13) with initial conditions $X_0 = [0, 0, 10^5, 0, 200, 2000, 0, 0, 0]^\top$ are shown in Figures 4.4 and 4.5, which depict the

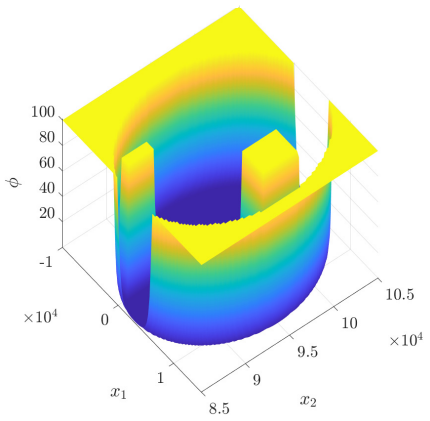

Figure 4.3: Subcost $\phi(z, a)$ as a function of x_1 and x_2 at landing.

vehicle's position coordinates and trajectory, respectively, from stage separation to reentry and landing. Figure 4.4 illustrates that at high altitudes and low air density, the atmospheric forces are negligible and gravitational, Coriolis, and centripetal accelerations dominate. At around $450\ s$ simulation time, the vehicle begins to reenter the atmosphere, its acceleration in the x_3 direction slows, and its accelerations in the x_1 and x_2 directions are dominated by nominal and gust wind velocities, w_n and w_g.

Figure 4.6 validates the linear covariance propagation method laid out in Section 4.3.1 against a 1000-run Monte Carlo simulation of the full nonlinear system (4.9) and numerical computation of the means ($\bar{x}_1$, $\bar{x}_2$, $\bar{x}_3$) and variances ($\sigma^2_{x,1}$, $\sigma^2_{x,2}$, $\sigma^2_{x,3}$,). The only significant difference between the two is in $\sigma^2_{x,3}$ after atmospheric reentry, likely due to the strong influence of the nonlinear drag acceleration during this part of the simulation. Because this work is primarily concerned with the vehicle's covariance in the $x_1 - x_2$ plane at landing, and the landing time prediction is based on the mean $\bar{x}_3$ which is accurately predicted, this deviation between the linear and nonlinear UQ schemes is considered acceptable. Hence LC UQ is utilized for the remainder of this work. In the MATLAB coding environment, running on a single core at 2.7 Ghz, approximate computation time is $0.65\ s$ for LC UQ and $130\ s$ for MC UQ.

Repeating this simulation for every set of atmospheric conditions in $\mathbb{A}$ results in N_A trajectories and N_A landing probability ellipses formed by the intersection of the β-probability tube with the $x_3 = 0$ plane, as shown in Figure 4.7. The probability ellipses are very close to circular, corresponding to a covariance matrix that is very close to diagonal. The corresponding PDF is shown in Figure 4.8. By integrating the portions of the PDF that intersect the landing zone constraints, we determine that the probability of a landing zone constraint violation is $p = 0.205$.

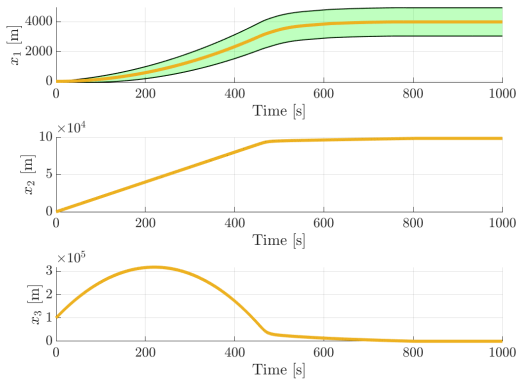

Figure 4.4: Vehicle coordinates as function of time, with β-probability bands.

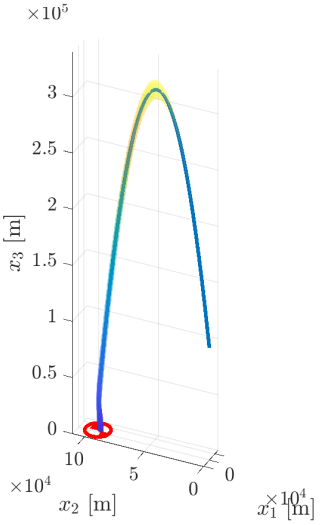

Figure 4.5: 3D β-probability trajectory tube of vehicle's trajectory from stage separation to reentry and landing. Landing zone constraints are shown in red.

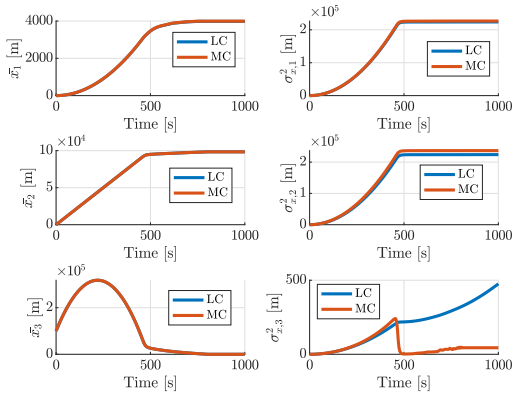

Figure 4.6: Comparison of Monte Carlo UQ (MC, red) and linear covariance UQ (LC, blue).

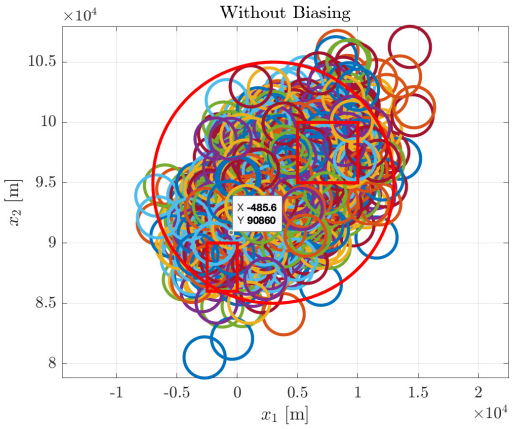

Figure 4.7: Landing ellipses without trajectory biasing, with landing zone constraints shown in red.

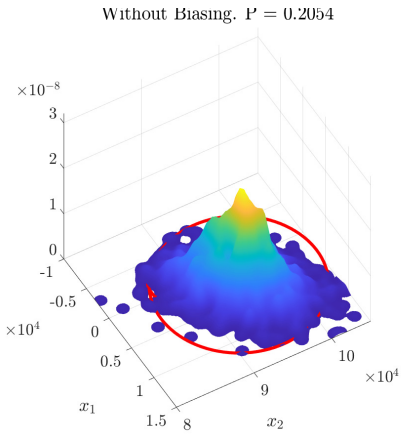

Figure 4.8: Landing PDF without trajectory biasing, with landing zone constraints shown in red. The probability of a landing zone constraint violation is $p = 0.205$.

Minimization of the cost function (4.15) subject to the maximum divert distance constraint of 5000m, i.e., $(\Delta x_1(0))^2 + (\Delta x_2(0))^2 \leq 5000^2$, where $\Delta x_1(0)$ and $\Delta x_2(0)$ are deviations of x_1 and x_2 from the nominal at the time of stage separation, results in the set of k optimal separation conditions shown in Figure 4.9. These separation conditions can be mapped back to a set of optimal ascent reference trajectories. Figure 4.9 reflects three optimal separation conditions resulting from the optimization and a nominal non-diverted trajectory which has also been included in the set for practical reasons. Specifically, in the envisioned application scenario the objective was to augment this non-diverted trajectory already used in existing launch operations. Finally, we re-simulate trajectories for every set of atmospheric conditions in $\mathbb{A}$ while incorporating the trajectory biasing logic and using whichever of the 4 optimal separation conditions results in the lowest cost of $\phi(z, a)$. This produces the landing PDF shown in Figure 4.10 that has been 'shaped' away from landing zone constraints, lowering the overall probability of a constraint violation to $p = 0.004$ and resulting in safer operation of the space vehicle.

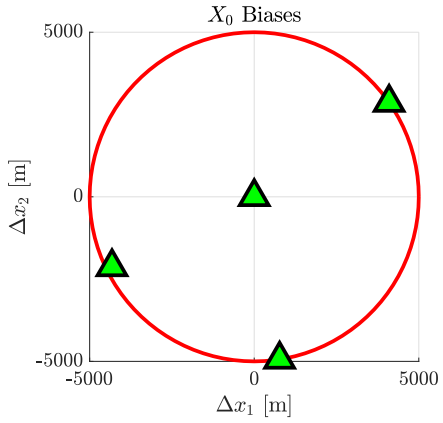

Figure 4.9: Solution to (4.15) with three separation conditions on the control constraint boundary and the nominal separation condition prescribed at $[0\ 0]^\top$.

By repeating this optimization and simulation for different values of the max divert control constraint, we can show how p decreases as the control constraint is relaxed (Figure 4.11).

4.5 Conclusions

In this paper we have considered an approach to optimally designing a set of trajectories, parameterized by the target cross track separation coordinates from the nominal (separation conditions) which can be followed by a suborbital space vehicle. Practical requirements

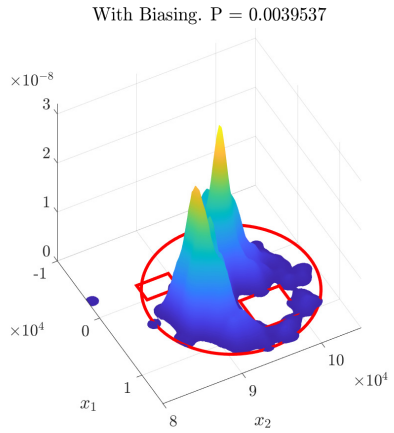

Figure 4.10: Landing PDF *with* trajectory biasing, with landing zone constraints shown in red. The probability of a landing zone constraint violation is $p = 0.004$.

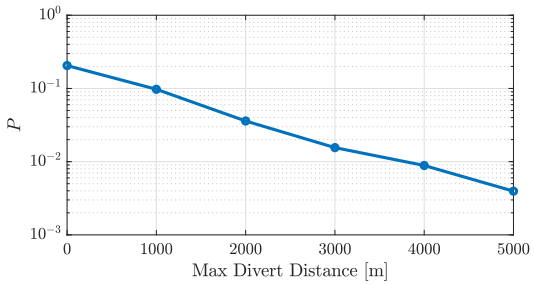

Figure 4.11: Semilog plot of p's sensitivity to the max divert distance.

dictate that a finite set of trajectories be designed and certified a priori and one of these finite number of trajectories be selected on the day of flight depending on the atmospheric conditions. We have shown that such a problem reduces to a clustering optimization problem and illustrated how models, cost function, and constraints can be defined to perform this clustering optimization for our suborbital space vehicle. We have additionally shown that the linear covariance propagation is accurate in predicting the effects of the continuous wind gusts on the cross track deviation of the trajectory at landing while is significantly faster computationally as compared to Monte Carlo simulations. Simulations show that our trajectory selection scheme reduced the probability of a landing zone constraint violation from $p = 0.205$ to $p = 0.004$. This illustrates an effective software-only method for decreasing the probability of a dangerous landing with no physical changes to the vehicle and only minimal changes to its flight controls software.

CHAPTER 5

Suboptimal Nonlinear Model Predictive Control Strategies for Tracking Near Rectilinear Halo Orbits

5.1 Introduction

It has been more than 40 years since a human last traveled beyond low earth orbit (LEO) on the Apollo 17 spacecraft in 1972. Recently, there has been renewed interest in human exploration of the solar system. In particular, cis-lunar space has emerged as a focus area as evidenced by the Global Exploration Roadmap [89, 90] and the NASA Artemis program [91], including the 2024 lunar landing goal [92]. Operations in cis-lunar space will support space-based facilities for robotic and human missions to the Moon and eventually to destinations such as asteroids or Mars.

Near Rectilinear Halo Orbits (NRHOs) have emerged as promising candidates for a long term lunar gateway module and/or as staging orbits between LEO and low lunar orbit [93, 94]. NRHOs are limit cycles near the co-linear L1 and L2 Lagrange points. Certain NHROs possess several useful properties including the existence of low-energy transfer orbits [95], good stability characteristics, unobstructed views of Earth, and favorable resonance properties that allow them to avoid eclipses [96]. They were first identified in the Circular Restricted Three Body Problem (CR3BP) and are periodic natural motion trajectories in the simplified setting of the CR3BP [97].

While certain NRHOs are stable or nearly stable in restricted three body systems [93], in reality maintaining the spacecraft on them is challenging due to perturbations caused by gravitational forces from other celestial bodies, navigational errors, hardware limitations, solar radiation and magnetic forces. This provides the motivation for the development of active station-keeping algorithms that can stabilize the orbit despite these disturbances. Ideally, these algorithms should optimally balance tracking error with propellant/energy con-

sumption. Computing actions/policies offline and uploading them reduces onboard computing requirements but may reduce robustness to disturbances. Alternatively, computing actions online can incorporating feedback leading to improved robustness. In particular, Model Predictive Control (MPC) is a promising methodology for online optimal control that can systematically account for constraints, nonlinearities, and both trajectory tracking and fuel minimization requirements [98–100]. However, online optimization approaches can be prohibitively expensive from a computational or power consumption perspective.

There is a growing body of literature on station-keeping for NRHOs. Dynamical systems theory based techniques using Cauchy-Green Tensors and X-Axis Crossing methods have, in particular, been investigated [96, 101]. Set-invariance and analytical minimum energy-based solutions to a linearized problem that account for measurement uncertainty have been proposed [102], and linear MPC and linear quadratic regulator (LQR) approaches have been developed [103]. There is also interest in station-keeping for other halo orbits about the L1 and L2 points. The Optimal Continuation Strategies method, based on 2-point boundary value problems, was validated in-flight during NASA's ARTEMIS mission [104] which flew longer period halo orbits around the L1 and L2 points. Discrete time sliding mode control has been applied to station-keeping on Halo and Lissajous orbits around L2 [105]. Analytical methods for station-keeping on Halo orbits in the CR3BP using the continuous-time linear quadratic regulator [106] (LQR), Floquet theory [107], and invariant manifolds [108] have also been investigated.

In this paper, we investigate the use of nonlinear model predictive control (NMPC) for station keeping on NRHOs using a low-thrust propulsion system. We show that NMPC is able to successfully maintain spacecraft flight along a halo orbit, satisfy thrust constraints, and demonstrates a high degree of robustness to disturbances and model mismatch. Moreover, while low-thrust actuators are well suited for long duration station-keeping, they require more frequent control updates, leading to more demanding onboard computing requirements. As such, we leverage new optimization algorithms [32] and time-distributed optimization [33] (TDO), a kind of suboptimal MPC, to demonstrate that it is possible to meet closed-loop performance requirements using little computational power. Our approach contrasts with existing literature which often assumes impulsive thrusters and uses control methodologies based on linearized models.

5.2 System Modelling

When considering spaceflight in the cis-lunar flight regime, it is most natural to consider the restricted three body problem in which the third body is of negligible mass compared

to the two primary bodies (as is the case for a spacecraft, the moon, and the earth) [109]. The two restricted three body problems considered in this work are the circular restricted three body problem (for reference trajectory generation and control prediction model), in which the primary and secondary bodies are assumed to travel in circular orbits about their barycenter, and the elliptical restricted three body problem (for simulation and validation), in which the orbits of the primary and secondary bodies are assumed to have nonzero eccentricities.

For both systems of equations, described below, the primary and secondary bodies lie along the x axis with the barycenter at the origin. The z axis points in the direction of angular momentum of the primary-secondary system, and the y axis completes the orthogonal frame.

5.2.1 The Elliptical Restricted Three Body Problem

The equations of motion of the spacecraft in the pulsating frame of Elliptic Restricted Three Body Problem (ER3BP) are given by:

$$x'' - 2y' = \frac{1}{1 + e\cos(\theta)}\frac{\partial U}{\partial x} + u_x \tag{5.1a}$$

$$y'' + 2x' = \frac{1}{1 + e\cos(\theta)}\frac{\partial U}{\partial y} + u_y \tag{5.1b}$$

$$z'' + z = \frac{1}{1 + e\cos(\theta)}\frac{\partial U}{\partial z} + u_z \tag{5.1c}$$

where $(\cdot)'$ denotes differentiation with respect to the true anomaly of the primaries θ, $u = (u_x, u_y, u_z)$ are the control accelerations provided by thrusters and

$$U(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1 - \mu}{\|(x + \mu, y, z)\|_2} + \frac{\mu}{\|(x - 1 + \mu, y, z)\|_2}, \tag{5.2}$$

is the pseudo-potential function, shown in Figure 5.1. Note that the independent variable in (5.1) is the true anomaly which is related to time, t by

$$t' = \frac{1}{(1 + e\cos(\theta))^2}. \tag{5.3}$$

In the absence of eccentricity, i.e., with $e = 0$, the ER3BP reduces to the circular restricted three body problem (CR3BP):

$$x'' - 2y' = \frac{\partial U}{\partial x} + u_x \quad y'' + 2x' = \frac{\partial U}{\partial y} + u_y, \quad z'' + z = \frac{\partial U}{\partial z} + u_z. \tag{5.4a}$$

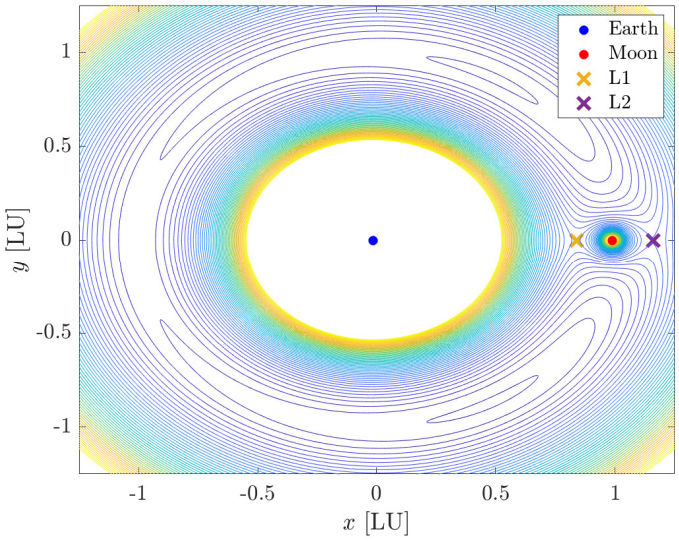

Figure 5.1: Contour plot of the pseudo-potential (5.2) with Lagrangian points L1 and L2 marked.

Note that in the circular case, the rate of change of time with respect to true anomaly is unity and so, barring wrap-around issues ($\theta \in [0, 2\pi]$) , $t = \theta$.

The equations of motion can be simplified by introducing the velocity $v = (x', y', z')$ and position $r = (x, y, z)$ vectors. Then (5.1) becomes

$$\begin{bmatrix} r' \\ v' \end{bmatrix} = \begin{bmatrix} 0 & I \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \left(\frac{\nabla U(r)}{1 + e\cos(\theta)} + u \right) \tag{5.5}$$

where

$$A_{21} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.6}$$

Finally, introducing the state vector $\xi = (r, v)$, (5.5) can be written compactly as

$$\xi' = f_c(\theta, \xi, u). \tag{5.7}$$

5.2.2 Halo Orbits

Halo orbits are families of periodic orbits near the collinear Lagrange points in the Circular Restricted Three Body Problem (CR3BP) [97]. Members of these orbital families that occur closer to the secondary body begin to exhibit properties of NRHOs, traveling very

nearly in a plane normal to the orbital plane of the primaries.

When considering the construction of these orbits in CR3BP we exploit the following property: the velocities x' and z' are equal to zero when the orbit crosses the x-z plane ($y = 0$) at apoapsis and periapsis. Additionally, orbits are symmetric across the x-z plane, so if we assume initial conditions of $X_0 = [x_0, 0, z_0, 0, y_0', 0]$ and integrate until the trajectory again crosses the x-z plane at $X_f = [x_f, 0, z_f, x_f', y_f', z_f']$, we only need to enforce $x_f' = z_f' = 0$ to ensure a periodic orbit. In this work, a single-shooting method is used to solve this problem, iterating on the initial conditions z_0 and y_0' for a given x_0 until the condition $x_f' = z_f' = 0$ is met to within some specified tolerance. Finally, in an attempt to reduce the discontinuity when a controller is tracking the reference trajectory over multiple orbital periods, the state at each time instant is shifted linearly such that the shift at X_0 is zero and the shift at X_f is such that $X_F = X_0$. This results in a reference trajectory that is no longer a natural motion trajectory, but it eliminates the aforementioned discontinuity. A family of periodic orbits about L2 is shown in Figure 5.2.

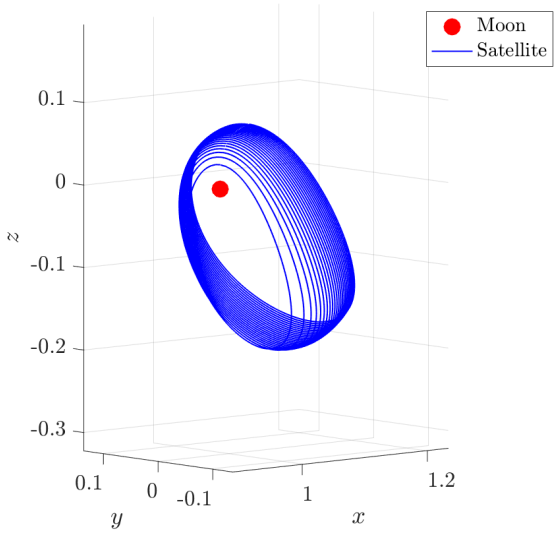

Figure 5.2: Family of Earth-Moon L2 halo orbits in CR3BP.

The single-shooting method utilized in this chapter is limited to finding periodic orbits in system (5.1) only for $e = 0$. The use of a more sophisticated multiple-shooting and homotopy methods to enable finding periodic reference trajectories in ER3BP for $e \neq 0$ is left to future work.

5.3 Control Design

This section describes the proposed controller. An MPC controller has three constituent components, a prediction model used to evaluate the impact of control actions, an Optimal Control Problem (OCP) formulation encapsulating the control problem, and an implementation strategy for solving the OCPs online.

5.3.1 Prediction Model

MPC uses a prediction model to estimate the response of the system to control actions. Such a model is typically of the form

$$\xi_{k+1} = f(\theta_k, \xi_k, u_k), \tag{5.8}$$

where $\xi_k = \xi(\theta_k)$ and $u_k = u(\theta_k)$. We derive a suitable prediction model in this form by discretizing (5.1). The control signal is held constant over each interval $[\theta_k, \theta_{k+1}]$ so the exact model is given by the state transition equation

$$\xi_{k+1} = \phi(\theta_k, \xi_k, u_k) = \xi_k + \int_{\theta_k}^{\theta_{k+1}} f_c(\theta, \xi(\theta), u_k)\, d\theta. \tag{5.9}$$

We approximate (5.9) numerically using a standard 4^{th} order Runge-Kutta (RK4) method:

$$\xi_{k+1} = f(\theta_k, \xi_k, u_k) = \xi_k + \frac{\Delta\theta}{\|b\|_1} \sum_{i=1}^{4} b_i\, k_i(\theta_k, \xi_k, u_k), \tag{5.10}$$

$$k_i = f_c(\theta_k + \Delta\theta\, c_i, \xi_k + \Delta\theta\, a_i\, k_{i-1}, u_k),\ k_0 = 0,$$

with a uniform step size $\Delta\theta > 0$ and where $a = (0, 0.5, 0.5, 1)$, $b = (1, 2, 2, 1)$, and $c = (0, 0.5, 0.5, 1)$ are the coefficients of the method.

5.3.2 Optimal Control Problem Formulation

In MPC the feedback law is defined implicitly through the solution of a receding horizon OCP. Let $N > 0$ be the length of the prediction horizon, i be the index along the prediction horizon, and k be the discrete true anomaly index. We use the notation $\xi_{i|k}$ to denote the predicted state i steps into the prediction horizon at the sampling instant k and denote the

planned control actions $u_{i|k}$ analogously. The OCP formulation is then

$$\min_{\tilde{\xi},\tilde{u}} \quad ||\xi_{i|k} - \bar{\xi}_{i|k}||_Q^2 + \frac{1}{2}\sum_{i=0}^{N-1} ||\xi_{i|k} - \bar{\xi}_{i|k}||_Q^2 + ||u_{i|k}||_R^2 \tag{5.11a}$$

$$s.t. \quad \xi_{0|k} = \xi_k, \tag{5.11b}$$

$$\xi_{i+1|k} = f(\theta_{i|k}, \xi_{i|k}, u_{i|k}), \;\; i = 0, \ldots, N-1, \tag{5.11c}$$

$$\|u_{i|k}\|_\infty \leq u_{max}, \;\; i = 0, \ldots, N, \tag{5.11d}$$

where $\theta_{i|k} = \theta_k + i\Delta\theta$, $\tilde{\xi} = (\xi_{0|k}, \ldots, \xi_{N|k})$ and $\tilde{u} = (u_{0|k}, \ldots, u_{N-1|k})$ are the optimization variables, $\bar{\xi}_{i|k}$ is the desired state, $u_{max} > 0$ is an upper bound on the input acceleration, f is defined in (5.10), and $Q = Q^T \succ 0$ and $R = R^T \succ 0$ are weighting matrices. The control input is then $u_k = u^*_{0|k}$ where $(\cdot)^*$ denotes a minimizer of (5.11).

5.3.3 Controller Implementation

An efficient method for solving (5.11) is essential for implementing NMPC. In this chapter, we use a suboptimal variant of Sequential Quadratic Programming (SQP) algorithm that exploits the time sequential structure of MPC. The OCP (5.11) can be written compactly as the nonlinear programming problem

$$\min_w \quad \phi(w) \tag{5.12a}$$

$$\text{s.t.} \quad g(w, \xi_k) = 0, \quad h(w) \leq 0, \tag{5.12b}$$

where $w = (\tilde{\xi}, \tilde{u})$. The NLP is solved using the sequential quadratic programming iteration

$$w_{i+1|k} = w_{i|k} + d^*, \quad \lambda_{i+1|k} = \lambda^*, \quad v_{i+1|k} = v^*, \tag{5.13}$$

where λ and v are dual variables associated with the equality and inequality constraints, respectively, $z_{i|k} = (w_{i|k}, \lambda_{i|k}, v_{i|k})$ is the solution estimate at the sampling instant θ_k after i SQP iterations and (d^*, λ^*, v^*) is the primal-dual solution to the following quadratic program

$$\min_w \quad d^T \nabla_w^2 \phi(w_{i|k}) d + \nabla_w \phi(w_{i|k})^T d \tag{5.14a}$$

$$\text{s.t.} \quad g(w_{i|k}, \xi_k) + \nabla_w g(w_{i|k}, \xi_k) d = 0, \tag{5.14b}$$

$$h(w_{i|k}) + \nabla_w h(w_{i|k}) d \leq 0. \tag{5.14c}$$

In time-distribution optimization, the SQP algorithm is limited to $\ell > 0$ iterations and the final solution guess from previous timestep is used to warmstart the SQP algorithm, i.e., $z_{0|k+1} = z_{\ell|k}$. This leads to a coupled plant-optimizer system as shown in Figure 5.3. Under appropriate assumptions it is possible to show that TDO based MPC recovers the stability and robustness properties of optimal MPC using a finite number of iterations [33].

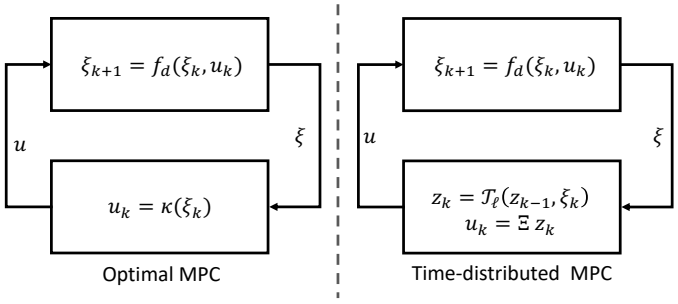

Figure 5.3: A comparison between optimal MPC, which is a static feedback law, and time-distributed MPC, which is a dynamic compensator. The operator $\mathcal{T}_\ell$ represents a fixed number of SQP iterations and Ξ selects the control input from the full solution.

Analytic derivatives, potentially coupled with symbolic optimization [110], are important for accelerating the SQP routine. The cost function and the inequality constraints are quadratic and linear functions so their derivatives are readily available. The derivative of the equality constraints, i.e., of the prediction model, are more complicated due to the RK4 integration scheme used in (5.10). Algorithm 5.1 evaluates the sensitivities $\nabla_\xi f$ and $\nabla_u f$. Finally, the quadratic programming algorithm used to solve (5.14) significantly influences the overall computation time. We use the FBstab method [32] which can exploit the sparsity structure of (5.14) and be easily warmstarted between SQP iterations. The plant and controller are both implemented in Simulink 2019b MATLAB function blocks and compiled into `C` code. We use a MATLAB implementation of FBstab[1] that is compatible with automatic code generation.

5.4 Simulation Results

The specific parameters used for simulations in this work are presented in Table 5.1. Simulations are performed under white Gaussian process noise with zero mean and variance τ. Figures 5.4-5.7 show the results of simulating Equation (5.1) with $e = 0$ subject to the NMPC methodology described in the Control Design Section. Because the reference trajectory being tracked is a periodic orbit in the CR3BP, the spacecraft is able to track the

[1] https://github.com/dliaomcp/fbstab-matlab

Algorithm 5.1 Sensitivities of RK4 Integration

Inputs: $\Delta\theta$, ξ, u,θ
Outputs: $f(\theta,\xi,u)$, $\nabla_\xi f(\theta,\xi,u)$, $\nabla_u f(\theta,\xi,u)$

$k_0 = 0$, $A_0 = 0$, $B_0 = 0$
for $i = 1,\dots 4$ **do**
  $\theta_i = \theta + c_i\Delta\theta$
  $\xi_i = \xi + a_i\Delta\theta k_{i-1}$
  $k_i = f_c(\theta_i,\xi_i,u)$
  $A_i = \nabla_\xi f_c(\theta_i,\xi_i,u)\,[I + a_i\Delta\theta A_{i-1}]$
  $B_i = \nabla_\xi f_c(\theta_i,\xi_i,u)\,[a_i\Delta\theta B_{i-1}] + \nabla_u f_c(\theta_i,\xi_i,u)$
end for
$f(\theta,\xi,u) = \xi + \frac{\Delta\theta}{\|b\|_1}\sum_{i=0}^{4} b_i k_i$
$\nabla_\xi f(\theta,\xi,u) = I + \frac{\Delta\theta}{\|b\|_1}\sum_{i=0}^{4} b_i A_i$
$\nabla_u f(\theta,\xi,u) = \frac{\Delta\theta}{\|b\|_1}\sum_{i=0}^{4} b_i B_i$

reference with comparatively lower state errors and less control input than when the simulation model eccentricity is set to $e = 0.055$, as in Figures 5.8-5.10. Additionally, both simulation scenarios utilize the CR3BP ($e = 0$) in the controller prediction model.

m	$10,000kg$	Spacecraft mass
N	35	OCP horizon length
M	3	Maximum number of SQP iterations
$\Delta\theta$	$0.01rad$	Discrete step size
μ	0.012	Mass ratio
e	0.055	Eccentricity
τ	$10^{-10}\frac{km}{s^2}$	Process noise variance
Q	$10^3\begin{bmatrix}10\mathbb{I}_3 & 0_3\\ 0_3 & \mathbb{I}_3\end{bmatrix}$	Cost function weighting
R	$\mathbb{I}_3$	Cost function weighting
u_{max}	$200mN$ or $2000mN$	Max control constraint

Table 5.1: Simulation parameters

A comparison of Figures 5.6 and 5.9 shows the difference in tracking error between the two simulations, and a comparison of Figures 5.7 and 5.10 illustrates the difference in control utilization and constraint handling. The spacecraft in this simulation required the max control constraint u_{max} to be an order of magnitude higher ($2N$ vs $200mN$) in order to stabilize the system with a nonzero eccentricity. This robustness to large model mismatches and the ability to seamlessly accommodate changes in control constraints is an advantage of the MPC methodology.

In order to compare the spacecraft's performance for different values of M, we intro-

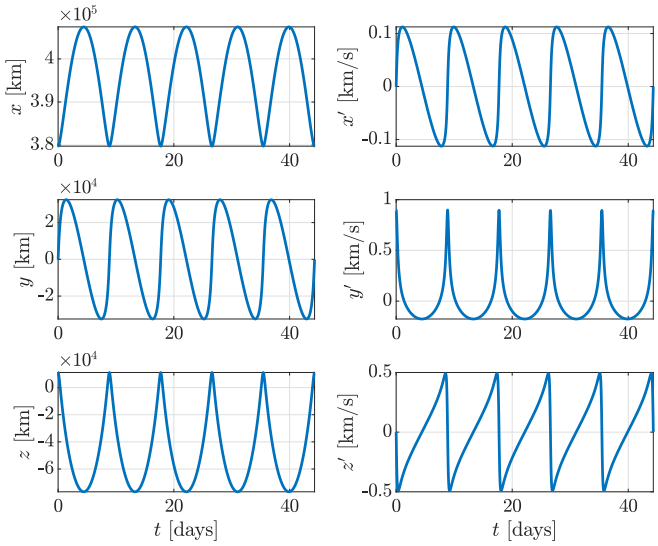

Figure 5.4: State trajectories for spacecraft in CR3BP tracking halo orbit.

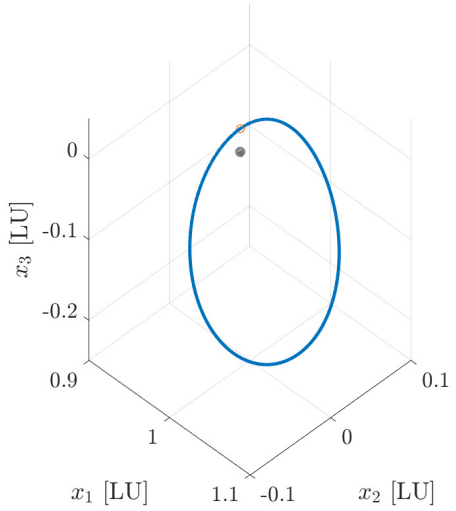

Figure 5.5: Spacecraft trajectory in CR3BP tracking halo orbit, displayed in non-dimensional length units [LU].

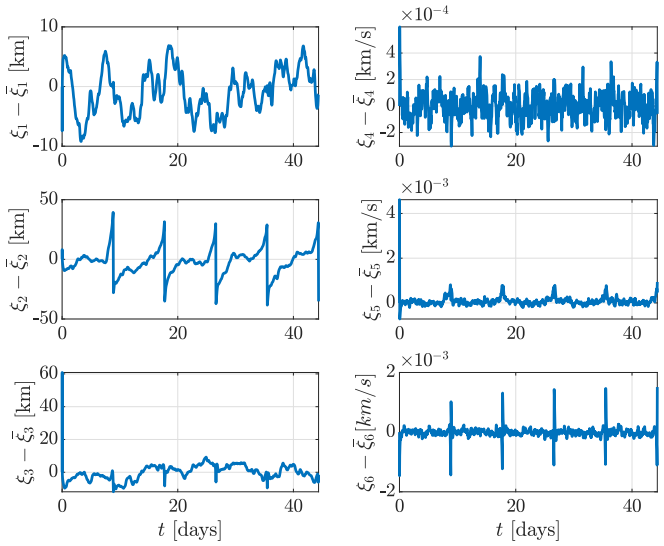

Figure 5.6: State error trajectories for spacecraft in CR3BP tracking halo orbit.

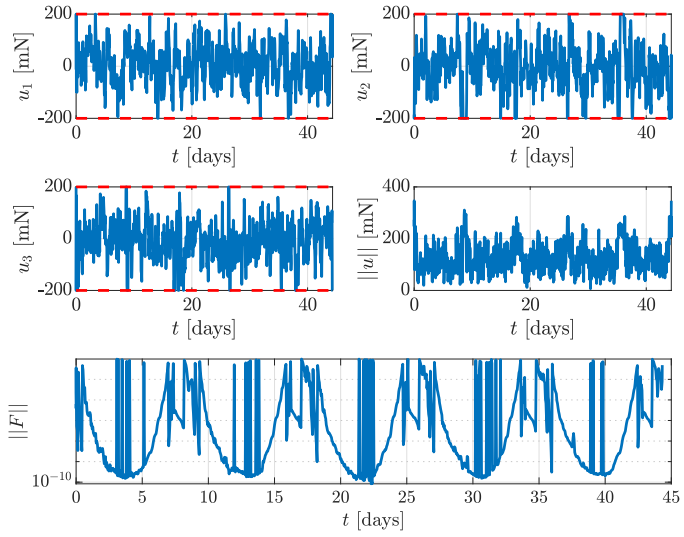

Figure 5.7: Control history and OCP residual $||F||$ for spacecraft in CR3BP tracking halo orbit.

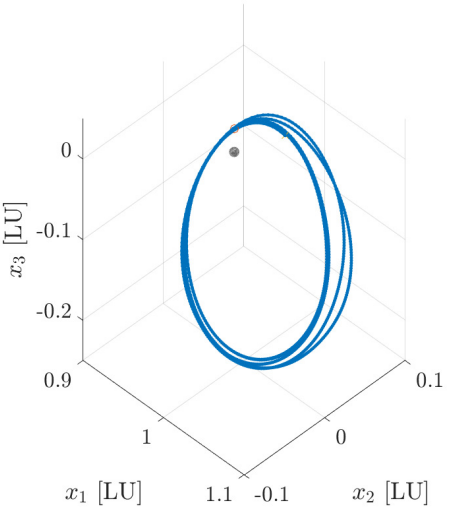

Figure 5.8: Spacecraft trajectory in ER3BP tracking halo orbit, displayed in non-dimensional length units [LU].

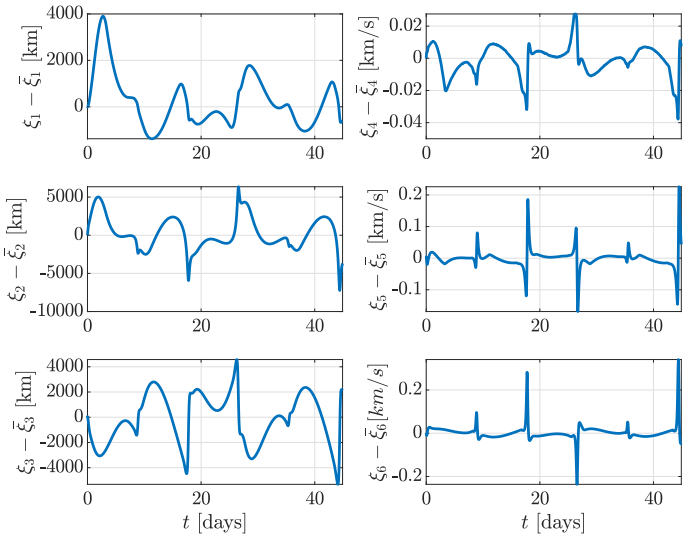

Figure 5.9: State error trajectories for spacecraft in ER3BP tracking halo orbit.

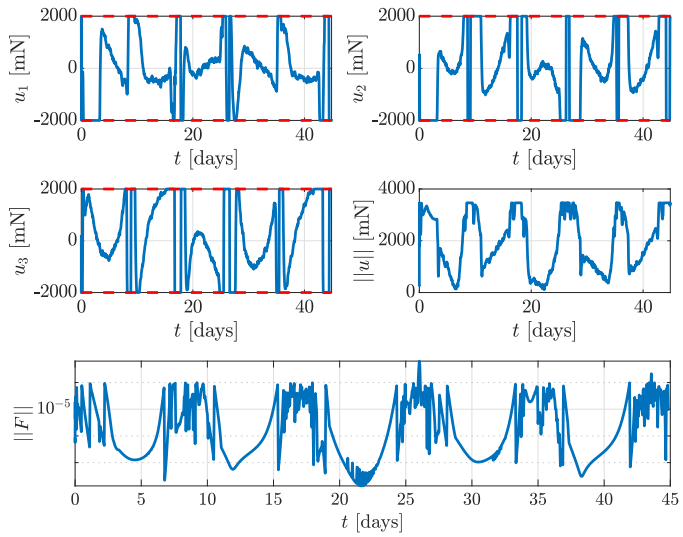

Figure 5.10: Control history and OCP residual $||F||$ for spacecraft in ER3BP tracking halo orbit.

duce a cost function J, analogous to (5.11a),

$$J = \sum_{j} ||\xi_j - \bar{\xi}_j||_Q^2 + ||u_j||_R^2 \tag{5.15}$$

for all discrete θ instants from $\theta = 0$ to the end of the simulation (here, 5 full revolutions of the halo orbit). The results of this sensitivity study are shown in Figure 5.11, showing minimal optimality benefits to running the SQP scheme for more than three iterations. This justifies the use of $M = 3$ and illustrates the computational efficiency benefits of the TDO scheme.

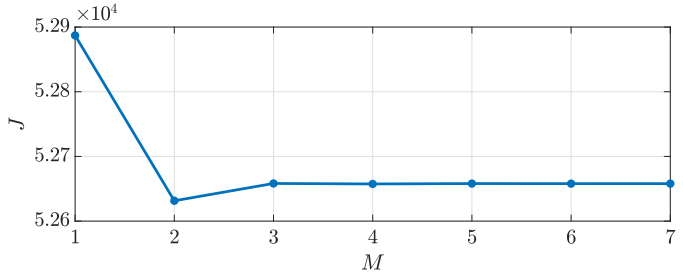

Figure 5.11: Sensitivity study between max SQP iterations M and relative optimality of spacecraft trajectory tracking J.

5.5 Extension to Stochastic Case

This final section introduces the concept of addressing uncertainty directly in the controller formulation instead of relying on deterministic closed loop feedback to reject disturbances. Being able to enforce chance constraints with a prescribed probability allows one to more easily balance performance and safety margins when compared to a deterministic control scheme with static buffers, in some scenarios resulting in dramatically different trajectories. This is introduced in more detail in Chapter 1.

The NRHO station-keeping problem is well-suited for a stochastic NMPC architecture because of the large number of uncertainties associated with spaceflight in the cis-lunar regime and also because of the relatively slow dynamics of the system, allowing plenty of time for computations in between MPC iterations.

5.5.1 OCP Reformulation

The OCP formulation presented in Section 5.3.2 has no state constraints aside from enforcing the system dynamics, so it makes little sense to consider chance constraints. However, there is another formulation for spacecraft station-keeping that involves trajectory tube constraints and makes for a more natural extension to the stochastic case:

$$\min_{\tilde{\xi},\tilde{u},s} \quad \frac{1}{2}\sum_{i=0}^{N} ||u_{i|k}||_R^2 + \gamma s_i \tag{5.16a}$$

$$s.t. \quad \xi_{0|k} = \xi_k, \tag{5.16b}$$

$$\xi_{i+1|k} = f(\theta_{i|k}, \xi_{i|k}, u_{i|k}), \;\; i = 0, \ldots, N-1, \tag{5.16c}$$

$$\|\xi_{i|k} - \bar{\xi}_{i|k}\|_\infty \leq \xi_{max} - d_i + s_i, \;\; i = 0, \ldots, N, \tag{5.16d}$$

$$\|u_{i|k}\|_\infty \leq u_{max}, \;\; i = 0, \ldots, N, \tag{5.16e}$$

$$s_i \geq 0, \;\; i = 0, \ldots, N. \tag{5.16f}$$

Here, s_i are slack variables to ensure the OCP is always feasible, γ is a scalar weighting to penalize use of the slack variables, the scalar ξ_{max} defines a cube state constraint centered at reference state $\bar{\xi}_{i|k}$, and d is a vector of dynamically-scaled buffers to ensure probabilistic satisfaction of the state constraints. Computation of d is covered in Section 5.5.2.

This formulation is motivated by fuel-sensitive missions in which the spacecraft might not need to be tracking the reference trajectory exactly but merely needs to be within some region of it. The OCP formulation does this by removing the quadratic penalty on state

errors, instead opting to only penalize control usage and constrain the spacecraft to be within a tube defined by the union of cube constraints centered on the reference trajectory.

5.5.2 Uncertainty Quantification

A detailed literature review on UQ methods is presented in Chapter 2, but it is envisioned that for real-time stochastic NMPC implementations, a linear covariance analysis is most appropriate.

Consider the discrete-time system given above in Equation 5.8 for an eccentricity of $e = 0$ so there is no explicit dependence on θ,

$$\xi_{k+1} = f(\xi_k, u_k), \tag{5.17}$$

and its θ-varying linearization,

$$\xi_{k+1} = A(\theta_k)\xi_k + B(\theta_k)u_k + \Gamma w_k, \tag{5.18}$$

$$A(\theta_k) = \left.\frac{\partial f}{\partial \xi}\right|_{\substack{\xi=\hat{\xi}(\theta_k) \\ u=\hat{u} \\ w=0}}, \tag{5.19}$$

$$B(\theta_k) = \left.\frac{\partial f}{\partial u}\right|_{\substack{\xi=\hat{\xi}(\theta_k) \\ u=\hat{u} \\ w=0}}, \tag{5.20}$$

where w_k is an n_w-vector disturbance. Here, $\hat{\xi}(\theta_k)$ is the nominal trajectory obtained by propagating Eq. (5.17) subject to $w = 0$ and the optimal control sequence generated by the previous MPC iteration. Physically, this corresponds to the nominal trajectory that the spacecraft would fly under zero-disturbance conditions.

We assume that the disturbance noise sequence is an independent and identically distributed Gaussian process with zero mean and unit covariance matrix, i.e.,

$$w_k \sim \mathcal{N}(0, \mathbb{I}), \tag{5.21}$$

and the process noise matrix Γ is defined as

$$\Gamma = \frac{1}{100}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.22}$$

The state is a then stochastic vector with Gaussian distribution

$$\xi_k \sim \mathcal{N}(\mu_k, P_k), \tag{5.23}$$

and its corresponding mean value μ_k and covariance matrix P_k may be propagated according to

$$\mu_{k+1} = A(\theta_k)\mu_k + B(\theta_k)\hat{u}_k \tag{5.24}$$

$$P_{k+1} = A(\theta_k)P_kA(\theta_k)^\intercal + \Gamma\Gamma^\intercal. \tag{5.25}$$

Thus it is possible to use (5.25) to compute the state covariance matrix at each discrete instant within the OCP prediction horizon.

Similar to the trajectory tube generation described in Chapter 2, we can use the nominal trajectory $\hat{\xi}(\theta_k)$, covariance matrix $P(\theta_k)$, and a Gaussian distribution density function to estimate the probability that the vehicle is contained in a specified set at discrete instant k. Given the mean position $r(\theta_k) = [\mu_1(\theta_k), \mu_2(\theta_k), \mu_3(\theta_k)]^\intercal$ and the block of the covariance matrix corresponding to these states, $\Sigma(\theta_k) = P_{(1:3,1:3)}(\theta_k)$, we can build a set from the definition of a probability ellipsoid [60]:

$$Prob\left[\omega \in \mathbb{R}^3 : (\omega - r_k)^\intercal \Sigma_k^{-1}(\omega - r_k) \leq c^2\right] = \beta, \tag{5.26}$$

where $\beta \in [0, 1]$ is a prescribed probability level and c is solved for using the three degree-of-freedom chi-squared distribution [61]. Thus the vehicle has a β probability of being within the set (5.26) at discrete instant k.

Finally, we assign d_i to be equal to the radius of the over-bounding sphere of the probability ellipsoid at discrete instant i for all $i = 0, \ldots, N$, ensuring that the stochastic trajectory tube defined by (5.26) does not violate the tube constraint defined by ξ_{max}.

5.5.3 Simulation Results

The vehicle and simulation parameters used in the section are detailed in Table 5.1 with $e = 0$ and $\beta = 0.99$. Figures 5.12 and 5.13 show that this stochastic formulation is capable of tracking the reference trajectory just as effectively as in Section 5.4, and Figures 5.14 and 5.15 illustrate the reformulation's advantages. The spacecraft's state error trajectory Figure 5.14 shows the spacecraft drifting with little control input until it approaches the covariance adjusted tube constraints, at which time control action is take to keep the trajectory feasible. Here we can clearly see that the spacecraft is satisfying the tube constraints in a probabilistic sense. Additionally, Figure 5.15 shows that the stochastic NMPC formulation accomplishes this while utilizing less control authority than either example previously shown in Figures 5.7 and 5.10.

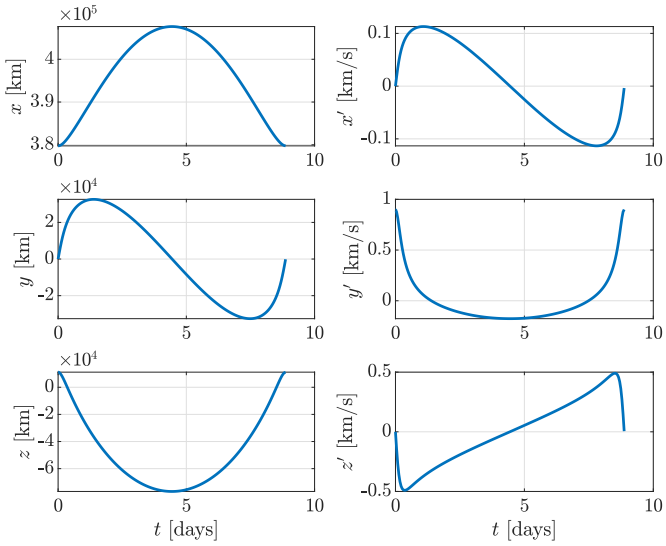

Figure 5.12: State trajectories for spacecraft in CR3BP tracking halo orbit.

5.6 Conclusion

In this chapter we have presented a nonlinear MPC-based approach to the station-keeping of halo orbits in the Earth-Moon system. A method is described for generating periodic halo orbits in the CR3BP and the same equations of motion, in combination with an RK4 integration scheme, are used in the prediction model for the controller. This nonlinear optimization problem is solved with time-distributed SQP techniques utilizing the FBstab quadratic programming algorithm. Finally, the controller is validated in numerical simu-

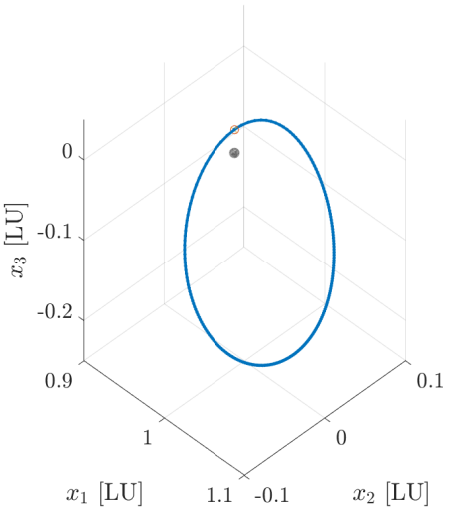

Figure 5.13: Spacecraft trajectory in CR3BP tracking halo orbit, displayed in non-dimensional length units [LU].

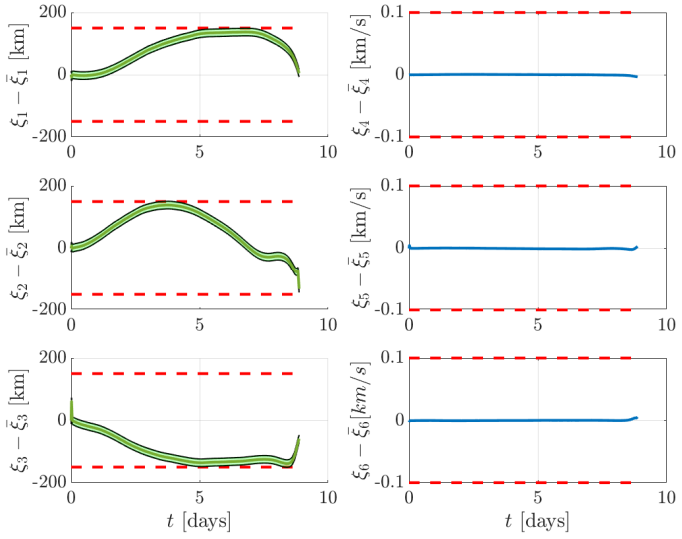

Figure 5.14: State error trajectories for spacecraft in CR3BP tracking halo orbit with stochastic OCP formulation. The β-probability band for state errors are marked in green.

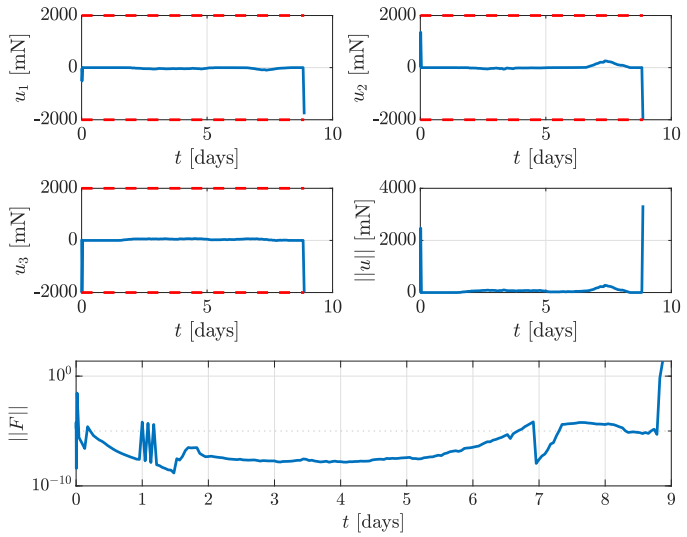

Figure 5.15: Control history and OCP residual $||F||$ for spacecraft in CR3BP tracking halo orbit with stochastic OCP formulation.

lations of the CR3BP and ER3BP with process noise across several full revolutions of the halo orbit.

The results showed that even for large disturbances and model mismatch, the NMPC scheme is capable of stabilizing the system about the reference trajectory while satisfying control constraints. Additionally, it is shown that a TDO approach to the NMPC problem allows for the use of fewer computational resources without significant reduction in performance.

Finally, a stochastic extension to the NRHO station keeping problem is introduced, illustrating state constraint satisfaction in a probabilistic sense.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

This dissertation has presented four different methodologies for the control and optimization of aerospace systems with stochastic disturbances, uncertainties, and constraints. Specifically, we focused on the development and application of stochastic control algorithms for multirotor and fixed-wing sUAS, LEO spacecraft, atmospheric reentry, and cis-lunar spacecraft while considering control constraints, state constraints, obstacle avoidance, fuel efficiency, and computational efficiency.

Firstly, we considered the trajectory validation and path planning needs of future UTM systems. We combined existing Dynamic RRT* and Informed RRT* algorithms, and added an obstacle buffer resizing technique with QP-based collision checking to solve the stochastic path planning problem with chance constraints in a computationally efficient manner. Trajectory validation examples were presented for both quadrotor and fixed-wing models in 3D flight in a non-static atmosphere. Path planning examples were presented for the quadrotor model, showing near-optimal navigation around multiple obstacles while enforcing chance state constraints.

Secondly, we extended the constrained Spacecraft Relative Motion Planning (SRMP) framework based on chained positively invariant constraint admissible sets to the setting of stochastic disturbances and output measurement with stochastic measurement noise. Maximal chance-constrained admissible sets are exploited in defining connectivity and the possibility of safe transitions between forced equilibria subject to chance constraints. The relative motion planning problem reduces to the graph search for a path between connected equilibria, resulting in a fuel-efficient trajectory that satisfies the chance constraints with specified probability.

Thirdly, we introduced a novel approach to suborbital launch operations. We presented a model of a suborbital space launch vehicle, implemented a linear covariance propaga-

tion scheme, and formulated a stochastic clustering optimization problem with the goal of reducing the probability of a landing zone constraint violation by biasing the rocket's ascent trajectory. Simulations showed that this trajectory selection scheme increases the probability of a safe landing by almost two orders of magnitude. This illustrates an effective software-only method for decreasing the probability of a dangerous landing with no physical changes to the vehicle and only minimal changes to its flight controls software.

Finally, we applied a suboptimal NMPC scheme to the problem of station-keeping on halo orbits in the Earth-Moon system. We use the CR3BP equations of motion to generate periodic halo orbits and as the prediction model for the controller, then solve the nonlinear optimization problem with time-distributed SQP techniques utilizing the FBstab quadratic programming algorithm. Numerical simulations show that the NMPC scheme is capable of stabilizing the system about the reference trajectory while satisfying control constraints, even for large process disturbances and model mismatch. Additionally, it was shown that a TDO approach to the NMPC problem allows for the use of fewer computational resources without significant reduction in performance. A stochastic extension to the NRHO station keeping problem with numerical simulation results was presented.

6.2 Future Work

There is ample opportunity to develop convergence guarantees for our Dynamic, Informed, RRT* path planning problem. Each of the individual RRT variants that have been utilized have convergence guarantees, individually. The author expects that some similar results could be produced for the buffer re-sizing algorithm.

The chance-constrained SRMP algorithm implemented in Chapter 2 utilizes chance state constraints for obstacle avoidance, but the methodology could just as easily accomodate control constraints. Thrust saturation constraints are very relevant to real-world SRMP applications and would be a valuable development. Additionally, extensions to the case of multiple control and observer gains, as in [30], seems possible.

The suborbital reentry optimization application presented in Chapter 4 would benefit greatly from more realistic sets of environmental parameters, such as those that might be obtained from NASA Earth-GRAM or similar software. Additionally, the use of a heuristic as the subcost in the clustering optimization problem was motivated by computational simplifications, but optimality could be improved by replacing this heuristic with a direct computation of the probability of a landing zone constraint violation.

Finally, the stochastic extension to the NRHO NMPC problem presented in Chapter 5 is ripe for expansion. Uncertainties related to the mass properties of spacecraft, thruster

mis-alignments, engine-out scenarios, and unmodeled dynamics of cis-lunar flight make this application a good candidate for stochastic control.

BIBLIOGRAPHY

[1] Forecast, F. A., "Fiscal Years 2020-2040," *Federal Aviation Administration*, 2020.

[2] Simelli, I. and Tsagaris, A., "The Use of Unmanned Aerial Systems (UAS) in Agriculture." *HAICTA*, 2015, pp. 730–736.

[3] Xu, Z., Yang, J., Peng, C., Wu, Y., Jiang, X., Li, R., Zheng, Y., Gao, Y., Liu, S., and Tian, B., "Development of an UAS for post-earthquake disaster surveying and its application in Ms7. 0 Lushan Earthquake, Sichuan, China," *Computers & Geosciences*, Vol. 68, 2014, pp. 22–30.

[4] Xu, J., *Design perspectives on delivery drones*, RAND, 2017.

[5] AIA, "U.S. Aerospace and Defense Facts and Figures 2019," Tech. rep., Aerospace Industries Association, 2019.

[6] Kyle, E., "Orbital Launch Summary by Year," Tech. rep., Space Launch Report www.spacelaunchreport.com/logyear.html, 2019.

[7] Box, G. E., "Science and statistics," *Journal of the American Statistical Association*, Vol. 71, No. 356, 1976, pp. 791–799.

[8] Blackmore, L., Ono, M., and Williams, B. C., "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, Vol. 27, No. 6, 2011, pp. 1080–1094.

[9] Ding, X. C., Pinto, A., and Surana, A., "Strategic planning under uncertainties via constrained Markov decision processes," 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 4568–4575.

[10] Bry, A. and Roy, N., "Rapidly-exploring random belief trees for motion planning under uncertainty," 2011 IEEE international conference on robotics and automation, IEEE, 2011, pp. 723–730.

[11] LaValle, S. M., "Rapidly-exploring random trees: A new tool for path planning," Online, 1998, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.1853&rep=rep1&type=pdf, Accessed November 21, 2019.

[12] Adiyatov, O. and Varol, H. A., "A novel RRT*-based algorithm for motion planning in Dynamic environments," Mechatronics and Automation (ICMA), 2017 IEEE International Conference on, IEEE, 2017, pp. 1416–1421.

[13] Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I. V., "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *IEEE Transactions on Control Systems Technology*, Vol. 22, No. 3, 2013, pp. 1018–1031.

[14] Jeong, S. C. and Park, P., "Constrained MPC algorithm for uncertain time-varying systems with state-delay," *IEEE Transactions on Automatic Control*, Vol. 50, No. 2, 2005, pp. 257–263.

[15] Oldewurtel, F., Jones, C. N., and Morari, M., "A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback," *2008 47th IEEE conference on decision and control*, IEEE, 2008, pp. 4731–4736.

[16] Kouvaritakis, B., Cannon, M., and Tsachouridis, V., "Recent developments in stochastic MPC and sustainable development," *Annual Reviews in Control*, Vol. 28, No. 1, 2004, pp. 23–35.

[17] Junkins, J. L., Akella, M. R., and Alfrined, K. T., "Non-Gaussian error propagation in orbital mechanics," *Guidance and Control*, 1996, pp. 283–298.

[18] Sabol, C., Hill, K., Alfriend, K., and Sukut, T., "Nonlinear effects in the correlation of tracks and covariance propagation," *Acta Astronautica*, Vol. 84, 2013, pp. 69–80.

[19] Geller, D. K., "Linear covariance techniques for orbital rendezvous analysis and autonomous onboard mission planning," *Journal of Guidance Control and Dynamics*, Vol. 29, No. 6, 2006, pp. 1404–1414.

[20] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event triggers in linear covariance analysis with applications to orbital rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 102.

[21] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A new approach for filtering nonlinear systems," Vol. 3 of *American Control Conference, Proceedings of the 1995*, IEEE, 1995, pp. 1628–1632.

[22] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on automatic control*, Vol. 45, No. 3, 2000, pp. 477–482.

[23] Wiener, N., "The homogeneous chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.

[24] Terejanu, G., Singla, P., Singh, T., and Scott, P. D., "Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 6, 2008, pp. 1623.

[25] DeMars, K. J., Bishop, R. H., and Jah, M. K., "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, 2013.

[26] Vittaldev, V. and Russell, R. P., "Space object collision probability using multidirectional gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, 2016.

[27] Berning, A. W., Girard, A., Kolmanovsky, I., and D'Souza, S. N., "Rapid uncertainty propagation and chance-constrained path planning for small unmanned aerial vehicles," *Advanced Control for Applications: Engineering and Industrial Systems*, 2019, pp. e23.

[28] Berning, A. W., Taheri, E., Girard, A., and Kolmanovsky, I., "Rapid uncertainty propagation and chance-constrained trajectory optimization for small unmanned aerial vehicles," 2018 Annual American Control Conference (ACC), IEEE, 2018, pp. 3183–3188.

[29] Berning, A. W., Li, N. I., Girard, A., Leve, F. A., Petersen, C. D., and Kolmanovsky, I., "Spacecraft Relative Motion Planning Using Chained Chance-Constrained Admissible Sets," *2020 Annual American Control Conference (ACC)*, IEEE, 2020.

[30] Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Safe positively invariant sets for spacecraft obstacle avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2014, pp. 720–732.

[31] Berning, A. W., Kehlenbeck, A., Girard, A., and Kolmanovsky, I., "Suborbital Reentry Uncertainty Quantification and Stochastic Optimization," *4th Conference on Control Technology and Applications (CCTA)*, IEEE, 2020.

[32] Liao-McPherson, D. and Kolmanovsky, I., "FBstab: A proximally stabilized semismooth algorithm for convex quadratic programming," *Automatica*, Vol. 113, 2020, pp. 108801.

[33] Liao-McPherson, D., Nicotra, M., and Kolmanovsky, I., "Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction," *Automatica*, Vol. 117, 2020, pp. 108973.

[34] Schaufele Jr, R. D., Corning, J., Ding, L., Miller, N., Barlett, A. H., Marotta, T., Lukacs, M., and Dipasis, B., "FAA Aerospace Forecast: Fiscal Years 2018-2038," *Washington, DC*, 2018.

[35] Gill, R. and D'Andrea, R., "An Annular Wing VTOL UAV: Flight Dynamics and Control," *Drones*, Vol. 4, No. 2, 2020, pp. 14.

[36] Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., and Robinson, J., "Unmanned aircraft system traffic management (UTM) concept of operations," Tech. rep., NASA, 2016.

[37] Dill, E. T., Young, S. D., and Hayhurst, K. J., "SAFEGUARD: An assured safety net technology for UAS," 2016 IEEE/AIAA 35th digital avionics systems conference (DASC), IEEE, 2016, pp. 1–10.

[38] D'Souza, S., "Developing a generalized trajectory modeling framework for small UAS performance in the presence of wind," AIAA Information Systems-AIAA Infotech @ Aerospace, AIAA SciTech Forum, 2017.

[39] Richardson, J. R., Atkins, E. M., Kabamba, P. T., and Girard, A. R., "Envelopes for flight through stochastic gusts," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1464–1476.

[40] Belcastro, C. M. and Foster, J. V., "Aircraft loss-of-control accident analysis," Proceedings of AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Paper No. AIAA-2010-8004, 2010.

[41] Administration, F. A., *Review of Aviation Accidents Involving Weather Turbulence in the United States 1992-2001*, Aviation Safety Data Analysis Center, 2004.

[42] Hoblit, F. M., "Gust Loads on Aircraft: Concepts and Applications, AIAA Education Series, American Institute of Aeronautics and Astronautics," *Inc., Washington, DC*, 1988.

[43] Richardson, J. R., *Quantifying and scaling airplane performance in turbulence*, Ph.D. thesis, University of Michigan, Ann Arbor, 2013.

[44] Liepmann, H. W., "On the application of statistical concepts to the buffeting problem," *Journal of the Aeronautical Sciences*, 1952.

[45] De Karman, T. and Howarth, L., "On the statistical theory of isotropic turbulence," Vol. 164.917 of *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, 1938, pp. 192–215.

[46] Von Karman, T., "Progress in the statistical theory of turbulence," *Proceedings of the National Academy of Sciences*, Vol. 34, No. 11, 1948, pp. 530–539.

[47] Von Kármán, T. and Lin, C., "On the statistical theory of isotropic turbulence," *Advances in Applied Mechanics*, Vol. 2, 1951, pp. 1–19.

[48] Diederich, F. and Drischler, J., "Effect of Spanwise Variations in Gust Intensity on the Lift due to Atmospheric Turbulence. National Advisory Committee for Aeronautics," Tech. Rep. Technical Note 3920, NASA, 1957.

[49] Luo, Y.-z. and Yang, Z., "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, Vol. 89, 2017, pp. 23–39.

[50] da Silva Arantes, M., Toledo, C. F. M., Williams, B. C., and Ono, M., "Collision-Free Encoding for Chance-Constrained Nonconvex Path Planning," *IEEE Transactions on Robotics*, Vol. 35, No. 2, 2019, pp. 433–448.

[51] Janson, L., Schmerling, E., and Pavone, M., "Monte Carlo motion planning for robot trajectory optimization under uncertainty," Robotics Research, Springer, 2018, pp. 343–361.

[52] Luders, B., Kothari, M., and How, J., "Chance constrained RRT for probabilistic robustness to environmental uncertainty," AIAA Guidance, Navigation, and Control Conference, AIAA, 2010, p. 8160.

[53] Pairet, È., Hernández, J. D., Lahijanian, M., and Carreras, M., "Uncertainty-based online mapping and motion planning for marine robotics guidance," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 2367–2374.

[54] Axelrod, B., Kaelbling, L. P., and Lozano-Pérez, T., "Provably safe robot navigation with obstacle uncertainty," *The International Journal of Robotics Research*, Vol. 37, No. 13-14, 2018, pp. 1760–1774.

[55] Kothari, M. and Postlethwaite, I., "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, Vol. 71, No. 2, 2013, pp. 231–253.

[56] Borowski, H. and Frew, E., "An evaluation of path planners for guidance with vision based simultaneous localization and mapping," AIAA Guidance, Navigation, and Control Conference, AIAA, 2012, p. 4906.

[57] Karaman, S. and Frazzoli, E., "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, Vol. 30, No. 7, 2011, pp. 846–894.

[58] Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D., "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 2997–3004.

[59] Kabamba, P. T. and Girard, A. R., *Fundamentals of Aerospace Navigation and Guidance*, Cambridge University Press, 2014.

[60] Malyshev, V. V., Krasilshikov, M. N., and Karlov, V. I., *Optimization of observation and control processes*, AIAA, 1992.

[61] Lancaster, H. O. and Seneta, E., *Chi-square distribution*, Wiley Online Library, 1969.

[62] Hanson, F. B., *Applied stochastic processes and control for jump-diffusions: modeling, analysis, and computation*, Vol. 13, Siam, 2007.

[63] Anz-Meador, P., "Orbital Debris Quarterly News," *NASA*, Vol. 23, No. 1, 2019.

[64] Council, N. R. et al., "Committee for the Assessment of NASAs Orbital Debris Programs," *Limiting Future Collision Risk to Spacecraft: An Assessment of NASAs Meteoroid and Orbital Debris Programs*, 2011.

[65] Liu, X. and Lu, P., "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.

[66] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764.

[67] Di Cairano, S., Park, H., and Kolmanovsky, I., "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 22, No. 12, 2012, pp. 1398–1427.

[68] Baldini, F., Bandyopadhyay, S., Foust, R., Chung, S.-J., Rahmani, A., de la Croix, J.-P., Bacula, A., Chilan, C. M., and Hadaegh, F., "Fast motion planning for agile space systems with multiple obstacles," *AIAA/AAS astrodynamics specialist conference*, 2016, p. 5683.

[69] Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., "A survey of spacecraft formation flying guidance and control (part I): Guidance," 2003.

[70] Frey, G. R., Petersen, C. D., Leve, F. A., Kolmanovsky, I. V., and Girard, A. R., "Constrained spacecraft relative motion planning exploiting periodic natural motion trajectories and invariance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3100–3115.

[71] Frey, G. R., Petersen, C. D., Leve, F. A., Girard, A. R., and Kolmanovsky, I. V., "Invariance-based Spacecraft Relative Motion Planning Incorporating Bounded Disturbances and Minimum Thrust Constraints," *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 658–663.

[72] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische mathematik*, Vol. 1, No. 1, 1959, pp. 269–271.

[73] Hart, P. E., Nilsson, N. J., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107.

[74] Frey, G. R., Petersen, C. D., Leve, F. A., Kolmanovsky, I. V., and Girard, A. R., "Incorporating periodic and non-periodic natural motion trajectories into constrained invariance-based spacecraft relative motion planning," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017, pp. 1811–1816.

[75] Berntorp, K., Danielson, C., Weiss, A., and Di Cairano, S., "Positive invariant sets for safe integrated vehicle motion planning and control," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 6957–6962.

[76] Berntorp, K., Weiss, A., Danielson, C., Kolmanovsky, I., and Di Cairano, S., "Automated driving: Safe motion using positively invariant sets," *Int. Conf. Intell. Transp. Syst.*, 2017.

[77] Clohessy, W., "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

[78] Åström, K. J., *Introduction to stochastic control theory*, Courier Corporation, 2012.

[79] Andrews, L. C., *Special functions of mathematics for engineers*, Vol. 49, Spie Press, 1998.

[80] Kalabić, U. V., Li, N. I., Vermillion, C., and Kolmanovsky, I. V., "Reference governors for chance-constrained systems," *Automatica*, Vol. 109, 2019, pp. 108500.

[81] National Aeronautics and Space Administration, "Sounding Rockets Annual Report," 2018.

[82] Wikipedia contributors, "New Shepard — Wikipedia, The Free Encyclopedia," 2020, [accessed January-2020].

[83] Waslander, S. and Wang, C., "Wind disturbance estimation and rejection for quadrotor position control," AIAA Infotech@Aerospace Conference and AIAA Unmanned Unlimited Conference, AIAA, 2009, p. 1983.

[84] Bagirov, A. M., Rubinov, A. M., and Yearwood, J., "A global optimization approach to classification," *Optimization and Engineering*, Vol. 3, No. 2, 2002, pp. 129–155.

[85] Rubinov, A., Soukhoroukova, N., and Ugon, J., "Minimization of the sum of minima of convex functions and its application to clustering," *Continuous Optimization*, Springer, 2005, pp. 409–434.

[86] Bagirov, A., "Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis," *Investigacao Operacional*, Vol. 19, 1999, pp. 75–93.

[87] Justh, H., Justus, C., and Keller, V., "Global reference atmospheric models, including thermospheres, for mars, venus and earth," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006, p. 6394.

[88] Torczon, V., "On the convergence of pattern search algorithms," *SIAM Journal on optimization*, Vol. 7, No. 1, 1997, pp. 1–25.

[89] Hufenbach, B., Laurini, K., Satoh, N., Lange, C., Martinez, R., Hill, J., Landgraf, M., and Bergamasco, A., "International missions to lunar vicinity and surface-near-term mission scenario of the Global Space Exploration Roadmap," *IAF 66th International Astronautical Congress*, 2015.

[90] Laurini, K. C., Hufenbach, B., Hill, J., and Ouellet, A., "The global exploration roadmap and expanding human/robotic exploration mission collaboration opportunities," *IAF 66th International Astronautical Congress*, 2015.

[91] Woodard, M., Folta, D., and Woodfork, D., "ARTEMIS: the first mission to the lunar libration orbits," *21st International Symposium on Space Flight Dynamics, Toulouse, France*, 2009.

[92] Aeronautics, N. and (NASA), S. A., "What is Artemis?" https://www.nasa.gov/what-is-artemis, Accessed: 2020-05-15.

[93] Zimovan, E. M., Howell, K. C., and Davis, D. C., "Near rectilinear halo orbits and their application in cis-lunar space," *3rd IAA Conference on Dynamics and Control of Space Systems, Moscow, Russia*, 2017, p. 20.

[94] Whitley, R. and Martinez, R., "Options for staging orbits in cislunar space," *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–9.

[95] Bury, L. and McMahon, J. W., "Landing Trajectories to Moons from the Unstable Invariant Manifolds of Periodic Libration Point Orbits," *AIAA Scitech 2020 Forum*, 2020, p. 2181.

[96] Guzzetti, D., Zimovan, E. M., Howell, K. C., and Davis, D. C., "Stationkeeping analysis for spacecraft in lunar near rectilinear halo orbits," *27th AAS/AIAA Space Flight Mechanics Meeting*, 2017, pp. 1–20.

[97] Howell, K. and Breakwell, J., "Almost rectilinear halo orbits," *Celestial mechanics*, Vol. 32, No. 1, 1984, pp. 29–52.

[98] Rawlings, J. B. and Mayne, D. Q., *Model predictive control: Theory and design*, Nob Hill Pub., 2009.

[99] Grüne, L. and Pannek, J., "Nonlinear model predictive control," *Nonlinear Model Predictive Control*, Springer, 2017, pp. 45–69.

[100] Ellis, M., Durand, H., and Christofides, P. D., "A tutorial review of economic model predictive control methods," *Journal of Process Control*, Vol. 24, No. 8, 2014, pp. 1156–1178.

[101] Davis, D., Bhatt, S., Howell, K., Jang, J.-W., Whitley, R., Clark, F., Guzzetti, D., Zimovan, E., and Barton, G., "Orbit maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits," 2017.

[102] Muralidharan, V., Weiss, A., and Kalabic, U. V., "Control Strategy for Long-Term Station-Keeping on Near-Rectilinear Halo Orbits," *AIAA Scitech 2020 Forum*, 2020, p. 1459.

[103] Kalabic, U., Weiss, A., Di Cairano, S., and Kolmanovsky, I., "Station-keeping and momentum-management on halo orbits around L2: Linear-quadratic feedback and model predictive control approaches," *Proc. AAS Space Flight Mechanics Meeting*, 2015, pp. 15–307.

[104] Folta, D. C., Pavlak, T. A., Haapala, A. F., Howell, K. C., and Woodard, M. A., "Earth–Moon libration point orbit stationkeeping: theory, modeling, and operations," *Acta Astronautica*, Vol. 94, No. 1, 2014, pp. 421–433.

[105] Lian, Y., Gómez, G., Masdemont, J. J., and Tang, G., "Station-keeping of real Earth–Moon libration point orbits using discrete-time sliding mode control," *Communications in nonlinear science and numerical simulation*, Vol. 19, No. 10, 2014, pp. 3792–3807.

[106] Breakwell, J. V., Kamel, A. A., and Ratner, M. J., "Station-keeping for a translunar communication station," *Celestial Mechanics*, Vol. 10, No. 3, 1974, pp. 357–373.

[107] Simó, C., Gómez, G., Llibre, J., Martinez, R., and Rodriguez, J., "On the optimal station keeping control of halo orbits," *Acta Astronautica*, Vol. 15, No. 6-7, 1987, pp. 391–397.

[108] Gómez, G., Llibre, J., Martínez, R., and Simó, C., "Station keeping of a quasiperiodic halo orbit using invariant manifolds," *Proceed. 2nd Internat. Symp. on spacecraft flight dynamics, Darmstadt*, 1986, pp. 65–70.

[109] Broucke, R., "Stability of periodic orbits in the elliptic, restricted three-body problem." *AIAA journal*, Vol. 7, No. 6, 1969, pp. 1003–1009.

[110] Walker, K., Samadi, B., Huang, M., Gerhard, J., Butts, K., and Kolmanovsky, I., "Design environment for nonlinear model predictive control," Tech. rep., SAE Technical Paper, 2016.

[111] Hull, D. G., *Fundamentals of airplane flight mechanics*, Springer, 2007.

APPENDIX A

Quadrotor Modeling

As a balance between model fidelity and computational efficiency, the quadrotor model used for uncertainty propagation is built upon double integrator dynamics augmented with aerodynamic drag of the following form:

$$r = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \; V_0 = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \; V_q = V_0 - w, \tag{A.1}$$

$$\dot{V}_0 = u - \frac{1}{2m} \rho S C_D V_q \left\| V_q \right\|, \tag{A.2}$$

where x_1, x_2, and x_3 are the vehicle's coordinates in an inertial reference frame, V_q is its velocity with respect to the atmosphere, $w \in \mathbb{R}^3$ is the velocity of the local atmosphere with respect to the ground, $u \in \mathbb{R}^3$ is the control input, m is the vehicle mass, ρ is the air density, S is the reference area, and C_D is the coefficient of drag, assumed to be constant. This model assumes that an inner-loop controller for vehicle attitude and thrust has been implemented that has a feed forward term to cancel out gravity and has sufficiently high bandwidth that we can control acceleration directly. It also assumes a non-static atmosphere and a drag coefficient that remains constant regardless of vehicle state or direction of the relative wind vector V_q. Uncertainty in the drag coefficient can, however, be handled using our UQ and planning algorithms.

When using the above model for uncertainty propagation, it is important that a controller be added so that the uncertainty in vehicle states does not grow so large as to be useless for trajectory prediction purposes. The method presented here is agnostic to controller choice, but for modeling purposes, it is beneficial if the controller can be expressed in a closed form that is easily linearizable. This work utilizes a dynamic extension controller of the following form:

$$e = r - r_{des}, \tag{A.3}$$

$$S_c = \dot{e} + K_q e, \tag{A.4}$$

$$u = \ddot{r}_{des} - K_q \dot{e} - \Lambda_q S_c, \tag{A.5}$$

where $K_q \in \mathbb{R}^{3\times3}$ and $\Lambda_q \in \mathbb{R}^{3\times3}$ are controller gains that may be treated as tuning pa-

rameters to match the flight characteristics of our model to the flight characteristics of a real-world vehicle for which the UTM may be trying to predict its trajectory.

Finally, we consider the model for our wind disturbance. A Dryden wind model specifies the power spectral density (PSD) for the body-fixed longitudinal, lateral, and vertical directions of a fixed-wing aircraft. Other works [83] apply this model directly to quadrotors. Here, the longitudinal channel is replicated in the x_1, x_2, and x_3 directions of the inertial frame. The resulting Dryden-like wind model is summarized as

$$H_i(s) = \sigma_i \sqrt{\frac{2L_i}{\pi ||V_0||}} \frac{1}{1 + \frac{L_i}{||V_0||} s}, \tag{A.6}$$

$$A_i = \frac{-||V_0||}{L_i}, \tag{A.7}$$

$$B_i = 1, \tag{A.8}$$

$$C_i = \sqrt{2}||V_0||\sigma_i \sqrt{\frac{L_i}{||V_0||}} \frac{1}{L_i\sqrt{\pi}}, \tag{A.9}$$

$$\dot{\eta}_i = A_i \eta_i + B_i n, \tag{A.10}$$

$$w_i = C_i \eta_i, \tag{A.11}$$

for $i = \{x_1, x_2, x_3\}$. Here, H_i is the transfer function of the Dryden model coloring filter, A_i, B_i, and C_i are its state space realization, σ_i is the gust intensity parameter, L_i is the characteristic length, and n is the Gaussian white noise input [42]. The output of this model, $w = [w_1, w_2, w_3]^\top$, is the wind velocity vector in (A.2).

APPENDIX B

Fixed-Wing Aircraft Modeling

To illustrate another typical setting which involves sUAS, the EOMs for a fixed-wing aircraft under closed-loop control in a non-static atmosphere were derived. Hull [111] provides EOM's for longitudinal flight with a moving atmosphere and EOM's for full three-dimensional flight in a static atmosphere, but does not provide EOM's for 3D flight in a moving atmosphere.

Figure B.1 shows the unit vectors and rotations necessary to define flight in three dimensions. Here, frame A is the inertial frame, while frames B, C, and D are rotated by angles ψ, γ, and μ, respectively, where ψ is velocity yaw or heading angle, γ is velocity pitch, and μ is velocity roll.

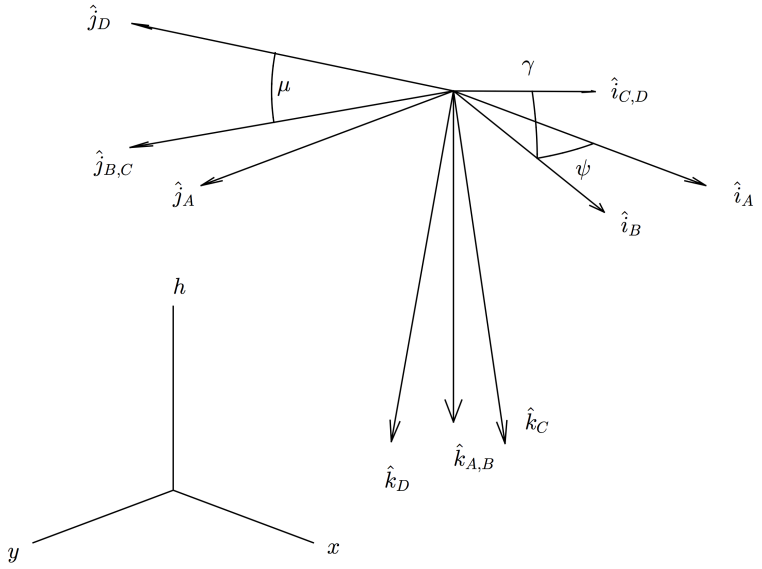

Figure B.1: Three-dimensional flight: unit vectors and rotations.

Additionally, the thrust, drag, lift and gravity forces acting on the aircraft can be defined as follows:

$$\vec{T} = T\hat{i}_D, \tag{B.1}$$
$$\vec{D} = -D\hat{i}_D, \tag{B.2}$$
$$\vec{L} = -L\hat{k}_D, \tag{B.3}$$
$$m\vec{g} = mg\hat{k}_A, \tag{B.4}$$

and the velocity of the aircraft can be represented as $\vec{V} = V\hat{i}_D$, as seen in Figure B.2.

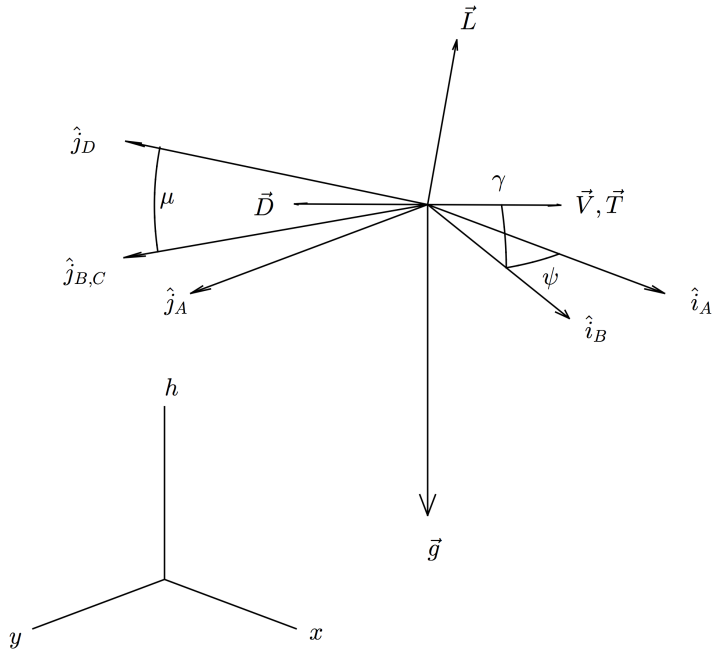

Figure B.2: Three-dimensional flight: forces and velocity.

following equations:

$$\mu = \kappa_\mu(\psi_{des} - \psi), \tag{B.5}$$
$$C_L = \bar{C_L} + \kappa_{C_L}(\gamma_{des} - \gamma), \tag{B.6}$$
$$\dot{T} = \kappa_{T,1}(T_{des} - T), \tag{B.7}$$
$$T_{des} = \bar{T} + \kappa_{T,2}(V_{des} - V). \tag{B.8}$$

Here, ψ_{des}, γ_{des}, T_{des}, and V_{des} are the desired values of the vehicle's yaw, pitch, thrust, and velocity, respectively. The feed forward terms $\bar{C_L}$ and $\bar{T}$ are the values of C_L and T, respectively, that are required to maintain steady flight:

$$\bar{C_L} = \frac{2mg\cos(\gamma)}{SV^2\rho}, \tag{B.9}$$

$$\bar{T} = mg\sin(\gamma) + \frac{1}{2}C_{D0}SV^2\rho + \frac{2K_d m^2 g^2 \cos^2(\gamma)}{SV^2\rho}. \tag{B.10}$$

The vehicle's drag polar parameter is denoted K_d. Similar to the quadrotor model, controller gains κ_μ, κ_{C_L}, $\kappa_{T,1}$, and $\kappa_{T,2}$ may be tuned to better replicate the behavior of a real-world UAS.

The outer-loop controller is separated into two components: longitudinal and lateral. Its design requires us to add two additional states (V_{des} and ψ_{des}) and to estimate $\ddot{\eta}_{des}$ via finite differences, which is undesirable from a computational efficiency perspective, but avoids wrap-around issues.

For the longitudinal controller, V is assumed to be constant and only the $\dot{h}$ dynamics are considered:

$$\dot{h} = V\sin(\gamma), \tag{B.11}$$

$$\text{Control} : \gamma, \tag{B.12}$$

$$e = h - h_{des}, \tag{B.13}$$

$$\dot{e} = -\kappa e, \tag{B.14}$$

$$\gamma_{des} = \sin^{-1}\left(\frac{\dot{h}_{des} - \kappa(h - h_{des})}{V}\right). \tag{B.15}$$

Note that because the above controller is part of the outer-loop, the output is γ_{des} and not γ.

For the lateral controller, we need to control $\dot{\psi}$ so a different approach is pursued. Here, γ is assumed to be constant and only the $\dot{x}$ and $\dot{y}$ kinematics are considered:

$$\dot{x} = V\cos(\gamma)\cos(\psi), \tag{B.16}$$

$$\dot{y} = V\cos(\gamma)\sin(\psi), \tag{B.17}$$

$$\text{Controls} : \dot{V}, \dot{\psi}, \tag{B.18}$$

$$\eta = \begin{bmatrix} x \\ y \end{bmatrix}, \; e = \eta - \eta_{des}, \tag{B.19}$$

$$S = \dot{e} + \kappa e, \tag{B.20}$$

$$\dot{S} = \ddot{\eta} - \ddot{\eta}_{des} + \kappa\dot{e}, \tag{B.21}$$

$$\dot{S} = \begin{bmatrix} \cos(\gamma)\cos(\psi_{des}) & -V_{des}\cos(\gamma)\sin(\psi_{des}) \\ \cos(\gamma)\sin(\psi_{des}) & V_{des}\cos(\gamma)\cos(\psi_{des}) \end{bmatrix} \begin{bmatrix} \dot{V}_{des} \\ \dot{\psi}_{des} \end{bmatrix} - \ddot{\eta}_{des} + \kappa\dot{e}, \tag{B.22}$$

$$\dot{S} = A\nu - \ddot{\eta}_{des} + \kappa\dot{e}, \tag{B.23}$$

$$\dot{S} = -\Lambda_f S, \; -\Lambda_f \text{ Hurwitz}, \tag{B.24}$$

$$\begin{bmatrix} \dot{V} \\ \dot{\psi} \end{bmatrix} = A^{-1}[\ddot{\eta}_{des} - \kappa\dot{e} - \Lambda_f S]. \tag{B.25}$$

This control is invalid if A is singular, corresponding to a pitch angle of $\pm 90^\circ$ or a desired velocity of $V_{des} = 0$, which are flight conditions not encountered in this work.

For the wind disturbance, the Dryden wind model defines wind in the longitudinal, lateral, and vertical body fixed directions, *not* in the x, y, h directions. In the two dimensional case [28], we assumed that γ is small and thus we have wind in the x and h directions. In the three-dimensional case, however, we cannot necessarily make the same assumptions about ψ. Thus, the following wind definitions are used:

$$w = w_u\hat{i}_D + w_w\hat{j}_D + w_v\hat{k}_D = w_x\hat{i}_A + w_y\hat{j}_A + w_h\hat{k}_A. \tag{B.26}$$

We can now write the full equations of motion as follows:

$$\dot{x} = V\cos(\gamma)\cos(\psi) + w_x, \tag{B.27}$$
$$\dot{y} = V\cos(\gamma)\sin(\psi) + w_y, \tag{B.28}$$
$$\dot{h} = V\sin(\gamma) + w_h, \tag{B.29}$$
$$\begin{aligned}\dot{V} =& \frac{T-D}{m} - g\sin(\gamma) - \dot{w}_x\cos(\gamma)\cos(\psi)\\ & - \dot{w}_y\cos(\gamma)\sin(\psi) + \dot{w}_h\sin(\gamma),\end{aligned} \tag{B.30}$$
$$\begin{aligned}\dot{\psi} =& \frac{-1}{Vm\cos(\gamma)}\big[L\sin(\mu) - m\dot{w}_x\sin(\psi)\\ & + m\dot{w}_y\cos(\psi)\big],\end{aligned} \tag{B.31}$$
$$\begin{aligned}\dot{\gamma} =& \frac{1}{Vm}\big[L\cos(\mu) - mg\cos(\gamma) + m\dot{w}_x\cos(\psi)\sin(\gamma)\\ & + m\dot{w}_y\sin(\gamma)\sin(\psi) + m\dot{w}_h\cos(\gamma)\big],\end{aligned} \tag{B.32}$$

where

$$\begin{aligned}w_x =& w_u\cos(\gamma)\cos(\psi) - w_w(\cos(\mu)\sin(\psi)\\ & + \cos(\psi)\sin(\gamma)\sin(\mu)) - w_v(\sin(\mu)\sin(\psi)\\ & - \cos(\mu)\cos(\psi)\sin(\gamma)),\end{aligned} \tag{B.33}$$
$$\begin{aligned}w_y =& w_v(\cos(\psi)\sin(\mu) + \cos(\mu)\sin(\gamma)\sin(\psi))\\ & + w_w(\cos(\mu)\cos(\psi) - \sin(\gamma)\sin(\mu)\sin(\psi))\\ & + w_u\cos(\gamma)\sin(\psi),\end{aligned} \tag{B.34}$$
$$\begin{aligned}w_h =& w_v\cos(\gamma)\cos(\mu) - w_u\sin(\gamma)\\ & - w_w\cos(\gamma)\sin(\mu),\end{aligned} \tag{B.35}$$

$$C_D = C_{D0} + K_d C_L^2, \tag{B.36}$$
$$L = \frac{1}{2}C_L\rho S V^2, \tag{B.37}$$
$$D = \frac{1}{2}C_D\rho S V^2. \tag{B.38}$$

If we assume that $|\dot{w}_u|, |\dot{w}_w|, |\dot{w}_v| \gg |\dot{\gamma}|, |\dot{\psi}|, |\dot{\mu}|$ then we can express the wind accelerations as follows:

$$\begin{aligned}\dot{w}_x &= \dot{w}_u \cos(\gamma)\cos(\psi) - \dot{w}_w(\cos(\mu)\sin(\psi) \\ &\quad + \cos(\psi)\sin(\gamma)\sin(\mu)) - \dot{w}_v(\sin(\mu)\sin(\psi) \\ &\quad - \cos(\mu)\cos(\psi)\sin(\gamma)), \qquad \text{(B.39)} \\ \dot{w}_y &= \dot{w}_v(\cos(\psi)\sin(\mu) + \cos(\mu)\sin(\gamma)\sin(\psi)) \\ &\quad + \dot{w}_w(\cos(\mu)\cos(\psi) - \sin(\gamma)\sin(\mu)\sin(\psi)) \\ &\quad + \dot{w}_u \cos(\gamma)\sin(\psi), \qquad \text{(B.40)} \\ \dot{w}_h &= \dot{w}_v \cos(\gamma)\cos(\mu) - \dot{w}_u \sin(\gamma) \\ &\quad - \dot{w}_w \cos(\gamma)\sin(\mu). \qquad \text{(B.41)}\end{aligned}$$

The power spectral densities (PSD) for the Dryden model are defined following [42]. For the longitudinal channel:

$$\Phi_u(\Omega) = \sigma_u^2 \frac{2L_u}{\pi} \frac{1}{[1 + L_u^2 + \Omega^2]^2}, \qquad \text{(B.42)}$$

For the lateral channel:

$$\Phi_w(\Omega) = \sigma_w^2 \frac{L_w}{\pi} \frac{1 + 3L_w^2\Omega^2}{[1 + L_w^2\Omega^2]^2}, \qquad \text{(B.43)}$$

For the vertical channel:

$$\Phi_v(\Omega) = \sigma_v^2 \frac{L_v}{\pi} \frac{1 + 3L_v^2\Omega^2}{[1 + L_v^2\Omega^2]^2}, \qquad \text{(B.44)}$$

where $\Omega = \frac{\omega}{V}$. The corresponding coloring filters are as follows:

$$G_u(s) = \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1 + \frac{L_u}{V}s}, \qquad \text{(B.45)}$$

$$G_w(s) = \sigma_w \sqrt{\frac{L_w}{\pi V}} \frac{1 + \frac{\sqrt{3}L_w}{V}s}{\left(1 + \frac{L_w}{V}s\right)^2}, \qquad \text{(B.46)}$$

$$G_v(s) = \sigma_v \sqrt{\frac{L_v}{\pi V}} \frac{1 + \frac{\sqrt{3}L_v}{V}s}{\left(1 + \frac{L_v}{V}s\right)^2}. \qquad \text{(B.47)}$$

These have state space realizations of the form:

$$\dot{\eta}_i = A_i\eta_i + B_i n, \qquad \text{(B.48)}$$

$$w_i = C_i\eta_i, \qquad \text{(B.49)}$$

$$\dot{w}_i = C_iA_i\eta_i + C_iB_i n, \qquad \text{(B.50)}$$

for $i \in \{u, w, v\}$, where n is the Gaussian white noise input and the matrices are defined as follows:

$$A_u = \frac{-V}{L_u}, A_w = \begin{bmatrix} \frac{-2V}{L_w} & \frac{-V^2}{L_w^2} \\ 1 & 0 \end{bmatrix}, A_v = \begin{bmatrix} \frac{-2V}{L_v} & \frac{-V^2}{L_v^2} \\ 1 & 0 \end{bmatrix}, \tag{B.51}$$

$$B_u = 1, B_w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, B_v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{B.52}$$

$$C_u = \frac{\sqrt{2}V\sigma_u\sqrt{\frac{L_u}{V}}}{L_u\sqrt{\pi}}, C_w = \frac{V\sigma_w\sqrt{\frac{L_w}{V}}}{L_w\sqrt{\pi}} \begin{bmatrix} \sqrt{3} & \frac{V}{L_w} \end{bmatrix},$$

$$C_v = \frac{V\sigma_v\sqrt{\frac{L_v}{V}}}{L_v\sqrt{\pi}} \begin{bmatrix} \sqrt{3} & \frac{V}{L_v} \end{bmatrix}. \tag{B.53}$$