

# Development of a Reachability Analysis Algorithm for Space Applications

Vom Fachbereich Produktionstechnik

der

UNIVERSITÄT BREMEN

zur Erlangung des Grades

*Doktor-Ingenieur*

genehmigte

Dissertation

von

M. Sc. Yunus Emre Arslantaş

Gutachter: Prof. Dr. rer. nat. Claus Braxmaier  
Fachbereich Raumfahrttechnologie,  
Zentrum für angewandte Raumfahrttechnologie  
und Mikrogravitation (ZARM)

Prof. Dr. Christof Büskens  
Zentrum für Technomathematik,  
Universität Bremen

Tag der mündlichen Prüfung: 20 Juni 2017

---

## Acknowledgements

After leaving four years behind, finally I can set the sails free and depart for the new destinations. I remember my very first day in this beautiful city, and my thoughts haven't even slightly changed after all these years. During this time, I experienced working in an international environment, met very kind people, adapted to a new atmosphere and tried to learn as much as I could. Most importantly, I enjoyed each single moment that I have spent in Bremen.

Foremost, I would like to express my sincere gratitude to Dr. Stephan Theil for providing me this unforgettable experience and his continuous support and trust in my research at DLR. His guidance paved me the way to stay focused on the right track with motivation.

I also would like to thank Prof. Dr. Claus Braxmaier and Prof. Dr. Christof Büskens for their valuable guidance during my study and recommendations for the way forward.

I thank to my colleagues at the GNC department, especially Dr. Marco Sagliano and Dr. Thimo Oehlschlaegel, for their support and contribution to this study with their expertise and patience during restless discussions. After leaving many good memories behind, I am looking forward to the new ones with enthusiasm.

I also take this opportunity to thank my fellow friends Dinçer Demirhan, Ş. Eser Kubalı, Hakan&Zeliha Aydoğan, Kemal Bayhan and O. Tuna Gül for sharing this nice experience with me and their precious friendship. I would like to give a special thank to İlter Haliloğlu for his continuous support from the very beginning. They were beside me whenever I needed them.

During the times, I met amazing people and shared many good memories together which will never be forgotten. I would like to thank all of them for letting me being a part of it.

Finally I pay high regards to my family for the encouragement, support and their understanding for my career decisions. I miss many things during my absence, but realized the value of them in a better way.

This research is supported by the German Aerospace Center - German Academic Exchange Service (DAAD) Fellowship programme. I would like to thank DLR and DAAD for their support in this research.

---

*"Science is the most reliable guide in life."*  
- Mustafa Kemal ATATÜRK

## Abstract

In the last decades developments in space technology paved the way to more challenging missions like asteroid mining, space tourism and human expansion into the Solar System. These missions require difficult tasks such as real-time capable guidance schemes for re-entry, landing on celestial bodies and implementation of large angle maneuvers for spacecraft. There is a need for an analysis tool to increase the robustness and success of these missions. Reachability analysis contributes to this requirement by obtaining the set of all achievable states for a dynamical system starting from an initial condition with given admissible control inputs of the system.

In this study, an optimal control based reachability analysis algorithm is developed for evaluating the performance of the guidance and control methods for space missions considering the desired performance index. The developed method considers a soft-landing problem for a Moon mission as the case study, and attainable area of the lander as the performance index. The method computes feasible trajectories for the lunar lander between the point where the terminal landing maneuver starts and points that constitutes the candidate landing region. The candidate landing region is discretized by equidistant points on a two dimensional plane, i.e. in downrange and crossrange coordinates, and for each grid point a distance function is defined. This distance function acts as an objective function for a related optimal control problem (OCP).

Each infinite dimensional OCP is transcribed into a finite dimensional Non-linear Programming Problem (NLP) by using Pseudo-Spectral Methods (PSM). The NLPs are solved using available tools to obtain feasible trajectories and approximated reachable sets with information about the states of the dynamical system at the grid points. The proposed method approximates reachable sets of the lander with propellant-to-reach and time-to-reach cost by solution of NLPs.

A polynomial-based Apollo guidance scheme is used to compare the results for the developed method. The coefficients that define the position of the lander are obtained by solving a series of explicit equations for the given initial and final states. A model inversion based PD controller is designed to track the generated trajectory. Feasible solutions that satisfy safe landing conditions are filtered and the results are compared for the two different approaches.

Finally, the uncertainties which are characterized by initial state error and system parameters are also considered. A multivariate trajectory interpolation tool is used to interpolate RS with different initial states. A Riccati equation-based controller is designed to track the previously obtained reference trajectories within presence of the uncertainties. Monte Carlo (MC) simulations are carried out to obtain safe attainable landing area of the lunar lander as probability maps. The same uncertainty set is used to verify these probability maps by propagating the uncertainties using unscented transform.

The developed tool analyzes the different guidance and control methods, for the attainable landing area of the lander, under various landing scenarios, with different dynamical models and controller parameters. Numerous quality metrics are used to compare the change of characteristics of the attainable landing area and performance of the guidance and control methods, and selected design parameters.

## Zusammenfassung

In den letzten Jahrzehnten erleichterte die Entwicklungen in der Raumfahrttechnik den Weg für anspruchsvollere Missionen wie Bergbau auf Asteroiden, Weltraumtourismus und die menschliche Expansion in das Sonnensystem. Diese Missionen erfordern schwierige Aufgaben wie echtzeitfähige Flugführungssysteme für den Wiedereintritt, der Landung auf Himmelskörpern und die Durchführung von großen Schwenkmanövern für Raumfahrzeuge. Es besteht derzeit der Bedarf für ein Analysewerkzeug, um die Robustheit und den Erfolg solcher Missionen zu erhöhen. Die Erreichbarkeitsanalyse (engl. Reachability analysis) trägt dazu bei einen Teil dieser Anforderung zu erfüllen. Sie erlaubt es die Menge aller erreichbaren Zustände für ein dynamisches System zu beschreiben. Dabei berücksichtigt sie den Anfangszustand sowie bestimmte zulässige Steuerungseingänge.

In der vorliegenden Arbeit wird ein auf Optimalsteuerung basierender Erreichbarkeits-Algorithmus entwickelt, der Methoden der Steuerung und Regelung für Raumfahrtmission anhand eines gegebenen Leistungsmaßes bewertet. Diese Arbeit betrachtet exemplarisch die Aufgabe, weich auf dem Mond zu landen, wobei die Fläche der erreichbaren Landeregion das Leistungsmaß darstellt. Die Methode berechnet zulässige Trajektorien für einen Mondlander zwischen Startpunkt des finalen Landemanövers und Punkten innerhalb einer potentiell möglichen Landeregion. Diese Landeregion wird durch äquidistante Punkte auf einem Gitter in Flugrichtung und orthogonal dazu diskretisiert (downrange und crossrange), und für jeden Punkt wird eine Abstandsfunktion definiert; diese dient als Zielfunktion des zugehörigen Optimalsteuerungsproblems.

Jedes dieser unendlich dimensional OCP wird in ein endlich dimensionales Nichtlineares Programm (engl. Nonlinear Programming Problem, NLP), unter Verwendung von Pseudospektrale Methoden (PSM), transkribiert. Die NLPs werden mit gängigen Werkzeugen gelöst, um potentielle Trajektorien und eine Approximation der Erreichbarkeitsmenge zu erhalten. Dazu werden die Zustandsinformationen an den Gitterpunkten ausgewertet. Das hier vorgeschlagene Verfahren approximiert die Erreichbarkeitsmenge mit Hilfe eines Zeit- und Treibstoffverbrauch abhängigen Kostenfunctionals und der Lösung der NLPs.

Ein Polynom-basiertes Flugführungsverfahren der Apollo Missionen wird hier verwendet, um die Ergebnisse für die entwickelte Methode zu vergleichen.

Die Koeffizienten, welche die Position des Landers definieren werden durch die Lösung einer Serie von expliziten Gleichungen mit definierten Anfangs- und Endzuständen berechnet. Um die erzeugte Trajektorie zu verfolgen, wird ein auf Modellinversion basierender PD-Regler entworfen. Realisierbare Lösungen, die eine sichere Landung gewährleisten, werden anschließend herausgefiltert und die Ergebnisse werden für die zwei unterschiedlichen Ansätze verglichen.

Abschließend werden die Unsicherheiten, die durch anfängliche Zustandsfehler und Systemparameter gekennzeichnet sind, ebenfalls berücksichtigt. Für die Interpolation zwischen den verschiedenen Erreichbarkeitsmengen wird ein multivariablen Trajektorien-Interpolations-Werkzeug verwendet. Ein auf der Riccati Gleichung basierender Regler wird entworfen, um die vorausberechnete Referenztrajektorie zu verfolgen. Monte Carlo (MC) Simulationen werden durchgeführt, um sichere erreichbare Landeplätze des Mondlanders zu bestimmen und diese als Wahrscheinlichkeitskarten abzubilden. Die gleichen Unsicherheiten werden weiterhin dazu verwendet, um die Wahrscheinlichkeitskarten durch eine Unscented Transformation zu verifizieren.

Das entwickelte Werkzeug analysiert verschiedene Steuerungs- und Regelungskonzepte für die erreichbare Landeregion in diversen Szenarien mit unterschiedlichen Dynamiken und Reglerparametern. Die Charakteristiken der Landeregion und die Leistung der Steuerungs- und Regelungsmethoden und der Designparameter werden anhand verschiedener Metriken verglichen.



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Reachability . . . . .	2
1.2 Safe Landing Area Determination for Soft Landing Problem . . . . .	4
1.3 Scope and Summary of Contributions . . . . .	6
1.4 Outline of Thesis . . . . .	8
<b>I Basic Concepts and Dynamical Models</b>	<b>10</b>
<b>2 State of Art and Problem Statement</b>	<b>11</b>
2.1 Reachable Set . . . . .	11
2.2 Computation of Reachable Sets . . . . .	12
2.3 Reachable Set Computation by Optimal Control . . . . .	14
2.3.1 Moon Landing . . . . .	19
<b>3 Nonlinear Dynamical Model</b>	<b>22</b>
3.1 $d-h$ Model . . . . .	23
3.2 $d-h-c$ Model . . . . .	24
3.3 Simplified $d-h-c$ Model . . . . .	30
3.4 Mission Scenario and Constraints . . . . .	31
<b>II Main Algorithm and Set Computation</b>	<b>33</b>
<b>4 Discretization of Optimal Control Problem</b>	<b>34</b>
4.1 Approximation of Functions using Legendre Polynomials . . . . .	35
4.2 Structure of Jacobian . . . . .	41

## CONTENTS

---

<b>5</b>	<b>Set Computations</b>	<b>47</b>
5.1	Solution of an Optimal Trajectory . . . . .	47
5.2	Reachable Sets for Sample Problems . . . . .	50
5.3	Reachable Sets for Nominal Dynamical Model . . . . .	51
5.4	Reachable Sets for Simplified Dynamical Model . . . . .	55
5.5	Validation of the Results . . . . .	58
<b>6</b>	<b>Polynomial Guidance</b>	<b>61</b>
6.1	Apollo Guidance Algorithm . . . . .	62
6.2	Extended Apollo Guidance Algorithm . . . . .	64
6.3	Lyapunov based PD Controller . . . . .	65
6.4	Polynomial Guidance Based RS . . . . .	71
6.5	Comparison of Reachable Sets . . . . .	75
<b>III</b>	<b>Uncertainties and Parameter Variation</b>	<b>78</b>
<b>7</b>	<b>Monte Carlo Analysis and Quality Assessment</b>	<b>79</b>
7.1	Uncertainties . . . . .	79
7.2	Feedback Controller . . . . .	81
7.3	Monte Carlo Simulations . . . . .	84
<b>8</b>	<b>Reachable Sets using Unscented Transform</b>	<b>91</b>
8.1	Unscented Transform . . . . .	91
8.2	Approximation of Multivariate Probability Density Function . . . . .	93
<b>9</b>	<b>Set Interpolation</b>	<b>100</b>
9.1	Univariate Interpolation . . . . .	100
9.2	Trajectory Interpolation based Set Interpolation . . . . .	102
<b>10</b>	<b>Conclusion</b>	<b>109</b>
10.1	Suggested Future Work . . . . .	110
	<b>References</b>	<b>112</b>
<b>A</b>	<b>Sample Set Computations</b>	<b>121</b>
<b>B</b>	<b>Additional Reachable Sets</b>	<b>123</b>
B.1	Additional Interpolated RS . . . . .	123
B.2	Extended Monte Carlo Analysis . . . . .	127

# List of Figures

1.1	Illustration of a Reachable Set. . . . .	2
1.2	An example forward reachable set of a Lunar lander. The lander can attain any point inside the shaded area. . . . .	3
1.3	An example backward reachable set of a Lunar lander. The lander can achieve a successful landing starting the maneuver from any point inside the shaded region. . . . .	4
1.4	A generic mission profile for Lunar lander. (not scaled) . . . . .	5
1.5	Hazard Detection and Avoidance System for the Landing Phase. . . . .	5
1.6	The Structure of the Thesis . . . . .	9
2.1	Flow Tube Approximation of Van Der Pol System . . . . .	14
2.2	Discretization of a Subset of the State Space . . . . .	17
2.3	Classification of Direct Methods . . . . .	18
2.4	EAGLE Lander Test Platform of DLR . . . . .	21
3.1	$x - y$ and $d - h$ coordinate system. . . . .	23
3.2	Body-Fixed coordinate system. . . . .	25
3.3	$x - y - z$ and $d - h - c$ Coordinate System. . . . .	26
3.4	Representation of Local Horizon and Body-Fixed Coordinate System. . . . .	27
3.5	Vehicle Carried Coordinate System and Body-Fixed Coordinate System. . . . .	27
3.6	Mission Scenario for the Moon Landing. (not scaled) . . . . .	31
4.1	Legendre Polynomials with Different Orders . . . . .	36
4.2	Collocation Points obtained by Different PSM Methods. . . . .	37
4.3	Approximation of a Continuous Function using 11 Uniform Points with Lagrange Polynomials . . . . .	37
4.4	Approximation of a Continuous Function using 41 Uniform Points with Lagrange Polynomials . . . . .	38
4.5	Approximation of a Continuous Function using 11 Nonuniform Points with Lagrange Polynomials. . . . .	39
4.6	Approximation of a Continuous Function using 41 Uniform Points with Lagrange Polynomials. . . . .	40
4.7	Discretization Error Comparison for PSM with Different Number of Nodes. . . . .	40
4.8	Resulting Structure of the Jacobian . . . . .	45

## LIST OF FIGURES

---

5.1	State Trajectories after solving Eq. (2.10). The control inputs are propagated and state trajectories are compared. . . . .	49
5.2	Control Inputs obtained by solving Eq. (2.10). . . . .	49
5.3	Comparison of RS for the Kenderov Problem. . . . .	50
5.4	Comparison of RS for the Rayleigh Problems. . . . .	51
5.5	Reachable Set with Fuel Cost - $t_f = 40$ (s). . . . .	52
5.6	Reachable Set with Fuel Cost - $t_f = \text{free}$ . . . . .	52
5.7	Reachable Set with Time Cost. . . . .	53
5.8	Evolution of the States. . . . .	54
5.9	Reachable Set with $\dot{d}_0 = \dot{c}_0 = 0$ (m/s). . . . .	54
5.10	Reachable Set with $\dot{d}_0 = \dot{c}_0 = 5$ (m/s). . . . .	55
5.11	RS with $-15^\circ/\text{s} < \omega < 15^\circ/\text{s}$ . . . . .	56
5.12	RS with Relaxed Angular Rates and $T_m$ . . . . .	56
5.13	RS with Simplified Dynamical Model. . . . .	57
5.14	The Euclidean Distance for RS with $t_f = 40$ (s). . . . .	58
5.15	The Euclidean Distance for RS with free $t_f$ . . . . .	59
5.16	The Euclidean Distance for RS with free $t_f$ and 90 Nodes for Discretizing the OCP. . . . .	59
6.1	Apollo Mission Descent Guidance Phases and Targets. . . . .	63
6.2	Comparison of High Order and Low Order Polynomial Trajectories. . . . .	65
6.3	Sample Trajectory - Downrange/Crossrange/Altitude. The initial starting point for the maneuver is highlighted. . . . .	66
6.4	Sample Trajectory - Downrange/Crossrange/Altitude vs. Time. . . . .	67
6.5	Block diagram of the closed-loop system. . . . .	71
6.6	States - w/o Feedback Controller for Polynomial Reference Trajectory. . . . .	72
6.8	Controls - w/o Feedback Controller for Polynomial Reference Trajectory. . . . .	72
6.7	Tracking Error - w/o Feedback Controller for Polynomial Reference Trajectory. . . . .	73
6.9	States with Feedback Controller for Polynomial Reference Trajectory. . . . .	74
6.10	Tracking Error with Feedback Controller for Polynomial Reference Trajectory. . . . .	74
6.11	Controls with Feedback Controller for Polynomial Reference Trajectory. . . . .	75
6.12	Comparison of OCP and Polynomial Guidance Based RS . . . . .	76
6.13	Comparison of OCP and Polynomial Guidance Based RS . . . . .	76
6.14	Area Comparison for Polynomial and OCP Based RS . . . . .	77
7.1	Histogram of Uncertainties. . . . .	82
7.2	Block Diagram of the Closed Loop System. . . . .	83
7.3	Simulation for the Tracking Problem. . . . .	84
7.4	State Error Between the Reference Trajectory and the Simulation. . . . .	85
7.5	Representation of the Nonconvex Region by Alpha Shapes. . . . .	86
7.6	Representation of the Maximum Area Ellipse. . . . .	86
7.7	RS with Uncertainty in $\dot{d}_0$ . . . . .	87

**LIST OF FIGURES**

---

7.8	RS with Uncertainty in $\dot{h}_0$ . . . . .	87
7.9	RS with Uncertainty in $\dot{c}_0$ . . . . .	87
7.10	RS with Uncertainty in $d_0$ . . . . .	87
7.11	RS with Uncertainty in $h_0$ . . . . .	87
7.12	RS with Uncertainty in $c_0$ . . . . .	87
7.13	RS with Uncertainty in $\beta_0$ . . . . .	88
7.14	RS with Uncertainty in $\chi_0$ . . . . .	88
7.15	RS with Uncertainty in $m_0$ . . . . .	88
7.16	RS with Uncertainty in SFC. . . . .	88
7.17	RS with Uncertainty in All States. . . . .	89
8.1	The Concept of Unscented Transform Using $\sigma$ Points. . . . .	92
8.2	Variation of $\kappa$ for Determining $\sigma$ Points. . . . .	94
8.3	Variation of $\alpha$ for Determining $\sigma$ -Points. . . . .	94
8.4	Variation of Probability Density Function with $\sigma$ -Points. . . . .	95
8.5	Probability Distribution of Final Velocities in Downrange and Crossrange. . . . .	95
8.6	Safe Landing Constraints of Final Velocities in Downrange and Crossrange. . . . .	96
8.7	Probability Map with Unscented Transform ( $1\sigma$ ) . . . . .	97
8.8	Probability Map with Unscented Transform ( $2\sigma$ ) . . . . .	98
8.9	Probability Map with Unscented Transform ( $3\sigma$ ) . . . . .	98
8.10	Comparison of MC and $\sigma$ Point Analysis . . . . .	99
8.11	Comparison of MC and $\sigma$ Point Analysis . . . . .	99
9.1	Trajectory Interpolation with Different Initial States. . . . .	102
9.2	Interpolation of RS with different Initial States and Comparison w.r.t. a Reference RS. . . . .	103
9.3	Concept of Hausdorff Distance . . . . .	104
9.4	Set Interpolation with $\dot{d}_0 = 3\text{m/s}$ and $\dot{d}_0 = 7\text{m/s}$ . . . . .	105
9.5	Set Interpolation with $\dot{h}_0 = -21\text{m/s}$ and $\dot{h}_0 = -17\text{m/s}$ . . . . .	106
9.6	Set Interpolation with $\dot{c}_0 = -2\text{m/s}$ and $\dot{c}_0 = 2\text{m/s}$ . . . . .	106
9.7	Set Interpolation with $d_0 = -15\text{m}$ and $d_0 = 15\text{m}$ . . . . .	106
9.8	Set Interpolation with $h_0 = 285\text{m}$ and $h_0 = 315\text{m}$ . . . . .	107
9.9	Set Interpolation with $c_0 = -15\text{m}$ and $c_0 = 15\text{m}$ . . . . .	107
9.10	Set Interpolation with $\beta_0 = -88.06^\circ$ and $\beta_0 = -84.06^\circ$ . . . . .	107
9.11	Set Interpolation with $\chi_0 = -2^\circ$ and $\chi_0 = 2^\circ$ . . . . .	108
9.12	Set Interpolation with %98 $m_0$ and %102 $m_0$ . . . . .	108
B.1	Set Interpolation with $\dot{d}_0 = 1\text{m/s}$ and $\dot{d}_0 = 9\text{m/s}$ . . . . .	123
B.2	Set Interpolation with $\dot{h}_0 = -23\text{m/s}$ and $\dot{h}_0 = -15\text{m/s}$ . . . . .	124
B.3	Set Interpolation with $\dot{c}_0 = -4\text{m/s}$ and $\dot{c}_0 = 4\text{m/s}$ . . . . .	124
B.4	Set Interpolation with $d_0 = -30\text{m}$ and $d_0 = 30\text{m}$ . . . . .	124
B.5	Set Interpolation with $h_0 = 270\text{m}$ and $h_0 = 330\text{m}$ . . . . .	125
B.6	Set Interpolation with $c_0 = -30\text{m}$ and $c_0 = 30\text{m}$ . . . . .	125
B.7	Set Interpolation with $\beta_0 = -90.06^\circ$ and $\beta_0 = -82.06^\circ$ . . . . .	125

## LIST OF FIGURES

---

B.8	Set Interpolation with $\chi_0 = -4^\circ$ and $\chi_0 = 4^\circ$ .	126
B.9	Set Interpolation with $96m_0$ and $104m_0$ .	126
B.10	MC Simulations with uncertainty in $\dot{d}_0$ .	127
B.11	Histogram of Final States with Uncertainty in $\dot{d}_0$ .	127
B.12	MC Simulations with uncertainty in $\dot{h}_0$ .	128
B.13	Histogram of Final States with Uncertainty in $\dot{h}_0$ .	128
B.14	MC Simulations with uncertainty in $\dot{c}_0$ .	129
B.15	Histogram of Final States with Uncertainty in $\dot{c}_0$ .	129
B.16	MC Simulations with uncertainty in $\dot{d}_0$ .	130
B.17	Histogram of Final States with Uncertainty in $d_0$ .	130
B.18	MC Simulations with uncertainty in $h_0$ .	131
B.19	Histogram of Final States with Uncertainty in $h_0$ .	131
B.20	MC Simulations with uncertainty in $c_0$ .	132
B.21	Histogram of Final States with Uncertainty in $c_0$ .	132
B.22	MC Simulations with uncertainty in $\beta_0$ .	133
B.23	Histogram of Final States with Uncertainty in $\beta_0$ .	133
B.24	MC Simulations with uncertainty in $\chi_0$ .	134
B.25	Histogram of Final States with Uncertainty in $\chi_0$ .	134
B.26	MC Simulations with uncertainty in $m_0$ .	135
B.27	Histogram of Final States with Uncertainty in $m_0$ .	135
B.28	MC Simulations with uncertainty in <i>SFC</i> .	136
B.29	Histogram of Final States with Uncertainty in <i>SFC</i> .	136
B.30	MC Simulations with uncertainty in all states and parameters.	137
B.31	Histogram of Final States with Uncertainty in all States and Parameters.	137

# List of Tables

3.1	Definition of constants in Eq. (3.25) and Eq. (3.27) . . . . .	29
3.2	State Information for Initial-Final Conditions and Allowed Bounds. . . .	32
3.3	Control Input Information with Initial-Final Values and Constraints. . .	32
4.1	Legendre Polynomials with Different Orders . . . . .	35
4.2	Dependency Matrix $M_d$ . . . . .	44
5.1	State Information for the example OCP . . . . .	48
5.2	Control Input Information for the example OCP . . . . .	48
5.3	Summary of RS Coordinates for Different Scenarios. . . . .	57
5.4	Accuracy of the RS Computation Based on Number of Discretization Nodes. . . . .	60
6.1	Gains for the Lyapunov based PD Controller. . . . .	73
7.1	List of Uncertainties for MC Analysis. . . . .	81
7.2	Gains for the States. . . . .	83
7.3	Gains for the Control Inputs. . . . .	83
7.4	Results of the MC Analysis and Properties of the Ellipse. . . . .	89
9.1	Results of Set Interpolation with Associated Hausdorff Distance for $\Delta s$ Perturbations . . . . .	105
9.2	Results of Set Interpolation with Associated Hausdorff Distance for $2\Delta s$ Perturbations . . . . .	105

## LIST OF TABLES

---



# 1

## Introduction

Within the last decades, exploration of space was the driving factor and rationale for space research. Experiences obtained during this era yielded many spin-off technologies and these breakthroughs resulted in a high social and economic life standards. As the trend is shifting from exploration to exploitation phase, there are plans for more challenging missions like in orbit servicing of satellites, establishing bases on celestial bodies and space mining. In order accomplish the mission objectives, complex and innovative approaches are considered for encountering harsh space environment and resulting challenges.

Autonomous execution of defined tasks while handling uncertainties with available capabilities is necessary for the success of these missions. In particular, execution of autonomous and precise guidance and control (G&C) algorithms is considered to be one of the most critical and vital segment during mission phase, especially in missions which require landing on celestial bodies.

The increased capabilities of space technology and continuously developed mathematical methods improved the landing precision substantially. Future missions target a fully autonomous landing using new technologies, such as absolute and relative visual navigation using camera images and known landmarks on the surface of the body, hazard detection and avoidance (HDA) in regions with low illumination, autonomous trajectory generation and large angle divert capabilities for safe landing sites. These precision landing systems could be further developed to lead the way for a pinpoint landing (PPL) capability required for future space missions that depend on landing on the desired point. The maturation of this key technology will provide enhanced safety with high success rate of future space missions enabling the positioning of the lander directly in a region of high interest scientifically or economically.

Autonomous landing requires acquiring necessary information from the on-board sensors, processing the data and creating the control law in real-time. Such an autonomous control system steers the spacecraft from the current point to a desired landing point while taking performance indices such as propellant or time of flight into account. In order to achieve such objective, the spacecraft is equipped with several instruments like cameras, laser altimeters and LIDAR. The data obtained from these sensors gives

## 1. INTRODUCTION

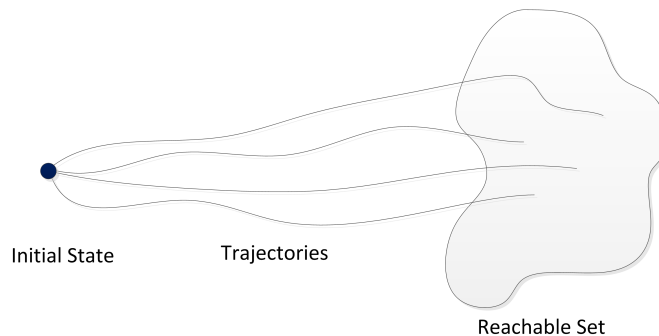
---

information about the position and the attitude of the vehicle, as well as surface features like the craters, density of the rocks and slope of the terrain. The surface properties are analyzed by HDA to obtain numerous risk maps.

However, these risk maps do not take into account the physical limitations and capabilities of the spacecraft. In order to ensure safety during landing, two factors should be considered. Firstly it must be identified if all candidate landing areas are achievable by the spacecraft considering the instant attitude and remained propellant. Second, if the predefined point is not suitable for landing, is the spacecraft able to land on another emergency point and implement other maneuver scheme. In order to answer these questions, a map of attainable and safe landing area that also considers the physical capabilities of the lander, actuators limitations and uncertainties is required. **Reachability analysis** meets the necessity of such objective by obtaining *Reachable Set* of the system.

### 1.1 Reachability

Reachable set is the set of all achievable states for a dynamical system starting from an initial condition with the admissible control inputs and possible uncertainties on the system parameters or initial states as illustrated in Figure 1.1.



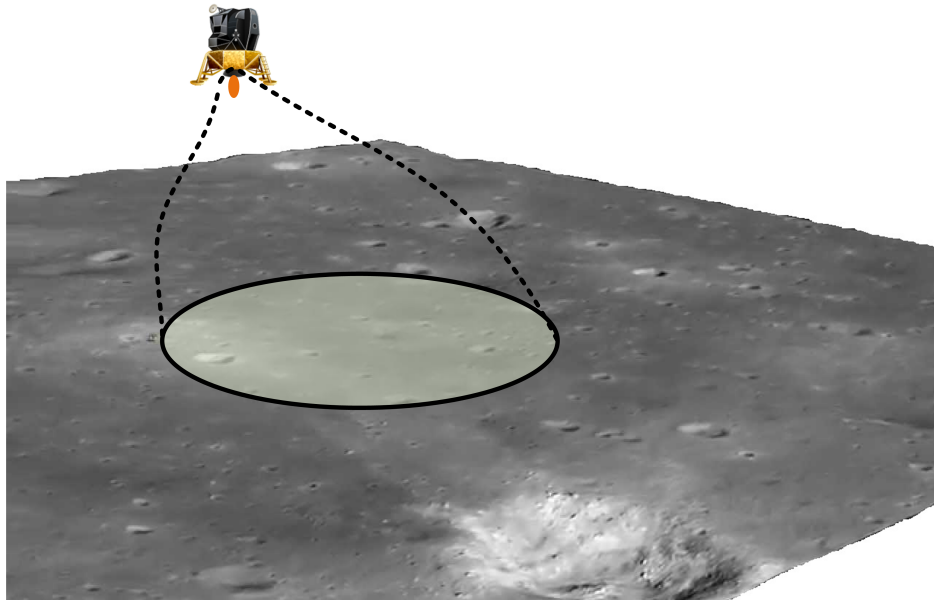
**Figure 1.1:** Illustration of a Reachable Set.

Reachability deals with the problem of whether from a given point, the system will eventually be able to reach another given point. At the end of reachability analysis, reachable sets which includes crucial information about the dynamical system of interest are obtained. These information could be used for different purposes:

- *Controller Design:* Reachability analysis can be used to find or tune controller parameters for given performance or safety indices<sup>1</sup>.
- *Mission Analysis:* Reachable sets can determine the evolution of all states of a given dynamical system. This could provide essential information about design considerations of a mission such as required control authority or avoidance of forbidden zones<sup>23</sup>.

- *State Estimation*: Set based observers could obtain the set of possible states by reachability analysis instead of using stochastic methods (e.g. Kalman filtering).

Reachability is classified into two categories depending on the formulation of the problem. For **forward reachability**, the aim is to find final set of states which start from a given set of initial states. For example, the forward reachability analysis yields reachable sets which shows the attainable landing area for a lunar lander for a given initial condition, nonlinear system dynamics, constraints and uncertainties as shown in the Figure 1.2.



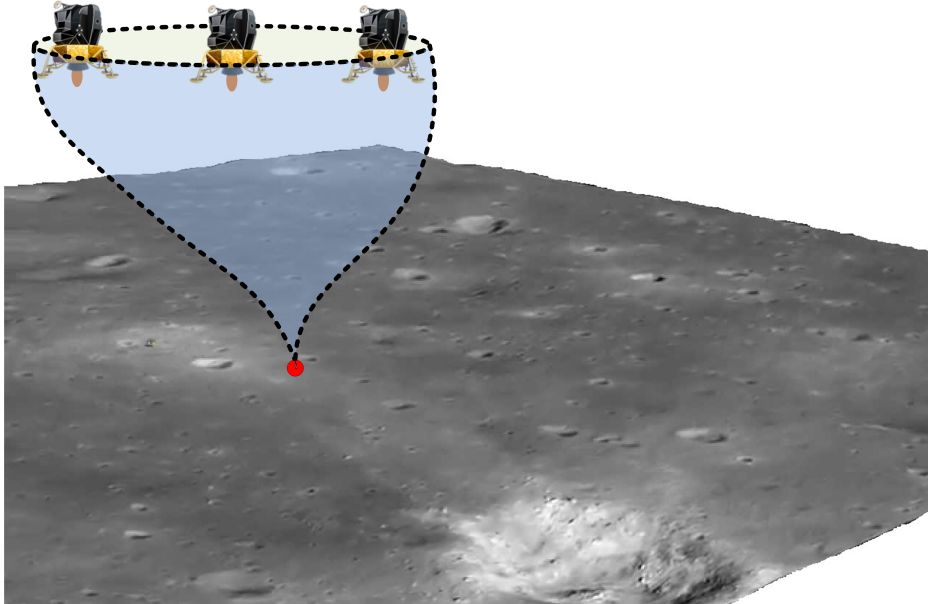
**Figure 1.2:** An example forward reachable set of a Lunar lander. The lander can attain any point inside the shaded area.

On the other hand, in **backward reachability** the objective is to find set of states which trajectories start and achieve a given target point. Similarly, for the same example, the landing point could be reachable with the given dynamical system if the initial state is inside the backward reachable set. Figure 1.3 shows the graphical representation of backward reachable set for a lunar lander.

There are different methods for obtaining reachable sets and have both drawbacks and advantages in terms of accuracy of the solution, necessity of the computational power or capability with handling the suitable class of systems. In this study only nonlinear continuous system dynamics with equality and inequality constraints are considered.

## 1. INTRODUCTION

---



**Figure 1.3:** An example backward reachable set of a Lunar lander. The lander can achieve a successful landing starting the maneuver from any point inside the shaded region.

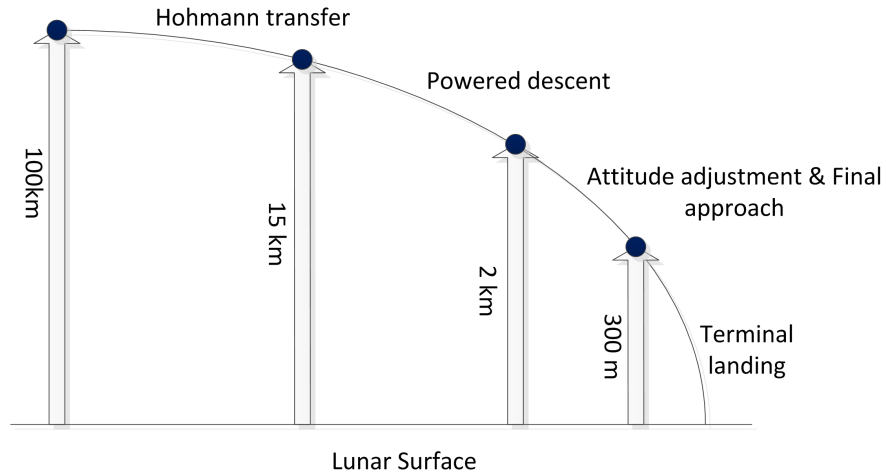
## 1.2 Safe Landing Area Determination for Soft Landing Problem

Soft landing is an interesting problem for reachability analysis, and results of the analysis could yield valuable information which could be used during the design of the related technologies. A soft landing is smooth and steady landing of spacecraft that does not result in damage to the vehicle or anything on-board. The soft landing problem characterizes the safety considerations by taking the safe landing conditions into account.

The reachable sets are considered for the terminal landing phase of a generic Moon mission. In this scenario, the spacecraft starts a landing maneuver from its lunar circular orbit. After a Hohmann transfer, the lander moves on an elliptical descent orbit from an altitude of 100 km to a perigee of 15 km. Powered descent starts at an altitude of 15 km and continues until 2 km. During the powered descent the lander slows down from orbital velocities using the main engine. At the approach stage, an attitude adjustment maneuver is performed and translational velocities are further decreased for terminal landing phase. The terminal landing phase of the landing maneuver is supposed to start at an altitude of 300 m.

Future Moon missions require the landing procedures to be executed autonomously. In order to increase the performance of the landing algorithm, optical navigation systems are used. Especially using high resolution images of Moon surface obtained by NASA's Lunar Reconnaissance Orbiter<sup>4</sup>, hazards could be cataloged easily to be further used

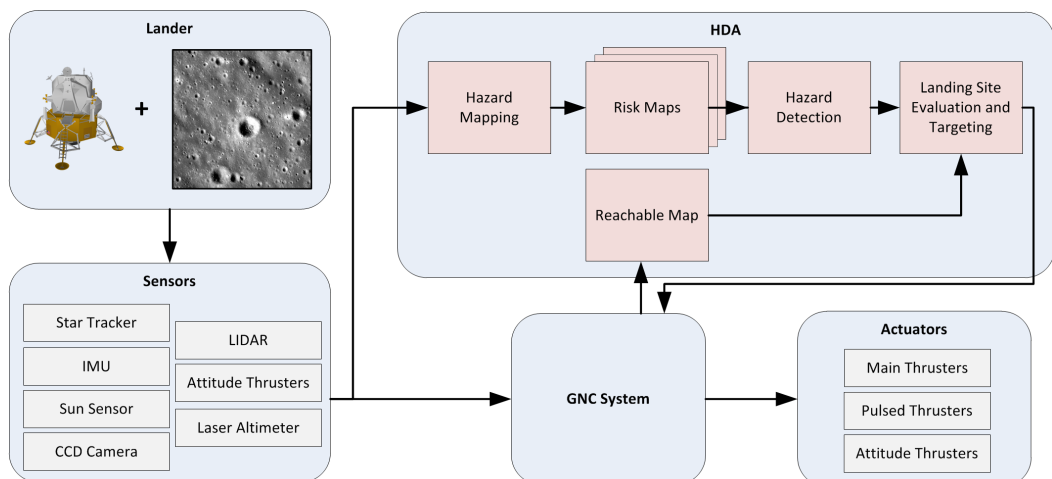
## 1.2 Safe Landing Area Determination for Soft Landing Problem



**Figure 1.4:** A generic mission profile for Lunar lander. (not scaled)

during landing. These risk maps include information about the slopes, craters, roughness of the surface and shadows. HDA algorithms sense the candidate landing area, and characterize the surface properties of the region in real-time. Finally, the system determines whether the chosen landing site is safe. If the safe landing conditions are not met, the control system steers the lander to another safe landing site.

Hazard avoidance algorithms compute the landing regions which satisfies most of these safety considerations. In addition, reachability analysis return the reachable landing area of the lander which takes the rest of the safe landing conditions into account. The fusion of these two different data sets yields the final safe landing area for landing on celestial bodies as shown in the Figure 1.5.



**Figure 1.5:** Hazard Detection and Avoidance System for the Landing Phase.

The final reachable set computations could increase the performance of the guidance

## 1. INTRODUCTION

---

and control algorithm in certain ways and could be used as an analysis tool for mission design or assess the performance of the guidance and control methods. The research develops reachability analysis algorithm while targeting the following research questions:

- What is the theoretical maximum reachable landing area of the Lunar lander for given constraints and mission scenario?
- How does the selected guidance and control method influences the reachable landing area?
- How does the characteristics of the reachable landing area change within the presence of uncertainties in the system parameters or initial state error?

The outcome of the reachability analysis algorithm could be used for different purposes. The algorithm computes reachable sets for different uncertainties in initial states or system parameters. This provides information about the diversion capabilities of the lander from the nominal landing point. If the selected region of interest is found to be unsafe, the another candidate landing point could be selected inside the reachable area. Guidance algorithm could re-plan the new trajectory for the new landing point.

In addition, reachable maps are obtained by computing the propellant optimal trajectories. It also considers the extra propellant consumed during diversion maneuvers. A cost function for HDA could be set such that the candidate landing point is selected based on safety considerations and propellant consumption. Finally, although the proposed algorithm is not real-time capable, it could compute the reachable sets off-line and with a proper approximation method, reachable sets could be approximated in real-time with acceptable approximation errors.

### 1.3 Scope and Summary of Contributions

The scope of the work is limited to obtaining reachable sets for the given nonlinear continuous dynamical systems. The focus is on obtaining safe landing area of a lunar lander during a generic Moon mission.

This work has several contributions for both to the guidance and control fields. Basically, it derives a nonlinear dynamical model for the lunar lander with varying and steerable throttle. In addition, it develops a guidance algorithm for the given system dynamics of a lunar lander to find feasible landing trajectories using optimal control. The infinite dimensional OCP is transcribed into finite dimensional NLP problem using Pseudo-spectral methods. It also designs a Riccati-equation-based feedback controller for the main guidance method.

The Apollo-era polynomial based guidance algorithm is used for comparison study. Exact linearization is used to invert the nonlinear system dynamics. A PD-like controller stabilizes the system and attenuates the errors due to model inversion or uncertainties on the initial states or system parameters.

### 1.3 Scope and Summary of Contributions

---

The reachable sets obtained by these two methods are compared using several quality metrics. In addition, a novel method is introduced to create reachable sets based on existing multivariate trajectory interpolation techniques.

In mission phase, on-board sensors acquire measurement data with uncertainty. Therefore the initial state error or uncertainty in the system parameters need to be considered. In this case, the robustness of the developed algorithm is proven by MC analysis. The sources for the uncertainties are identified. During the MC simulations, initial states and system parameters are sampled using Gaussian distribution and attainable landing area of the closed loop system are obtained within the presence of uncertainties in order to analyze the impact of the uncertainties on the RS. Unscented transform is used to propagate uncertainties in a more computationally efficient way compared to MC simulations. Using the transformed mean and covariance, a probability map of the respective point is created for the associated uncertainties. The probability density function with proper bounds is integrated and the resulting probability maps are compared with the probability maps from MC simulations.

In this study, considerations about the computational power necessities are neglected although polynomial approach is shown to be real time capable for generating feasible trajectories. The design parameters are fixed and no optimization procedures are carried out for the parameters of the feedback controllers. In addition, creation and synthesis of the various feature maps are not included in the algorithm development.

During the study the following tools were developed for creating the reachable sets:

- Development of real-time capable polynomial guidance algorithm based on model inversion.
- Development of reachable set computation tools in Matlab.
- Comparison of reachable sets obtained with two different methods.
- Reachable set computation for closed loop systems with uncertainties.
- Comparison of MC simulations with probability maps generated with unscented transform.

Moreover, this research contributed to the development of DLR's in-house trajectory optimization tool SPARTAN (Shefex-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis). The following papers related to this work were published in whole or in part by the author:

1. *Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis*<sup>5</sup>
2. *Safe Landing Area Determination for a Moon Lander by Reachability Analysis*<sup>6</sup>
3. *Safe Landing Area Determination for a Moon Lander by Reachability Analysis*<sup>7</sup>

## 1. INTRODUCTION

---

4. *Real-time Capable Nonlinear Model Predictive Controller Design for The Upper Stage of a Launch Vehicle*<sup>8</sup>
5. *SPARTAN: An Improved Global Pseudospectral Algorithm for High-fidelity Entry-Descent-Landing Guidance Analysis*<sup>9</sup>
6. *Real-time Capable Nonlinear Model Predictive Controller Design for The Upper Stage of a Launch Vehicle*<sup>10</sup>
7. *Exact Hybrid Jacobian Computation for Optimal Trajectory Generation via Dual Number Theory*<sup>11</sup>
8. *Attainable Landing Area Computation of a Lunar Lander with Uncertainty by Reachability Analysis*<sup>12</sup>

### 1.4 Outline of Thesis

This dissertation is arranged as follows. Chapter 1 provides motivation and background information for this work and lists this work's contributions to the field.

Part I is devoted to the basic concepts and dynamical models and includes Chapters 2-3. Chapter 2 discusses existing methodologies for reachable set computation methods. It also summarizes the research about trajectory optimization and lunar lander guidance methods. Chapter 3 presents the derivation of the 2-D and 3-D nonlinear dynamical model for the lunar lander.

Part II includes Chapters 4-6 which are related to the main algorithm development and set computations for various scenarios. Chapter 4 contains sections for discretization of the OCP to the related NLP using PSM. Chapter 5 includes the set computation and numerical results for the nominal cases. The results obtained in this chapter is considered as the reference solution for the rest of the thesis. Chapter 6 provides information for the polynomial based Apollo guidance method and tracking controller using model inversion. In this chapter, RS obtained by Apollo guidance method is compared with reference solution.

Part III includes Chapters 7-10 and is related to handling uncertainties and parameter variation. Chapter 7 presents MC analysis and discussion about selection of uncertainty parameters and quality metrics for the results. In Chapter 8, an unscented transform based method is introduced for the computation of probability maps. Chapter 9 introduces multivariate interpolation based reachable set interpolation and assessment of the results.

Finally, Chapter 10 concludes the thesis and provides a summary of the study. Figure 1.6 illustrates the structure of the thesis.



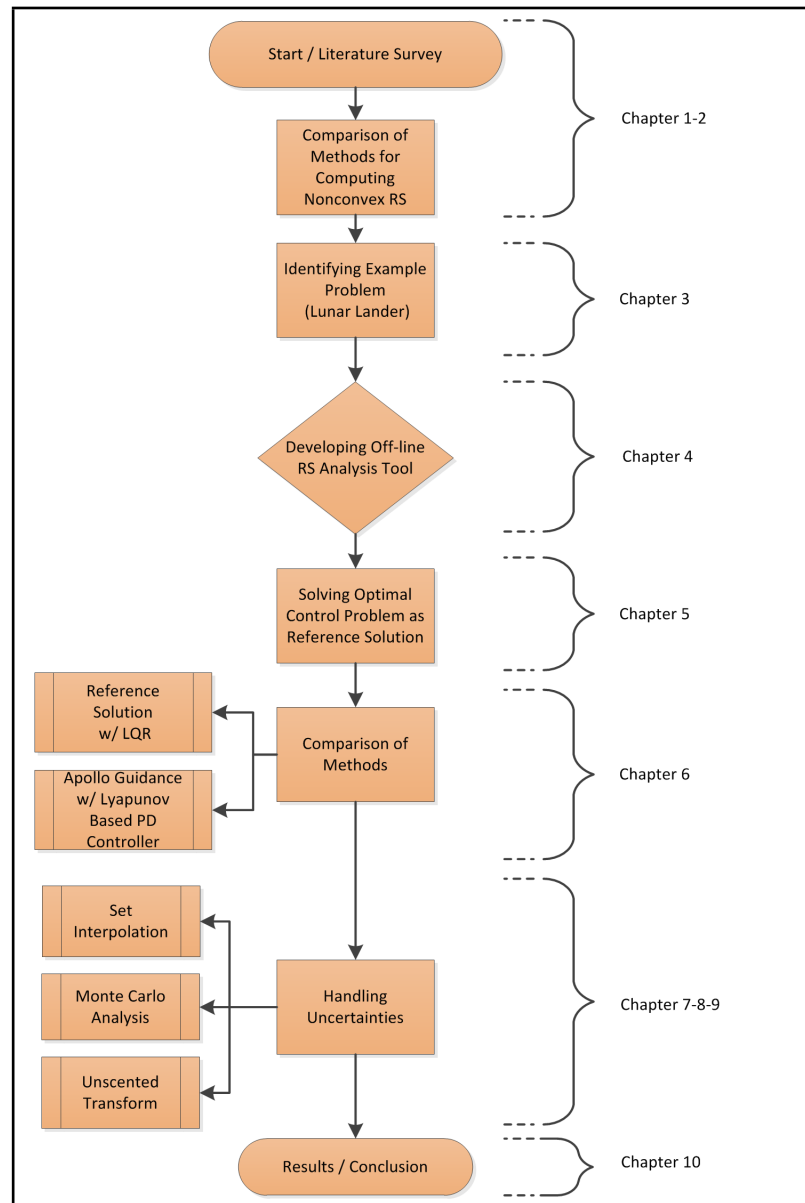


Figure 1.6: The Structure of the Thesis

## Part I

# Basic Concepts and Dynamical Models

## 2

# State of Art and Problem Statement

There are variety of methods in the literature that have been developed for computing the reachable sets. In order to understand the difference of these methods, in addition to *reachability*, *reachable set* and should be defined properly.

### 2.1 Reachable Set

Reachability analysis is a comprehensive methodology providing information about the evolution of states of a given dynamical system. Consider the nonlinear dynamical system represented by the state equation

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)), \quad t \in \mathbb{I}, \\ x(t_0) &= x_0 \in X,\end{aligned}\tag{2.1}$$

where  $X \subset \mathbb{R}^n$  denotes the state space and  $U \subset \mathbb{R}^m$  the input space.  $x_0 = x(t_0)$  is the state at initial time  $t_0$  while  $x_f = x(t_f)$  denotes the final state. The control input is contained in the space of admissible control signals, that is  $u(t) \in \mathcal{U} = L^\infty([t_0, t_f], U)$ . For each point in time, the solution or state trajectory  $x(\cdot) : \mathbb{R}_+ \rightarrow X$  of Eq. (2.1) is an element of the state space  $x(\cdot) \in X$ .

**Definition 1.** *Given  $x_0$  and the set of all admissible  $\mathcal{U}$ , the reachable set of Eq. (2.1) at time  $t_f$  is defined as*

$$\mathcal{R}_{t_f}(x_0) = \{x_f \in X \mid \exists u \in \mathcal{U} \wedge \exists x(\cdot) \in X \text{ s.t. Eq. (2.1) holds} \wedge x_f = x(t_f)\}.\tag{2.2}$$

*Similarly, if final time  $t_1$  is in the interval  $\mathbb{I}$ , then reachable set is defined as*

$$\mathcal{R}_I(x_0) = \{x_1 \in X \mid \exists t_1 \in \mathbb{I} \wedge \exists u \in \mathcal{U} \wedge \exists x(\cdot) \in X \text{ s.t. Eq. (2.1) holds} \wedge x_1 = x(t_1)\}.\tag{2.3}$$

## 2. STATE OF ART AND PROBLEM STATEMENT

---

The system is assumed to be deterministic, i.e. produces the same results for the same inputs. If the closed-loop dynamics of a feedback control system is considered, control input  $u$  is a function of state  $x$  and time  $t$  and has the following form

$$u(t) = g(x(t)) \quad (2.4)$$

Therefore, the nonlinear system in Eq. (2.1) attains the following form

$$\dot{x} = f(x(t), g(x(t))). \quad (2.5)$$

Considering the formulation above, reachable set equals to the set of all states attainable by the given dynamical system with the admissible set of inputs and the predefined equality and inequality constraints. Similarly, reachability deals with the problem of whether from a given point, the system will eventually be able to reach another point of interest, without active control constraints if a closed loop system is considered.

### 2.2 Computation of Reachable Sets

Reachability analysis is a strong tool for verification and determination of the safety of the system and has been applied to several real world problems including safe landing envelope of an aircraft<sup>13</sup>, landing corridor of a reusable launch vehicle<sup>14</sup>, re-entry problems<sup>15</sup>, for planetary landing<sup>16,17</sup>, collision avoidance for aircraft<sup>18</sup> and cars<sup>19</sup>, unmanned aerial vehicles(UAVs)<sup>20</sup> and even biological systems<sup>21</sup>.

There are various methodologies for obtaining the RS for different classes of systems. The methodologies have a balance between the accuracy of the solution and the computation power required. RS computation suffers from a concept called *curse of dimensionality*. As the dimension of the state space increases, the computational power required to compute the RS crosses beyond practical limits. Instead of computing RS for the full state space, some algorithms compute a subset of the RS<sup>22</sup>.

Another classification lies behind the accuracy of the results. In terms of accuracy, methods can be classified into two categories. A technique is said to be **convergent** if it can be proved that approximation of the reachable set converges to the true reachable set as the approximation is refined by using finer discretization steps. On the other hand, in **overapproximation** methods are implemented such that any approximation error introduced during computations makes the reachable set even larger<sup>2</sup>.

Obtaining RS for linear systems is well understood and efficient algorithms were developed recently<sup>23,24,25,26,27</sup>. RS for nonlinear systems are generally nonconvex and only few methods are available for the computation of RS for these class of systems. For general nonlinear systems, global optimization techniques<sup>28</sup>, face lifting methods<sup>29</sup> or optimization based methods<sup>30</sup> can be applied. The works on reachability analysis of linear and nonlinear continuous systems can be extended to hybrid systems considering the switching in the continuous dynamics. However considerations related to hybrid systems are not within the scope of this thesis. In general, three basic approaches for reachability set computation can be classified as

- Set Valued methods<sup>2,31</sup>
- Level Set methods<sup>32</sup>
- Optimization based algorithms<sup>30</sup>

A family of algorithms called *level set methods* have been designed specifically to compute approximations to reachable set of a dynamical system by obtaining the viscosity solution for time dependent PDEs with continuous initial conditions and Hamiltonians. In level set methods, the surface of the reachable set is characterized by an implicit PDE equation. These PDEs might be difficult to solve either analytically or numerically. The solutions may not converge where the derivatives become discontinuous in finite time.

Based on the numerical solution of PDEs, another way to differentiate these two classes of reachable set methods is as Eulerian or Lagrangian approximations. An *Eulerian* approach approximates the solution's values at the nodes of a fixed grid using finite difference, finite element or finite volume techniques. In contrast, a Lagrangian approach follows the flow of the solution by computing along trajectories of the dynamics, a process that is equivalent to solving a PDE by the method of characteristics.

Level set methods computes the reachable set for all states of the given dynamical system. Therefore, in the cases where a subset of the reachable set important, the full reachable set is projected into two or three dimensional subspace.

The level set methods are limited to systems where the PDE representing the surface of the reachable set could be solved numerically. At the best case the solutions are obtained for dynamical systems with four states. For higher order systems, the discretization step in spatial coordinates becomes smaller and the computations become numerically unstable.

The second class of methods describes the reachable set as the solution of a differential inclusion or a related integral-funnel equation. These classes of methods are computationally tractable for nonlinear systems. Moreover, these approaches are designed to provide an approximation of the reachable set and the convergence to the RS is not guaranteed. Figure 2.1 illustrates the flow tube approximation of Van Der Pol system. The details of the computation of this figure is included in Appendix A.

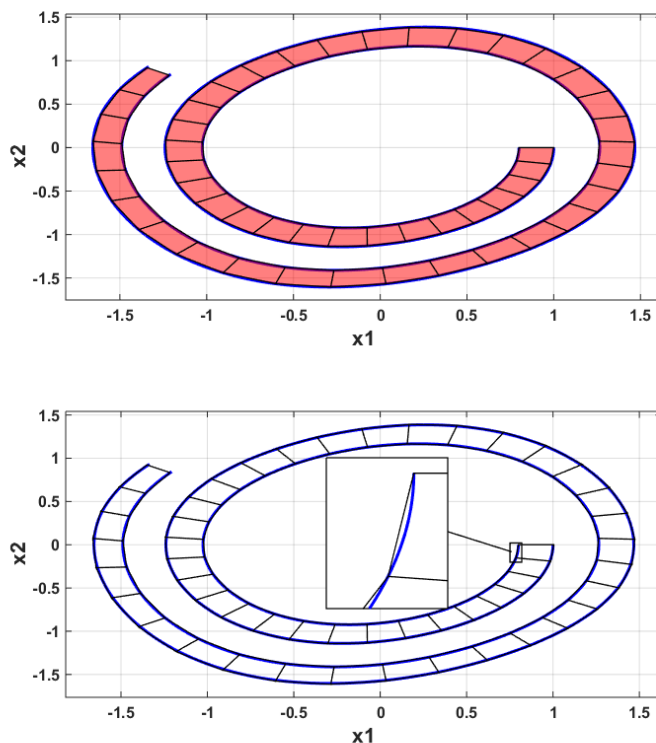
A large portion of the work in literature considers the reachable set under several assumptions, typically dealing with linear ODEs or discrete time models. Several geometric representations for linear continuous systems have been investigated to obtain reachable sets. Polytopes<sup>28</sup>, ellipsoids<sup>33 31</sup>, oriented rectangular hulls<sup>34</sup>, zonotopes<sup>2 25</sup>, or support functions<sup>35</sup> are used for the representation of reachable sets.

The algorithms for linear continuous systems can be applied to general nonlinear systems or hybrid systems. This is achieved by conservative linearization, i.e. by considering the linearization error as an additional uncertain input of the linearized system.

Set valued methods suffer from the curse of dimensionality. As the dimension of the state space increases, the computations increase exponentially, making the computational

## 2. STATE OF ART AND PROBLEM STATEMENT

---



**Figure 2.1:** Flow Tube Approximation of Van Der Pol System

requirement higher even for linear systems. Yet another problem with the set valued methods are wrapping effect. The error obtained for the computation of reachable set is accumulated in the next computation step. Therefore, for long time horizons the set valued methods may approximate the reachable sets in an over-conservative way.

In addition to above classification of the available methods, barrier certificate are used to find unreachable region of the state space<sup>36</sup>. In the state space, a barrier certificate guarantees that all system trajectories stays in a region and does not cross it. Thus unreachable region is found without computing the flow explicitly.

### 2.3 Reachable Set Computation by Optimal Control

Optimization plays an important role in the computations of RS and is used in set valued methods, level set methods and finding barrier certificates. However, apart from the previous discussion, the method proposed in this thesis purely relies on the optimal control. In this set of techniques, an objective function is defined which tries to find an control input to the system to steer it to predefined set of states. This is achieved by transcribing continuous optimal control problem (OCP) into finite dimensional

## 2.3 Reachable Set Computation by Optimal Control

---

nonlinear programming (NLP) problem which is solved by existing solvers. In some cases the computations can be executed in parallel which results in high speedup<sup>37</sup>. In addition, depending on the properties of the RS, both convex and nonconvex RS can be computed by applying different approaches. However main disadvantage of the problem is the need for high computational power.

### OCP for Convex Reachable Sets

Support functions are used to compute convex RS<sup>13</sup>. In this method, the aim is to maximize the value of a state in a defined direction. By successive iterations, boundary of the convex RS is found. Since the RS is convex, this implies that the region within the convex set is also reachable.

Denote by  $C(\mathbb{R}^n)$  the set of all nonempty convex compact sets in  $\mathbb{R}^n$  and let  $c \in C(\mathbb{R}^n)$  and  $l \in \mathbb{R}^n$ . Then considering the definition of support function

$$\rho(l, C) = \sup_{c \in C} \langle l^T c \rangle \quad (2.6)$$

the set of supporting points of  $C$  in direction  $l$  is defined by

$$Y(l, C) := c \in C | l^T c = \rho^*(l, C). \quad (2.7)$$

Thus, to calculate a supporting point  $x(t_f)$  on the boundary of the reachable set  $\mathcal{R}_{[t_f, t_0]}(x_0)$  in a fixed direction  $l$  we have to find an admissible control function  $u(t) \in \mathcal{U}$  that maximizes the functional  $y \rightarrow l^T y$  (resulting in the support function  $\rho(l, \mathcal{R}_{[t_f, t_0]}(x_0))$  as optimal value). This constitutes the following special optimal control problem of Mayer type:

$$\begin{aligned} \max \quad & l^T x(t_f) \\ \text{s.t.} \quad & \dot{x} = f(x(t), u(t)), \text{ a.e. in } [t_0, T] \\ & u(t) \in U \text{ a.e. in } [t_0, T] \\ & x(0) \in X_0 \\ & x(\cdot) \text{ is the corresponding solution to } u(\cdot). \end{aligned} \quad (2.8)$$

If the reachable set is compact and convex, reachable set is

$$\mathcal{R}_{[t_f, t_0]}(x_0) = \text{Ch}\{x^*(t_f; l) | l \in \mathbb{R}^n, \|l\|_2 = 1\}. \quad (2.9)$$

This approach is only suitable for convex sets. Main assumption is that the RS is connected and convex. So only by obtaining boundary of the RS implies that any point inside the boundary is reachable by the dynamical system. However in this method, no information is provided about the value of states at final time (i.e.,  $x(t_f)$ ) inside the boundaries.

## 2. STATE OF ART AND PROBLEM STATEMENT

---

### OCP for Nonconvex Reachable Sets

In this approach equidistant grid points are projected onto the RS. Similar to previous method, a distance function is defined and at each grid point, an optimal control problem (OCP) is solved<sup>30 38 39</sup>. Advantage of this method is to get rid of cumbersome set operations (Minkowski sum, convex hull, conversion of set properties). Despite the computation power requirement, the method is applied to real world problems<sup>19</sup>.

As discussed previously, this method also suffers from the curse of dimensionality. As the size of space state dimension increases, the number of grid points needed to capture the behaviour of the system increases exponentially. However, with the proposed method a subset of RS can be computed to reduce the need for these intensive computations. In order to find the maximum attainable landing area, grid points are projected onto the plane defined by downrange and crossrange.

In order to determine the approximated RS, an associated OCP should be solved to find control inputs that steer the system from the initial state to the final grid points. If a feasible solution is found, the point is said to be reachable. Otherwise, if the solution cannot be found, the candidate point is excluded from the set. Instead, final states of the associated problem are included in the set. In other words, the solution which minimizes the distance between the candidate point and final states are taken into account. The set which encloses all these final points constitutes the RS. The OCP is formulated as

$$\begin{aligned}
 \min \quad & \frac{1}{2} \|x(t_f) - g_s\|_2^2 \\
 \text{s.t.} \quad & \dot{x} = f(x(t), u(t)), \text{ a.e. in } [t_0, t_f] \\
 & u(t) \in \mathcal{U} \text{ a.e. in } [t_0, t_f] \\
 & x_0 \in X_0.
 \end{aligned} \tag{2.10}$$

where  $X_0 \subset \mathbb{R}^n$  denotes the state space,  $\mathcal{U} \subset \mathbb{R}^m$  denotes the input space,  $t_0$  denotes initial time,  $t_f$  denotes the final time. The approximation of the reachable set can be then formulated as

$$\mathcal{R}_{[t_0, t_f]_s}(X_0) = \bigcup_{g_s \in G_s} \{x^*(t_f; g_s)\} \tag{2.11}$$

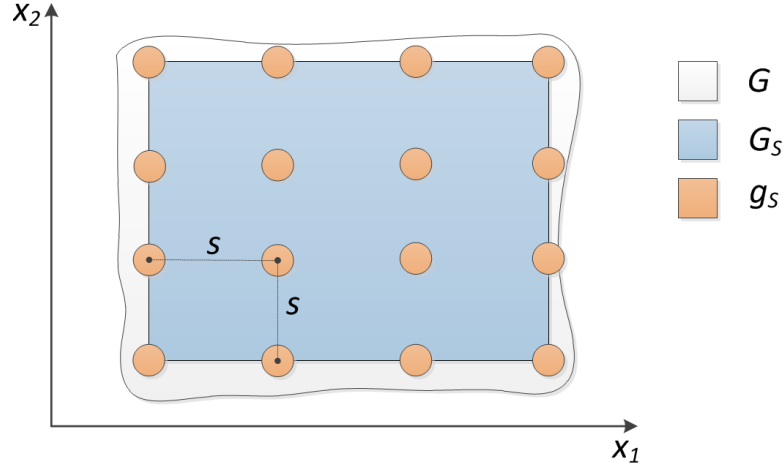
where  $\mathcal{R}_{t_f}(x_0)$  denote the set of all possible states  $x$  at time  $t_f$  given an initial condition  $x_0$  and control in  $\mathcal{U}$ ,  $\mathcal{R}_{[t_0, t_f]_s}(X_0)$  denotes the *reachable set* (RS) of states for Eq. (2.1), and for given  $x_0$ ,  $x^*(\cdot; g_s)$  is the corresponding solution to  $u^*(\cdot; g_s)$ . The main idea is as follows:

- Choose a region  $G \subseteq \mathbb{R}^n$  and discretize  $G$  by a grid  $G_s \subset G$  with step-size  $s$ , as shown in the Figure 2.2
- For every  $g_s \in G_s$  solve the resulting optimal control problem  $OCP(g_s)$  given by Eq. (2.10)



## 2.3 Reachable Set Computation by Optimal Control

In the case of free final time, the definition is same but the methodology for the computation of the resulting OCP changes. This change in the computation method is further explained in Section 4.2. In the second case,  $t_f$  is also an optimization variable. State information at  $t_f$  is used to obtain approximation of the RS.



**Figure 2.2:** Discretization of a Subset of the State Space

### Indirect Methods

In an indirect method, Pontryagin's minimum principle and the calculus of variations<sup>40 41</sup> is applied to determine the first-order necessary conditions for an optimal solution. Applying the calculus of variations transforms the optimal control problem to a Hamiltonian boundary-value problem (HBVP). The solution to the HBVP is then approximated using one of the various numerical approaches. Basic approaches for solving the HBVP are shooting, finite difference<sup>42</sup>, multiple shooting<sup>43</sup> and collocation<sup>44</sup>. Although using an indirect method has the advantage that a highly accurate approximation can be obtained and that the proximity of the approximation to the optimal solution can be established, indirect methods have several disadvantages.

- Implementing an indirect method requires that the complicated first-order necessary optimality conditions be derived.
- The indirect methods require that a very good initial guess on the unknown boundary conditions must be provided. These guesses include a guess for the costate which is a mathematical quantity inherent to the HBVP. Because the costate is a non-intuitive and non-physical quantity, providing such a guess is difficult.
- Whenever a problem needs to be modified (e.g., adding or removing a constraint), the necessary conditions need to be reformulated.

## 2. STATE OF ART AND PROBLEM STATEMENT

---

- For problems whose solutions have active path constraints, a priori knowledge of the switching structure of the path constraints must be known.

### Direct Methods

In a direct method, the continuous functions of time (the state and/or the control) of the optimal control problem are approximated and the problem is transcribed into a finite-dimensional NLP<sup>45</sup>. The NLP is then solved using well developed algorithms and software<sup>46 47 48 49</sup>. Well-known direct methods can be classified and illustrated in Figure 2.3.

In the case where only the control is approximated, the method is called a control-parameterization method. When both the state and the control are approximated, the method is called a state and control-parameterization method. Direct methods overcome the disadvantages of indirect methods because the optimality conditions do not need to be derived, the initial guess does not need to be as good as that required by an indirect method, a guess of the costate is not needed, and the problem can be relatively easily modified. Direct methods, however, are not as accurate as indirect methods, require much more work to verify optimality, and many direct methods do not provide any information about the costate. Many different direct methods have been developed. The two earliest developed direct methods for solving optimal control problem are the direct shooting method and the direct multiple-shooting method<sup>50</sup>. Both direct shooting and direct multiple-shooting methods are control parameterization methods where the control is parameterized using a specified functional form and the dynamics are integrated using explicit numerical integration (e.g., a time-marching algorithm). A direct shooting method is useful when the problem can be approximated with a few number of variables.

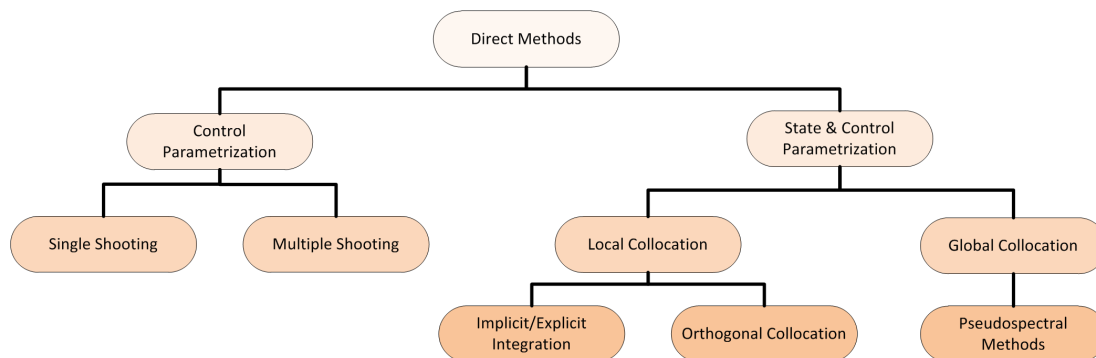


Figure 2.3: Classification of Direct Methods

As the number of variables used in a direct shooting method grows, the ability to successfully use a direct shooting method declines. In the direct multiple-shooting method, the time interval is divided into several subintervals and then the direct shooting method is used over each interval. At the interface of each subinterval, the state continuity condition is enforced and the state at the beginning of each subinterval is

---

## 2.3 Reachable Set Computation by Optimal Control

a parameter in the optimization. The direct multiple-shooting method is an improvement over the standard direct shooting method as the sensitivity to the initial guess is reduced because integration is performed over significantly smaller time intervals. Both the direct shooting method and the direct multiple-shooting method, however, are computationally expensive due to the numerical integration operation and require a priori knowledge of the switching structure of inactive and active path constraints. A well-known computer implementation of direct shooting methods is POST<sup>51</sup>.

Another approach is that of direct collocation methods<sup>52 53 54</sup>, where both the state and the control are parameterized using a set of trial (basis) functions and a set of differential-algebraic constraints are enforced at a finite number of collocation points. In contrast to indirect methods and direct shooting methods, a direct collocation method does not require a priori knowledge of the active and inactive arcs for problems with inequality path constraints. Furthermore, direct collocation methods are much less sensitive to the initial guess than either the aforementioned indirect methods or direct shooting methods. Some examples of computer implementations of direct collocation methods are SOCS<sup>55</sup>, DIDO<sup>56</sup>, GPOPS<sup>57</sup> and SPARTAN<sup>9</sup>.

The two most common forms of direct collocation methods are local collocation<sup>45</sup> and global collocation<sup>52 53 58 54</sup>. In a direct local collocation method, the time interval is divided into subintervals and a fixed low-degree polynomial is used for approximation in each subinterval. The convergence of the numerical discretization is achieved by increasing the number of subintervals.

Direct local collocation leads to a sparse NLP with many of the constraint Jacobian entries as zero. Sparsity in the NLP greatly increases the computational efficiency. However, the convergence to the exact solution is at a polynomial rate and often an excessively large number of subintervals are required to accurately approximate the solution to an optimal control problem resulting in a large NLP with often tens of thousands of variables or more. In a direct global collocation method, the state and the control are parameterized using global polynomials. In contrast to local methods, the class of direct global collocation methods uses a small fixed number of approximating intervals (often only a single interval is used). Convergence to the exact solution is achieved by increasing the degree of polynomial approximation in each interval.

### 2.3.1 Moon Landing

This thesis considers terminal landing phase for a lunar lander as an example problem. A lander test platform developed by DLR is illustrated in Figure 2.4. The aim is to develop a trajectory-based RS analysis algorithm which can be used to assess the performance of the different guidance and control methods for the desired index. The thesis also aims to identify the advantages, disadvantages, and limitations of the proposed method, demonstrating the robustness while testing with different scenarios.

In the literature there are various methods for the soft landing problem. The basic soft landing guidance problem is defined as finding a trajectory for the lander between any given state at the beginning of terminal landing maneuver to a desired terminal state without violating any state constraints (e.g., reaching the target without

## 2. STATE OF ART AND PROBLEM STATEMENT

---

flying subsurface) and control constraints (e.g., actuator saturation) with the available propellant. These methods have different characteristics in terms of handling constraints, computing propellant optimal trajectories and computational requirements.

Due to limited capabilities of Apollo-era flight computers, explicit polynomial based guidance methods were used extensively during the early space missions. The gravity turn was the first active guidance law to be used for soft-landing of spacecraft by NASA in the Surveyor program<sup>59</sup>. The method relies on solving a quadratic equation for the acceleration in order to ensure zero velocity on the surface<sup>60</sup>.

Several methods that solve a boundary value problem analytically and are capable of computing the trajectory in-flight have been developed<sup>61,62</sup>. The Apollo guidance system is an adapted version of these class of methods and was used on six successful Lunar landings with the strongest heritage<sup>63</sup>. The Apollo landing is composed of the approach phase, braking phase and the terminal descent phase. In the braking and approach phase the aim is to reduce the velocity of the vehicle from orbital speed. Once these phases are completed, the velocities in the downrange, crossrange and altitude are close to their final values. During the terminal landing phase, the G&C system steers the vehicle from low gate to the landing point. These phases use this same guidance algorithm with differences in the reference constants.

Modified versions of Apollo guidance methods were developed to compute propellant optimal trajectories and use different acceleration profiles<sup>64,65,66,67</sup>.

Numerical methods solve the same guidance problem using optimal control while satisfying equality and inequality constraints. This makes such guidance algorithms very robust and propellant optimal with the cost of more computational resources. However, with the recent advances in space technology, optimal solutions can be computed for the soft-landing problem in real-time with certain assumptions and simplifications while satisfying constraints.

Gradient-based numerical optimizations methods have been proposed to the guidance problem<sup>68</sup>. Although computationally expensive, classical numerical optimization techniques are also applied to solve the guidance problem for the computation of the reachable landing area<sup>17</sup>. The hybrid problem is solved for lunar landers with non-throttled engines<sup>69</sup>. Some methods obtains off-line trajectory libraries and a feasible solution is approximated by interpolating optimal trajectories<sup>70</sup>. The soft-landing problem is considered using Pseudo-Spectral methods<sup>5,6</sup>.

Açıkmeşe and Ploen published a promising guidance mode based on convex optimization theory using "lossless convexification"<sup>71,72</sup>. The original guidance problem is formulated as a second-order cone problem (SOCP). The convex approximation makes it very computationally efficient with real-time capabilities. The method has received lots of attention in the past decade, and has been used in the "fly-away" phase of Curiosity rover landing on Mars surface.

## 2.3 Reachable Set Computation by Optimal Control

---



**Figure 2.4:** EAGLE Lander Test Platform of DLR

### 3

## Nonlinear Dynamical Model

In this section, the equations of motion of the nonlinear dynamical system is derived. The nonlinear dynamical system has the general form

$$\dot{x} = f(x(t), u(t)) \quad (3.1)$$

where  $x = [r \ v]^T \in \mathbb{R}^6$  and  $u \in \mathbb{R}^3$  is a control vector, and

$$f(x(t), u(t)) = \begin{bmatrix} v \\ g(r) + a(x, u) \end{bmatrix} \quad (3.2)$$

where  $g$  is the gravitational acceleration and the vector function  $f(\cdot)$  is continuously differentiable. The spacecraft position and velocity vectors represented in lunar centered inertial coordinates and are given by

$$\begin{aligned} \mathbf{r} &= [r_1 \ r_2 \ r_3]^T \\ \mathbf{v} &= [v_1 \ v_2 \ v_3]^T \end{aligned}$$

Moon centered inertial coordinates define an inertial reference frame centered at the lunar center of mass and described by three orthogonal unit vectors. The equations of motion are represented in Moon centered inertial frame (MCI) for two reasons. First reason is to have a valid method which provides the same results independent of the location of the landing point. Second reason is the necessity of non-rotating frame assumption for the polynomial guidance method. In this approach, the translational motion of the vehicle is demonstrated in a non-rotating frame with flat surface assumption. The accelerations due to gravity and thrust are given by the vectors  $g$  and  $a$ , respectively.

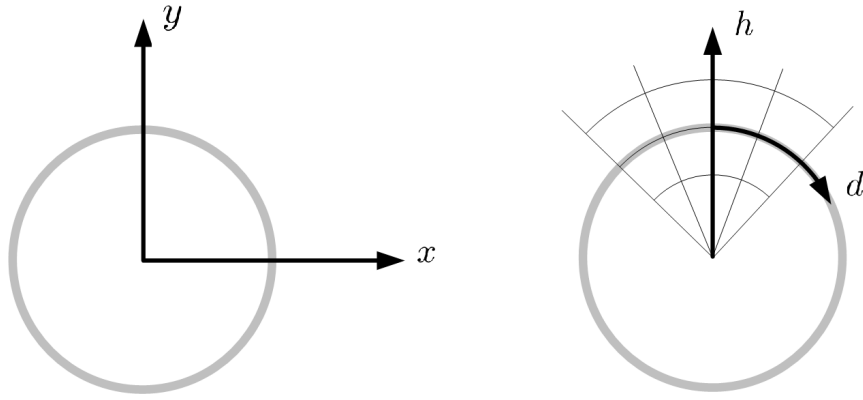
The RS computation considers a lunar lander in the terminal landing phase. Therefore, for simplification of the guidance law, the equations of motion is derived in downrange-altitude-crossrange ( $d-h-c$ ) frame. First of all, a simple 2-D model in  $d-h$  frame is derived. The model is further expanded to include the motions in 3D by following the same method. Finally, the 3D model will be simplified and the reasons and assumptions for the simplification is mentioned in the related section.

### 3.1 $d$ - $h$ Model

Two dimensional position of the spacecraft can be expressed as downrange  $d = d(t)$  and altitude  $h = h(t)$ . In  $x - y$  coordinate system, the position is defined as

$$x = (h + r) \sin \frac{d}{r} \quad (3.3)$$

$$y = (h + r) \cos \frac{d}{r} \quad (3.4)$$



**Figure 3.1:**  $x - y$  and  $d - h$  coordinate system.

Taking the time derivative of the both side yields of Eq. (3.4) and Eq. (3.3)

$$\dot{x} = \dot{h} \sin \frac{d}{r} + \frac{\dot{d}}{r} (h + r) \cos \frac{d}{r} \quad (3.5)$$

$$\dot{y} = \dot{h} \cos \frac{d}{r} - \frac{\dot{d}}{r} (h + r) \sin \frac{d}{r} \quad (3.6)$$

Taking again the time derivative of Eq. (3.5) and Eq. (3.6) gives

$$\ddot{x} = \ddot{h} \sin \frac{d}{r} + 2 \frac{\dot{d}}{r} \dot{h} \cos \frac{d}{r} + \frac{\ddot{d}}{r} (h + r) \cos \frac{d}{r} - \frac{\dot{d}^2}{r^2} (h + r) \sin \frac{d}{r} \quad (3.7)$$

$$\ddot{y} = \ddot{h} \cos \frac{d}{r} - 2 \frac{\dot{d}}{r} \dot{h} \sin \frac{d}{r} - \frac{\ddot{d}}{r} (h + r) \sin \frac{d}{r} - \frac{\dot{d}^2}{r^2} (h + r) \cos \frac{d}{r} \quad (3.8)$$

Define  $\gamma$  such that

$$\gamma = \arctan \frac{(h + r) \sin \frac{d}{r}}{(h + r) \cos \frac{d}{r}} = \frac{d}{r}.$$

The total forces acting on the spacecraft is expressed as

### 3. NONLINEAR DYNAMICAL MODEL

---

$$\ddot{x} = -g \sin \gamma - \frac{T}{m} \cos \delta \quad (3.9)$$

$$\ddot{y} = -g \cos \gamma - \frac{T}{m} \sin \delta \quad (3.10)$$

where  $\delta$  is the angle of the thrust vector perpendicular to the local horizon. Arranging terms for Eq. (3.5)-(3.6) and Eq. (3.7)-(3.8) leads to

$$\ddot{h} \sin \frac{d}{r} + 2\frac{\dot{d}}{r}\dot{h} \cos \frac{d}{r} + \frac{\ddot{d}}{r}(h+r) \cos \frac{d}{r} - \frac{\dot{d}^2}{r^2}(h+r) \sin \frac{d}{r} = -g \sin \frac{d}{r} - \frac{T}{m} \cos \delta \quad (3.11)$$

$$\ddot{h} \cos \frac{d}{r} - 2\frac{\dot{d}}{r}\dot{h} \sin \frac{d}{r} - \frac{\ddot{d}}{r}(h+r) \sin \frac{d}{r} - \frac{\dot{d}^2}{r^2}(h+r) \cos \frac{d}{r} = -g \cos \frac{d}{r} - \frac{T}{m} \sin \delta. \quad (3.12)$$

After simplification of Eq. (3.11) and Eq. (3.12)

$$\begin{aligned} \ddot{h} - \frac{\dot{d}^2}{r^2}(h+r) &= -g - \frac{T}{m} \left( \cos \delta \sin \frac{d}{r} + \sin \delta \cos \frac{d}{r} \right) \\ 2\frac{\dot{d}}{r}\dot{h} + \frac{\ddot{d}}{r}(h+r) &= -\frac{T}{m} \left( \cos \delta \cos \frac{d}{r} - \sin \delta \sin \frac{d}{r} \right). \end{aligned}$$

Finally rearranging the terms yields

$$\begin{aligned} \ddot{h} &= \frac{\dot{d}^2}{r^2}(h+r) - g - \frac{T}{m} \left( \cos \delta \sin \frac{d}{r} + \sin \delta \cos \frac{d}{r} \right) \\ \dot{d} &= -2\frac{\dot{d}}{r+h}\dot{h} - \frac{Tr}{m(h+r)} \left( \cos \delta \cos \frac{d}{r} - \sin \delta \sin \frac{d}{r} \right). \end{aligned}$$

### 3.2 $d$ - $h$ - $c$ Model

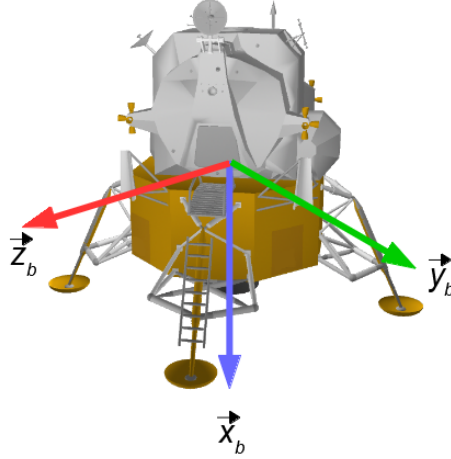
The equations of motions of the lunar lander is derived leading to the nonlinear dynamical model presented by Oehlschlagel (2011)*et al.* in reference<sup>69</sup>. The model includes three non-throtttable main engines. Only one main engine provides thrust during the final phase of landing. Reaction control system (RCS) thrusters are used for the rotation of the spacecraft. The forces of motion are derived in an inertial coordinate system and converted to the Moon Centered Moon Fixed coordinate system. The components of the thrust acting on the spacecraft in body-fixed coordinate system is illustrated in Figure 3.2 and can be expressed by

$$\mathbf{T}_{BF} = \frac{1}{m} \begin{pmatrix} -T_h - T_m \\ -T_s \\ -T_q \end{pmatrix} \quad (3.13)$$

After rotation

$$\mathbf{T}_{MCI} = \mathbf{R}_z(\beta - \delta)\mathbf{R}_y(\chi)\mathbf{T}_{BF} \quad (3.14)$$





**Figure 3.2:** Body-Fixed coordinate system.

$$\mathbf{T}_{\text{MCI}} = \begin{bmatrix} \cos(\beta - \delta) & -\sin(\beta - \delta) & 0 \\ \sin(\beta - \delta) & \cos(\beta - \delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \chi & 0 & \sin \chi \\ 0 & 1 & 0 \\ -\sin \chi & 0 & \cos \chi \end{bmatrix} \mathbf{T}_{\text{BF}} \quad (3.15)$$

The local horizon is represented with the vehicle carried coordinate frame (V) and the difference between the body-fixed coordinate system are shown in Figure 3.4 and Figure 3.5.

Gives

$$\mathbf{T}_{\text{MCI}} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} -\cos(\beta - \delta)((T_m + T_h) + T_q \sin \chi) + \sin(\beta - \delta)T_s \\ -\sin(\beta - \delta)((T_m + T_h) + T_q \sin \chi) - \cos(\beta - \delta)T_s \\ (T_m + T_h) \sin \chi - T_q \cos \chi \end{pmatrix}$$

where

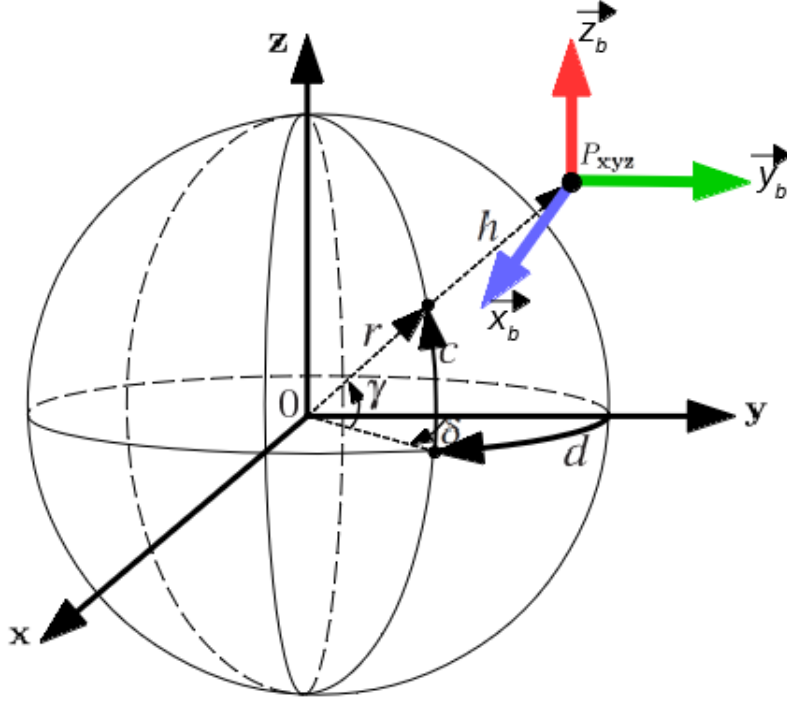
$$\begin{aligned} \delta &= \arctan \frac{x}{y} = \frac{d}{r} \\ \gamma &= \arctan \frac{z}{\sqrt{x^2 + y^2}} = \frac{c}{r}. \end{aligned} \quad (3.16)$$

After simplification of Eq. (3.2)

$$\mathbf{T}_{\text{MCI}} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} -\cos(\beta - \frac{d}{r})((T_m + T_h) + T_q \sin \chi) + \sin(\beta - \frac{d}{r})T_s \\ -\sin(\beta - \frac{d}{r})((T_m + T_h) + T_q \sin \chi) - \cos(\beta - \frac{d}{r})T_s \\ (T_m + T_h) \sin \chi - T_q \cos \chi \end{pmatrix}. \quad (3.17)$$

### 3. NONLINEAR DYNAMICAL MODEL

---



**Figure 3.3:**  $x - y - z$  and  $d - h - c$  Coordinate System.

Effect of gravity of Moon on the spacecraft in vehicle carried coordinate frame is

$$\mathbf{g}_V = \begin{pmatrix} 0 \\ -\frac{MG}{(r+h)^2} \\ 0 \end{pmatrix} \quad (3.18)$$

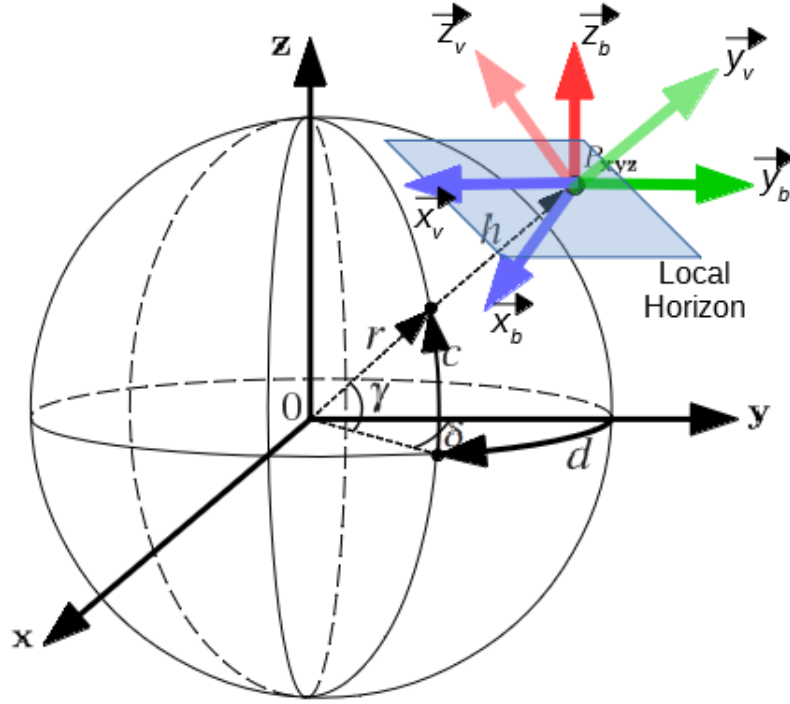
Rotation in respective coordinates gives

$$\mathbf{g}_{MCI} = \mathbf{R}_z(\delta)\mathbf{R}_x(-\gamma)\mathbf{g}_V \quad (3.19)$$

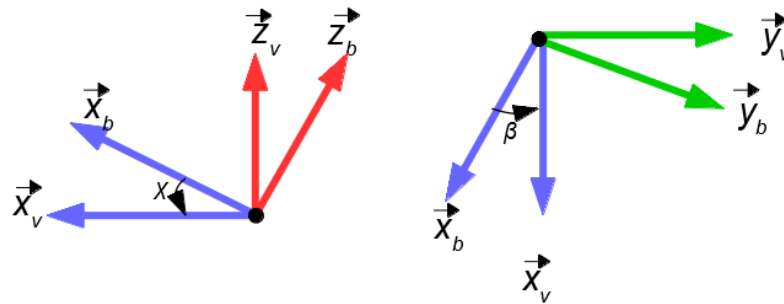
$$\mathbf{g}_{MCI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} \cos \delta & \sin \delta & 0 \\ -\sin \delta & \cos \delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{g}_V \quad (3.20)$$

Finally the forces acting on the system are added.

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \mathbf{g}_{MCI} + \mathbf{T}_{MCI} \quad (3.21)$$



**Figure 3.4:** Representation of Local Horizon and Body-Fixed Coordinate System.



**Figure 3.5:** Vehicle Carried Coordinate System and Body-Fixed Coordinate System.

The position of the spacecraft is expressed as

$$\begin{aligned}
 x &= (h + r) \cos \gamma \sin \delta &= (h + r) \cos \frac{c}{r} \sin \frac{d}{r} \\
 y &= (h + r) \cos \gamma \cos \delta &= (h + r) \cos \frac{c}{r} \cos \frac{d}{r} \\
 z &= (h + r) \sin \gamma &= (h + r) \sin \frac{c}{r}
 \end{aligned} \tag{3.22}$$

### 3. NONLINEAR DYNAMICAL MODEL

---

Taking time derivative of both sides in Eq. (3.22) yields

$$\begin{aligned}
 \dot{x} &= -\frac{(r+h)\sin\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{c}}{r} + \frac{(r+h)\cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{d}}{r} + \cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{h} \\
 \dot{y} &= -\frac{(r+h)\sin\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{c}}{r} - \frac{(r+h)\cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{d}}{r} + \cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{h} \quad (3.23) \\
 \dot{z} &= \frac{(r+h)\cos\left(\frac{c}{r}\right)\dot{c}}{r} + \sin\left(\frac{c}{r}\right)\dot{h}
 \end{aligned}$$

And the time derivative of Eq. (3.23) gives

$$\begin{aligned}
 \ddot{x} &= -\frac{(r+h)\cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{c}^2}{r^2} - \frac{2(r+h)\cos\left(\frac{d}{r}\right)\sin\left(\frac{c}{r}\right)\dot{c}\dot{d}}{r^2} \\
 &\quad - \frac{(r+h)\cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{d}^2}{r^2} - \frac{2\sin\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{c}\dot{h}}{r} \\
 &\quad + \frac{2\cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{d}\dot{h}}{r} - \frac{(r+h)\sin\left(\frac{d}{r}\right)\sin\left(\frac{c}{r}\right)\ddot{c}}{r} \\
 &\quad + \frac{(r+h)\cos\left(\frac{d}{r}\right)\cos\left(\frac{c}{r}\right)\ddot{d}}{r} + \cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\ddot{h} \\
 \ddot{y} &= -\frac{(r+h)\cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{c}^2}{r^2} + \frac{2(r+h)\sin\left(\frac{d}{r}\right)\sin\left(\frac{c}{r}\right)\dot{c}\dot{d}}{r^2} \\
 &\quad - \frac{(r+h)\cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\dot{d}^2}{r^2} - \frac{2\cos\left(\frac{d}{r}\right)\sin\left(\frac{c}{r}\right)\dot{c}\dot{h}}{r} \\
 &\quad - \frac{2\cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\dot{d}\dot{h}}{r} - \frac{(r+h)\cos\left(\frac{d}{r}\right)\sin\left(\frac{c}{r}\right)\ddot{c}}{r} \\
 &\quad - \frac{(r+h)\cos\left(\frac{c}{r}\right)\sin\left(\frac{d}{r}\right)\ddot{d}}{r} + \cos\left(\frac{c}{r}\right)\cos\left(\frac{d}{r}\right)\ddot{h} \\
 \ddot{z} &= -\frac{(r+h)\sin\left(\frac{c}{r}\right)\dot{c}^2}{r^2} + \frac{2\cos\left(\frac{c}{r}\right)\dot{c}\dot{h}}{r} \\
 &\quad + \frac{(r+h)\cos\left(\frac{c}{r}\right)\ddot{c}}{r} + \sin\left(\frac{c}{r}\right)\ddot{h}
 \end{aligned} \quad (3.24)$$

Finally recalling Eq.(3.21) and solving for  $\ddot{d}, \ddot{h}, \ddot{c}$  gives

$$\begin{pmatrix} \ddot{d} \\ \ddot{h} \\ \ddot{c} \end{pmatrix} = \begin{pmatrix} \frac{r}{m\tilde{h}\tilde{c}\tilde{c}}(-T_1c\tilde{d} + T_2s\tilde{d}) + 2\dot{d}\left(\frac{\dot{c}}{r}t\tilde{c} - \frac{\dot{h}}{\tilde{h}}\right) \\ \frac{1}{m}[(-T_1s\tilde{d} - T_2c\tilde{d})c\tilde{c} - T_3s\tilde{c}] + [(\dot{c}\tilde{c})^2 + \dot{c}^2]\frac{\tilde{h}}{r^2} - \frac{MG}{(\tilde{h})^2} \\ \frac{r}{m(\tilde{h})}[(T_1s\tilde{d} + T_2c\tilde{d})s\tilde{c} - T_3c\tilde{c}] - \frac{\dot{d}^2}{r}s\tilde{c}\tilde{c} - \frac{2\dot{c}\dot{h}}{\tilde{h}} \end{pmatrix} \quad (3.25)$$

$$\begin{aligned}
 \tilde{h} &= h + r, & s\tilde{d} &= \sin\frac{d}{r}, & c\tilde{d} &= \cos\frac{d}{r} \\
 s\tilde{c} &= \sin\frac{c}{r}, & c\tilde{c} &= \cos\frac{c}{r}, & t\tilde{c} &= \tan\frac{c}{r}
 \end{aligned}$$

**Table 3.1:** Definition of constants in Eq. (3.25) and Eq. (3.27.)

Constant	Definition	Value	Unit
$r$	radius of the Moon	1.737e6	m
$M$	mass of the Moon	7.3490e22	kg
$G$	gravitational constant	6.6743e-11	N·m <sup>2</sup> /kg <sup>2</sup>
$\sigma_m$	fuel consumption per unit thrust by the main engine	5e-4	-
$\sigma_{RCS}$	fuel consumption per unit thrust by the RCS	3.75e-4	-

Thrust vectors are defined in non-dimensional form. The first derivatives of the pitch ( $\beta$ ) and yaw ( $\chi$ ) angles are also added to the system of differential equations, where  $\omega_\beta$  and  $\omega_\chi$  are commanded angular rate

$$\dot{\beta} = \omega_\beta, \quad \dot{\chi} = \omega_\chi \quad (3.26)$$

Thrust dependent fuel consumption depending on  $\sigma_m$  and  $\sigma_{RCS}$  is defined as

$$\dot{m} = -|T_m| \cdot \sigma_m - (|T_u| + |T_s| + |T_q|) \cdot \sigma_{RCS} \quad (3.27)$$

where  $\sigma_m$  and  $\sigma_{RCS}$  define the coefficients for consumed propellant mass per unit generated thrust. Finally all states of the dynamical system defined by Eq. (3.25)-(3.27) are represented in the following state vector

$$x = (\dot{d}, \dot{h}, \dot{c}, d, h, c, \beta, \chi, m)^T \quad (3.28)$$

The control vector is represented by

$$u = (T_u, T_s, T_q, \omega_\beta, \omega_\chi)^T \quad (3.29)$$

The explanation and the values of the parameters used in the simulations are tabulated in Table 3.1.

As explained before the equations of motion are represented in Moon centered inertial frame. In fact, the assumption holds since the rotation of the Moon around its axis has negligible effect on the final position of the spacecraft. Acceleration of the spacecraft represented in MCI frame is expressed as

$$\mathbf{a}_{MCI} = \left( \frac{d^2 \mathbf{r}}{dt^2} \right)_{MCI} = \left( \frac{d\mathbf{v}}{dt} \right)_{MCI} = \left[ \left( \frac{d}{dt} \right)_r + \boldsymbol{\Omega} \times \right] \left[ \left( \frac{d\mathbf{r}}{dt} \right)_r + \boldsymbol{\Omega} \times \mathbf{r} \right] \quad (3.30)$$

where  $\boldsymbol{\Omega}$  is the angular velocity of the Moon around its axis. After carrying out the differentiations and rearranging the terms yields the acceleration in the rotating frame

$$\mathbf{a}_r = \mathbf{a}_{MCI} - 2\boldsymbol{\Omega} \times \mathbf{v}_r - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) - \frac{d\boldsymbol{\Omega}}{dt} \times \mathbf{r}. \quad (3.31)$$

### 3. NONLINEAR DYNAMICAL MODEL

---

The Moon completed one rotation around its axis in 27.32 days. So the angular velocity of the Moon is

$$\Omega = \frac{2\pi}{27.32 \cdot 86400} \approx 2.66 \cdot 10^{-6} \text{ rad/s.} \quad (3.32)$$

Neglecting the orbital perturbations of the Moon by other celestial bodies, the rotational velocity change of Moon is zero. Hence the last term in Eq. (3.30) drops. The term  $\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r})$  is also negligible considering the rotational velocity of the Moon. During the simulations, the largest component of the velocity  $\mathbf{v}$  is -19 m/s. Substituting the values in the term  $2 \boldsymbol{\Omega} \times \mathbf{v}_r$  and assuming a flight time of 50 s, the difference in the final position is approximately 0.12 m, which is inline with the assumption to express the equations of motion in a non-rotating frame.

### 3.3 Simplified $d$ - $h$ - $c$ Model

Recalling Eq. (3.33), the orthogonal thrusters  $T_h, T_s$  and  $T_q$  are modulated for attitude control. The availability of the technology and simplicity, the design could be altered such that, the thrust provided by the main engine is throttable. Hence the components of the thrust provided by RCS can be omitted with  $T_h = T_s = T_q = 0$ . This leads a simplified model with

$$\mathbf{T}_{\text{MCI}} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} -\cos(\beta - \frac{d}{r})(T_m) \\ -\sin(\beta - \frac{d}{r})(T_m) \\ T_m \sin \chi \end{pmatrix} \quad (3.33)$$

$$\begin{pmatrix} \ddot{d} \\ \ddot{h} \\ \ddot{c} \end{pmatrix} = \begin{pmatrix} \frac{r}{mhc\tilde{c}}(-T_1c\tilde{d} + T_2s\tilde{d}) + 2\dot{d}(\frac{\dot{c}}{r}t\tilde{c} - \frac{\dot{h}}{h}) \\ \frac{1}{m}[(-T_1s\tilde{d} - T_2c\tilde{d})c\tilde{c} - T_3s\tilde{c}] + [(d\dot{c}\tilde{c})^2 + \dot{c}^2]\frac{\tilde{h}}{r^2} - \frac{MG}{(h)^2} \\ \frac{r}{m(h)}[(T_1s\tilde{d} + T_2c\tilde{d})s\tilde{c} - T_3c\tilde{c}] - \frac{\dot{d}^2}{r}s\tilde{c}\tilde{c} - \frac{2\dot{c}\dot{h}}{h} \end{pmatrix} \quad (3.34)$$

Similarly,  $\omega_\beta$  and  $\omega_\chi$  are commanded angular rate

$$\dot{\beta} = \omega_\beta, \quad \dot{\chi} = \omega_\chi \quad (3.35)$$

Thrust dependent fuel consumption depending on  $\sigma_m$  and  $\sigma_{RCS}$  is defined as

$$\dot{m} = -|T_m| \cdot \sigma_m \quad (3.36)$$

Finally all states of the dynamical system defined by Eq. (3.34), Eq. (3.35), Eq. (3.36) are represented in the following state vector

$$x = (\dot{d}, \dot{h}, \dot{c}, d, h, c, \beta, \chi, m)^T \quad (3.37)$$

The control vector is represented by

$$u = (T_m, \omega_\beta, \omega_\chi)^T \quad (3.38)$$

In the subsequent sections, the reference RS is compared with RS computed with a modified Apollo guidance method. In the latter scenario, the nonlinear dynamical model is inverted to follow a given trajectory in  $x - y - z$  coordinates. However, the inversion cannot be performed for the nonlinear dynamical model with five control inputs. Therefore, the number of control inputs are decreased to three by omitting the additional thrust from the RCS.

### 3.4 Mission Scenario and Constraints

The reachable set for the lander is obtained for the terminal landing phase of the mission. The spacecraft starts a landing maneuver from its lunar circular orbit. After a Hohmann transfer, the lander moves on an elliptical descent orbit from an altitude of 100 km to an altitude of 15 km. Powered descent starts at an altitude of 15 km until 2 km. At this stage attitude adjustment maneuver is performed. The final phase of the landing maneuver is supposed to start at an altitude of 300 m as shown in Figure 3.6.

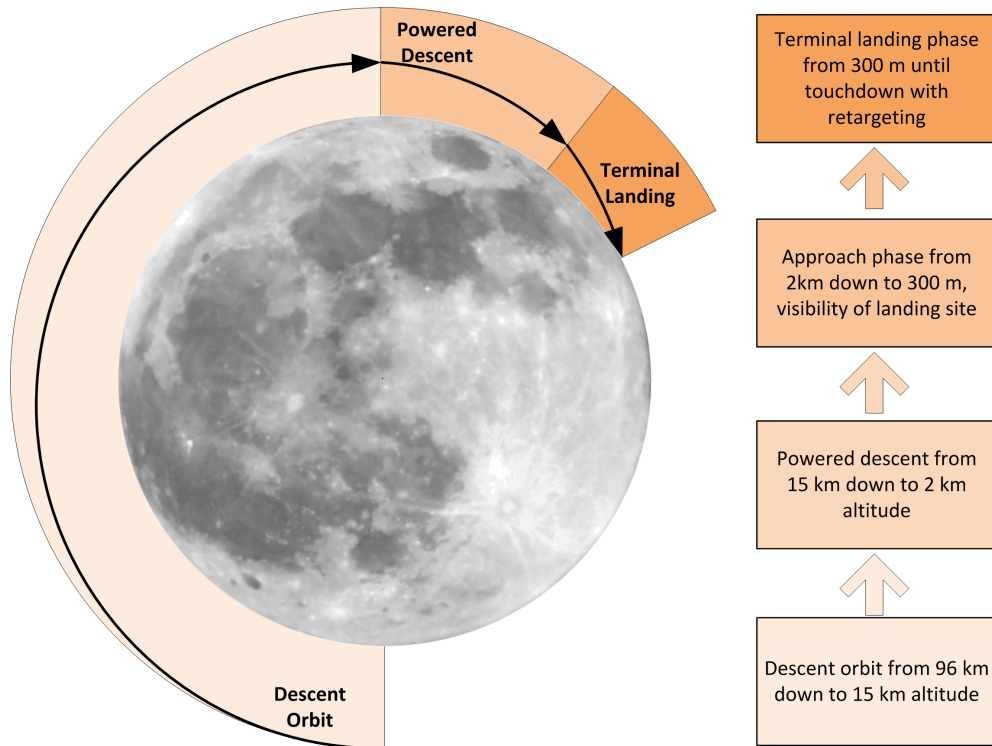


Figure 3.6: Mission Scenario for the Moon Landing. (not scaled)

Initial conditions are determined from state information at an altitude of 300 m using the optimal trajectory defined in<sup>69</sup>. The mission requires the following initial and terminal conditions

### 3. NONLINEAR DYNAMICAL MODEL

---

$$x_0 = (\dot{d}_0, \dot{h}_0, \dot{c}_0, \text{free}, h_0, \text{free}, \beta_0, \chi_0, m_0)^T \quad (3.39)$$

$$x_f = (0, 0, 0, \text{free}, 0, \text{free}, -\frac{\pi}{2}, \text{free}, m_f)^T \quad (3.40)$$

where  $(x_0 = x(t_0))$  and  $(x_f = x(t_f))$ . Additional constraints results from final landing condition. A landing is considered successful if the condition

$|\Delta x(t_f)| \leq \Delta x_{max}$  holds with

$$\Delta x_{max} = (1\text{m/s}, 1\text{m/s}, 1\text{ m/s}, d_f, 1\text{ m}, c_f, 10^\circ, 180^\circ, \text{free})^T \quad (3.41)$$

The initial conditions, final conditions and constraints are summarized in Table 3.2.

**Table 3.2:** State Information for Initial-Final Conditions and Allowed Bounds.

	$x_0$	$x_f$	$x_l$	$x_u$
$\dot{d}$ (m/s)	5	0	free	free
$\dot{h}$ (m/s)	-19	0	free	free
$\dot{c}$ (m/s)	0	0	free	free
$d$ (m)	0	free	free	free
$h$ (m)	300	0	0	free
$c$ (m)	0	free	free	free
$\beta$ ( $^\circ$ )	-86	-90	-90	90
$\chi$ ( $^\circ$ )	0	free	0	180
$m$	0.5397	$0.5 \leq$	0.5000	0.5397

where  $x_u$  and  $x_l$  denote upper and lower bounds on states respectively for the whole time interval. Mass of the vehicle  $m$  is nondimensionalized with respect to initial total mass, and final value must be larger than 0.5 of the initial total mass. Similarly  $u_0$  and  $u_f$  represents initial and final control inputs,  $u_u$  and  $u_l$  are associated upper and lower bounds on the control inputs which is summarized in the Table 3.3. Control inputs in final time ( $u_f$ ) can attain any values between the given interval. The upper and lower bounds on the rotation rates are defined considering the actual limits for the Lunar module of Apollo mission<sup>73</sup>. In addition, the state trajectories with active constraints, either on the states or controls, are not included in the reachable sets.

**Table 3.3:** Control Input Information with Initial-Final Values and Constraints.

	$u_0$	$u_f$	$u_l$	$u_u$
$T_m$	0	free	1/0	1/5
$T_u$	0	free	-0.222	0.222
$T_s$	0	free	-0.222	0.222
$T_q$	0	free	-0.222	0.222
$\omega_\beta$ ( $^\circ$ /s)	0	free	-2/ - 15	2/15
$\omega_\chi$ ( $^\circ$ /s)	0	free	-2/ - 15	2/15



## Part II

# Main Algorithm and Set Computation

## 4

# Discretization of Optimal Control Problem

In this Chapter, the OCP defined in Eq. (2.10) is discretized into finite dimensional NLP using Pseudo-Spectral Methods. In a PSM, the basis functions are typically the Chebyshev or Lagrange polynomials and the continuous functions are collocated at the collocation points which are obtained using linear combination of the roots of Legendre Polynomials or their derivatives. Gauss Pseudospectral Method (GPM), Lobatto Pseudospectral Method (LPM) and Radau Pseudospectral Method (RPM) with its flipped version (f-RPM) are the well known pseudospectral methods for discretization of continuous problem with respect to time domain<sup>74,75</sup>.

These methods are based on spectral methods and typically have exponential convergence rates compared to the traditional discretization methods. PSM were applied to optimal control problems in the late 1980s using Chebyshev polynomials. The classification of these methods lies in the selection of the generation of collocation points and basis functions used for the approximation of the continuous function. Legendre-Gauss-Lobatto (LGL) is a Legendre-based pseudospectral method using Lagrange polynomials. An extension of the Legendre-based PSM called Lobatto pseudospectral method (LPM) developed by Fahroo (2002)<sup>53</sup> *et al.* At the same time, another Legendre-based pseudospectral method called the Gauss pseudospectral method (GPM) was developed by Benson and Huntington (2007)<sup>76</sup> which uses Lagrange polynomials as basis functions and Legendre-Gauss (LG) points for collocation.

Despite the many advantages of direct methods, many of them do not give any information about the costate. The costate is important for verifying the optimality of the solution, sensitivity analysis, and real time optimization. In PSM, costate estimates were developed by relating the Karush-Kuhn-Tucker (KKT) conditions of the NLP to the continuous costate dynamics. A costate mapping principle has been derived to estimate the costate from the KKT multipliers<sup>52</sup>.

A novel variation of PSM called the Radau pseudospectral method (RPM) was used by Garg (2011)<sup>74</sup> using Legendre-Gauss-Radau (LGR) points. This variation has several advantages. First, the formulation of the method is not complicated and can be

## 4.1 Approximation of Functions using Legendre Polynomials

---

modified to incorporate any changes related to the constraints. Second, an accurate solution can be found without providing any initial guess for the costates for the sparse NLP solvers. Moreover, KKT multipliers of the NLP can be used to estimate the costate. And finally Radau pseudospectral method shows a fast convergence behaviour which is typical in PSM.

The OCP defined in 2.10 is discretized using SPARTAN (SHEFEX-3 Pseudospectral Algorithm for Reentry Trajectory ANalysis), an in-house developed discretization tool<sup>77 78 79</sup>.

### 4.1 Approximation of Functions using Legendre Polynomials

In mathematics, Legendre functions are solutions to Legendre's differential equation:

$$\frac{d}{dx} \left[ (1-x^2) \frac{d}{dx} P_n(x) \right] + n(n+1)P_n(x) = 0 \quad (4.1)$$

The Legendre differential equation may be solved using the standard power series method. The equation has regular singular points at  $x = 1$  so, in general, a series solution about the origin will only converge for  $|x| < 1$ . When  $n$  is an integer, the solution  $P_n(x)$  that is regular at  $x = 1$  is also regular at  $x = -1$ , and the series for this solution terminates (i.e. it is a polynomial).

These solutions for  $n = 0, 1, 2, \dots$  (with the normalization  $P_n(1) = 1$ ) form a polynomial sequence of orthogonal polynomials called the Legendre polynomials. Each Legendre polynomial  $P_n(x)$  is an  $n^{\text{th}}$  degree polynomial as shown in Figure 4.1. It may be expressed using Rodrigues' formula:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2)^n] \quad (4.2)$$

The first few Legendre polynomials are tabulated in Table 4.1.

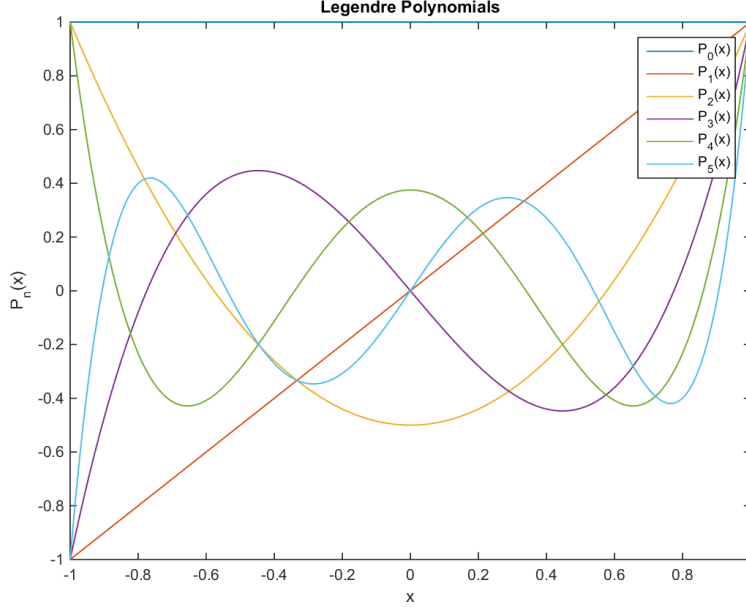
**Table 4.1:** Legendre Polynomials with Different Orders

$n$	$P_n(x)$
0	1
1	$x$
2	$\frac{1}{2}(3x^2 - 1)$
3	$\frac{1}{2}(5x^3 - 3x)$
4	$\frac{1}{8}(35x^4 - 30x^2 + 3)$
5	$\frac{1}{8}(63x^5 - 70x^3 + 15x)$

The collocation points are the roots of linear combination of Legendre polynomials. Legendre polynomials are orthogonal in the interval  $[-1, 1]$ . Therefore time vector  $t$  is

#### 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---



**Figure 4.1:** Legendre Polynomials with Different Orders

mapped to this domain by using following affine transformation

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0}. \quad (4.3)$$

The location of the roots for different PSM methods within the interval of  $[-1, 1]$  are illustrated in Figure 4.2.

In LGR method, Lagrange polynomials are used as basis functions for the approximation of continuous functions. With the Lagrange polynomial approximation, given a set of  $N$  discretization points of a continuous function of time,  $(t_1, \dots, t_N)$ ,  $f(t)$  on the interval  $t_i \in [t_0, t_f]$ , there exists a unique polynomial,  $F(t)$ , of degree  $N - 1$  such that

$$F(t_i) = f(t_i), \quad 1 \leq i \leq N. \quad (4.4)$$

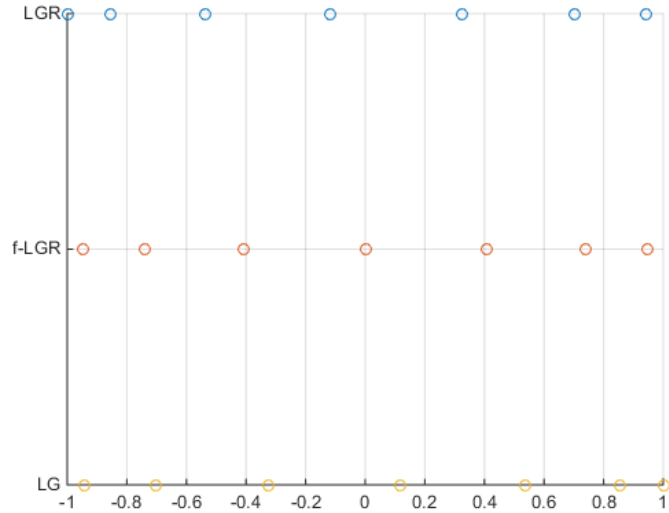
The unique polynomial approximation to the function,  $f(t)$ , is given by the Lagrange polynomial approximation formula:

$$F(t) = \sum_{i=1}^N f_i L_i(t) \quad (4.5)$$

where  $f_i = f(t_i)$ .  $L_i(t)$  are the Lagrange polynomials, defined as

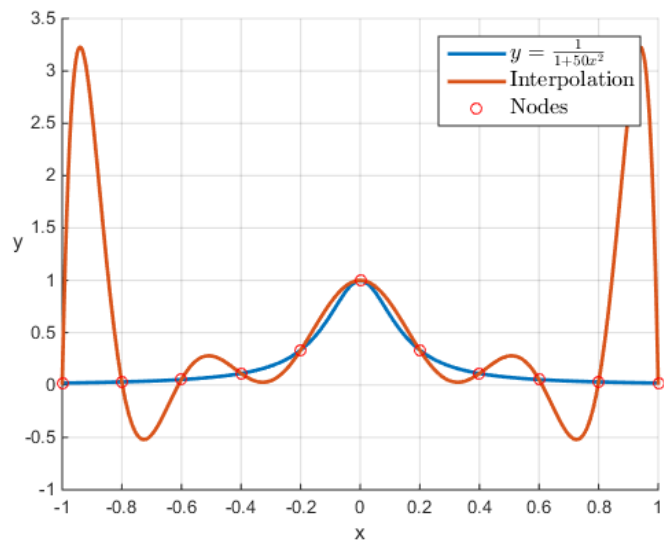
$$L_i(t) = \prod_{\substack{j=1 \\ j \neq i}}^N \left( \frac{t - t_j}{t_i - t_j} \right) \quad (4.6)$$

## 4.1 Approximation of Functions using Legendre Polynomials



**Figure 4.2:** Collocation Points obtained by Different PSM Methods.

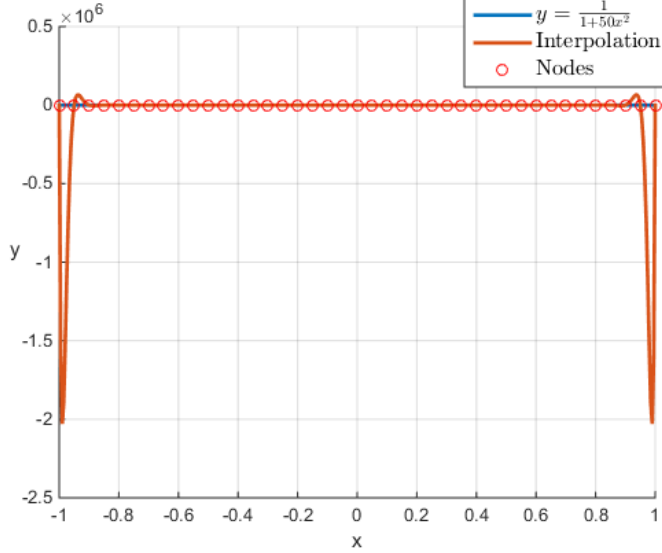
Selecting uniform points for the approximation of the function  $f(t)$  causes large oscillations at the boundaries of the time domain. This behaviour is called Runge's phenomenon. In Figure 4.3, the continuous function is approximated using  $N = 11$  uniform points. As the number of points is increased to  $N = 41$ , approximation error diverges due to Runge's phenomenon as shown in Figure 4.4.



**Figure 4.3:** Approximation of a Continuous Function using 11 Uniform Points with Lagrange Polynomials

#### 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---



**Figure 4.4:** Approximation of a Continuous Function using 41 Uniform Points with Lagrange Polynomials

In order to eliminate this problem, nonuniform collocation points are used. Let  $P_N(\tau)$  denote the  $N^{\text{th}}$  degree Legendre polynomial. The LGR discretization points are the roots obtained from  $P_{N-1}(\tau) + P_N(\tau)$  where

$$P_N(\tau) = \frac{1}{2^N N!} \frac{d^N}{d\tau^N} [(\tau^2 - 1)^N] \quad (4.7)$$

Figure 4.5 shows the approximation of the same function using nonuniform grid points with Lagrange polynomials as basis functions. As the number of collocation points is increased to  $N = 41$ , no oscillations exist at the boundaries of the time domain as illustrated in Figure 4.6.

Yet another advantage of PSM methods is their convergency behaviour. As the number of collocation points increases, the approximation error decreases exponentially. Figure 4.7 compares the approximation error for different number of nodes. As the number of collocation points is increased, the approximation error decreases exponentially for PSM.

In discretizing OCP into NLP, the dynamics of the system defined in (2.1) can be written in terms of algebraic nonlinear constraints evaluated at these discretization points

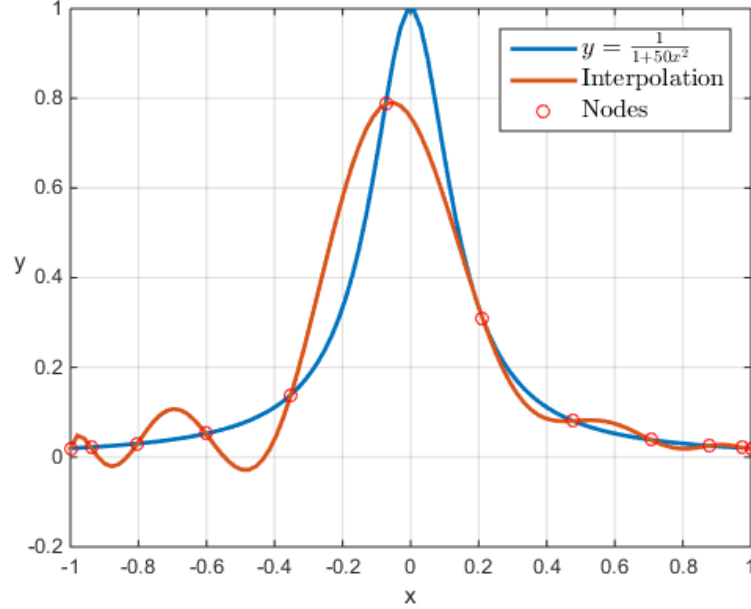
$$f_i = \sum_{j=1}^{N+1} D_{i,j} \cdot x_{d_j} - \frac{t_f - t_0}{2} f(\tau_i, x_{d_i}, u_{d_i}). \quad (4.8)$$

$$i = 1, \dots, N \quad j = 1, \dots, N + 1$$

where  $f_i$  is the residual at a given discrete node,  $D$  is the differential matrix with elements  $D_{i,j}$  which can be computed via using barycentric Lagrange interpolation

## 4.1 Approximation of Functions using Legendre Polynomials

---



**Figure 4.5:** Approximation of a Continuous Function using 11 Nonuniform Points with Lagrange Polynomials.

using the method in <sup>(80)</sup> or either

$$D = \text{reduce}(\hat{D}) \quad (4.9)$$

with

$$\hat{D}_{ij} = \frac{\sum_{\substack{k=0 \\ k \neq j}}^N \left[ \prod_{\substack{m=0 \\ m \neq j, k}}^N (\tau_i - \tau_m) \right]}{\prod_{\substack{k=0 \\ k \neq j}}^N (\tau_j - \tau_k)} \quad (4.10)$$

The operator *reduce* removes the first and last row of the associated matrix.

The continuous states and control inputs are approximated with polynomials which interpolate the values at the nodes. One well known property is that all information related to the states and the control inputs are equal their approximation at nodal values. In other words

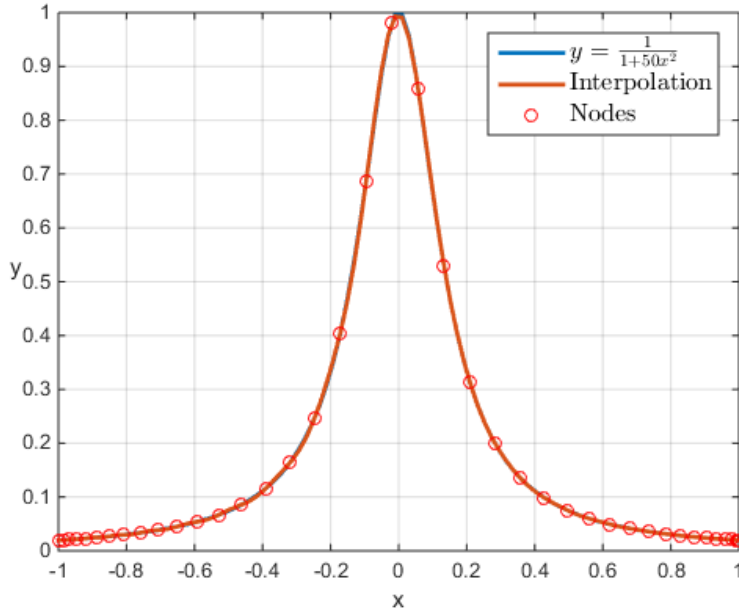
$$\begin{aligned} x(t_i) &= x_{d_i}, & x_{d_i} &\in \mathbb{R}^n, & i &= 1, \dots, N \\ u(t_i) &= u_{d_i}, & u_{d_i} &\in \mathbb{R}^m, & i &= 1, \dots, N. \end{aligned} \quad (4.11)$$

Similarly the objective function is discretized as

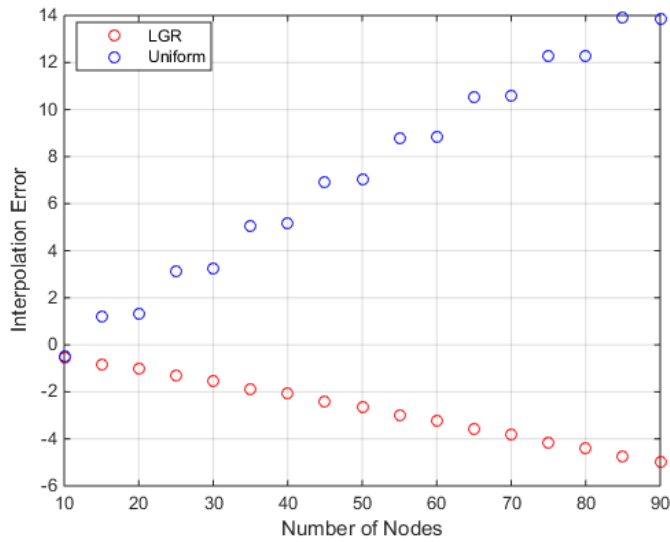
$$J = \phi(x(\tau_N), \tau_N). \quad (4.12)$$

#### 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---



**Figure 4.6:** Approximation of a Continuous Function using 41 Uniform Points with Lagrange Polynomials.



**Figure 4.7:** Discretization Error Comparison for PSM with Different Number of Nodes.

In this thesis, flipped LGR points are used for approximation of the continuous functions. The collocation points are obtained by flipping the LGR points about the origin, i.e., by taking the negative of LGR points. The flipped Radau pseudospectral



method differs from the standard Radau PSM because the final time is a collocation point in flipped Radau pseudospectral, whereas in standard Radau PSM, initial time is a collocation point.

As a result, in flipped Radau PSM, control at the final time is obtained. Initial control input is obtained by extrapolating the second discrete control input in the time domain. It is shown that the flipped Radau PSM results in a system of transformed KKT conditions that is equivalent to the discretized form of continuous first-order optimality conditions<sup>74</sup>. The subsequent section discusses the structure of the Jacobian of the NLP in detail.

## 4.2 Structure of Jacobian

For a given problem with  $N$  discretization points,  $n_s$  states,  $n_c$  controls and  $t_f$  as an optimization variable, the size of the Jacobian  $\mathcal{J}$  is

$$\dim(\mathcal{J}) = [N \cdot n_s + 1] \times [(N + 1) \cdot n_s + N \cdot n_c + 1] \quad (4.13)$$

The first row includes the derivatives of the objective function  $J$ . The latter rows are associated to the equality constraints or path constraints. The states are collocated using  $N + 1$  nodes. Different from states, the control inputs are collocated using  $N$  nodes. It should be noted that since f-RPM is used for discretization, initial control input  $u_0$  does not appear in the  $\mathcal{J}$ . Instead, it is extrapolated once the problem is solved. The size of the  $\mathcal{J}$  is different for the set computations where  $t_f$  is fixed and taken as an optimization variable. The structure of the resulting Jacobian is

$$\mathcal{J} = \begin{bmatrix} \frac{\partial J}{\partial x_0} & \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial u_1} & \cdots & \frac{\partial J}{\partial x_N} & \frac{\partial J}{\partial u_N} & \frac{\partial J}{\partial t_f} \\ \frac{\partial f_{c1}}{\partial x_0} & \frac{\partial f_{c1}}{\partial x_1} & \frac{\partial f_{c1}}{\partial u_1} & \cdots & \frac{\partial f_{c1}}{\partial x_N} & \frac{\partial f_{c1}}{\partial u_N} & \frac{\partial f_{c1}}{\partial t_f} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial f_{cN}}{\partial x_0} & \frac{\partial f_{cN}}{\partial x_1} & \frac{\partial f_{cN}}{\partial u_1} & \cdots & \frac{\partial f_{cN}}{\partial x_N} & \frac{\partial f_{cN}}{\partial u_N} & \frac{\partial f_{cN}}{\partial t_f} \\ \frac{\partial g_1}{\partial x_0} & \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial x_N} & \frac{\partial g_1}{\partial u_N} & \frac{\partial g_1}{\partial t_f} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial g_N}{\partial x_0} & \frac{\partial g_N}{\partial x_1} & \frac{\partial g_N}{\partial u_1} & \cdots & \frac{\partial g_N}{\partial x_N} & \frac{\partial g_N}{\partial u_N} & \frac{\partial g_N}{\partial t_f} \end{bmatrix} \quad (4.14)$$

The  $\mathcal{J}$  of the associated NLP is expressed as sum of three different contributors<sup>81</sup>.

$$\mathcal{J} = \mathcal{J}_{ps} + \mathcal{J}_{n_s} + \mathcal{J}_{th} \quad (4.15)$$

$\mathcal{J}_{ps}$  is related to the use of f-RPM. It is the contribution to the  $\mathcal{J}$  and to the constraints represented given by the use of discrete differential matrix  $D$ .

$$\mathcal{J}_{ps} = \begin{bmatrix} \tilde{D}_{1,1} & \cdots & O_{1 \times [(N+1) \cdot n_s + N \cdot n_c + 1]} & \cdots & \tilde{D}_{1,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \tilde{D}_{N,1} & \cdots & \cdots & \cdots & \tilde{D}_{N,N} \end{bmatrix} O_{[N \cdot n_s + 1] \times 1} \quad (4.16)$$

## 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---

where

$$\tilde{D} = D_{i,j} \cdot I_{n_s}, \quad i = 1, \dots, N, j = 1, \dots, N \quad (4.17)$$

$I_{n_s}$  is the identity matrix with dimension  $n_s$ .

Second contributor, numerical Jacobian  $\mathcal{J}_n$ , can be computed using finite difference method, complex step differentiation<sup>82</sup> or dual number theory. In SPARTAN all this methods can be selected for the computation of the derivatives. The results of the accuracy and computational requirements was demonstrated on benchmark problems by D'Onofrio (2016)<sup>11</sup> *et al.*

### Derivatives

The numerical derivatives can be computed using Finite Difference, Complex Step Differentiation, Dual Numbers.

#### Finite Difference

These derivatives are obtained using finite difference method by using the Taylor series expansion of the objective function.

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \mathcal{O}(\Delta x^2). \quad (4.18)$$

Solving for  $f'(x)$  gives

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x). \quad (4.19)$$

The perturbation value  $\Delta x \ll 1$  is problem dependent and chosen by tuning. Obtaining the derivatives of the objective function is the most time consuming phase during solution of the associated NLP.

#### Complex Step

The derivation of the complex-step derivative approximation is accomplished by approximating a nonlinear function with a complex variable using a Taylor's series expansion.

$$f(x + ih) = f(x) + ihf'(x) - h^2 \frac{f''(x)}{2!} - ih^3 \frac{f'''(x)}{3!} + \dots \quad (4.20)$$

Taking only the imaginary parts of both sides gives

$$\text{Im}[f(x + ih)] = hf'(x) - h^3 \frac{f'''(x)}{3!} \quad (4.21)$$

Dividing by  $h$  and rearranging yields

$$f'(x) = \frac{\text{Im}[f(x + ih)]}{h} + \mathcal{O}(h^2) \quad (4.22)$$

The price for the accuracy is the computational complexity since performing complex operations are more difficult compared to operations with real numbers. In order to overcome this problem, each element of  $\mathcal{J}_n$  is checked if there is dependency between discretized functions and variables. Another advantage is that, the derivatives are not sensitive to the chosen perturbation amount. Thus, there is no need for tuning of the perturbation amount during the computation of the derivatives.

### Dual Numbers

In linear algebra, the dual numbers extend the real numbers by introducing additional term  $\epsilon$  with the property  $\epsilon^2 = 0$  with the form

$$z = a + b\epsilon \tag{4.23}$$

where  $a$  and  $b$  are real numbers and, in particular

$$\begin{aligned} a &= \mathit{real}(z) && \text{Real Part} \\ b &= \mathit{dual}(z) && \text{Dual Part} \end{aligned}$$

The main idea relies on the fact that  $\epsilon^2 = 0$  with  $\epsilon \neq 0$ . This condition is not correct mathematically while  $\epsilon^2 = 0$  requires  $\epsilon = 0$ . Therefore these numbers were also called "fake numbers" since this class of number may not exist. In matrix notation the dual numbers can be represented as

$$\epsilon = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; \quad z = a + b\epsilon = \begin{pmatrix} a & b \\ 0 & a \end{pmatrix} \tag{4.24}$$

It is easy to see that the matrix form satisfies all the properties of the dual numbers. In order to implement the dual numbers, algebraic operations on these numbers should be properly defined. The dual numbers have been implemented in MATLAB as a new class of numbers, using operator overloading.

The dual-step differentiation method uses the dual numbers to provide first order derivatives up to the machine precision. Consider the Taylor series of a function  $f(x)$  for  $x \in \mathbb{R}$  for a given perturbation value  $a$ .

$$f(x + a) = f(x) + af'(x) + \frac{1}{2!}a^2f'' + \frac{a^3f'''(x)}{3!} \tag{4.25}$$

For  $a$  the dual number

$$a = a_1\epsilon \quad \text{with} \quad \epsilon^2 = 0 \quad \text{and} \quad \epsilon \neq 0 \tag{4.26}$$

so that  $a^2 = 0, a^3 = 0, \dots$ , the Taylor series truncates exactly at the first-derivative term yielding

$$f(x + a) = f(x) + a_1f'(x)\epsilon \tag{4.27}$$

#### 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---

For  $f'(x)$  taking the related term with  $\epsilon$  component and divide by  $a_1$ , yields the dual-step first derivative formula

$$f'(x) = \frac{\text{Dual}[f(x+a)]}{a_1} \quad (4.28)$$

Since the dual-step derivative approximation does not involve a difference operation and no terms of the Taylor series are ignored, this formula is subject neither to truncation error, nor to round-off error. There is no need to make the step size small and the simplest choice is  $a_1 = 1$ , which eliminates the need to divide by the step size. Therefore, using the dual-step method, the error between numerical and analytical derivative ( $\eta = |f' - f'_{ref}|/|f'_{ref}|$ ) is machine zero regardless of the selected step size. The disadvantage of the use of dual numbers is that, discontinuous data cannot be used for the derivatives such as look-up tables.

For all the available methods, the price for the accuracy is the computational complexity since performing complex or dual operations are more difficult compared to operations with real numbers. In order to overcome this problem, each element of  $J_n$  is checked if there is dependency between discretized functions and variables. Dependency matrix  $M_d$  is defined as in Table 4.2.

**Table 4.2:** Dependency Matrix  $M_d$

	variable 1	variable 2
function	0 (no dependency)	1 (dependent)

Simplified  $\mathcal{J}_{n_s}$  is evaluated by:

$$\mathcal{J}_{n_s} = \mathcal{J}_n \circ M_d \quad (4.29)$$

where  $\circ$  represents the Hadamard product.

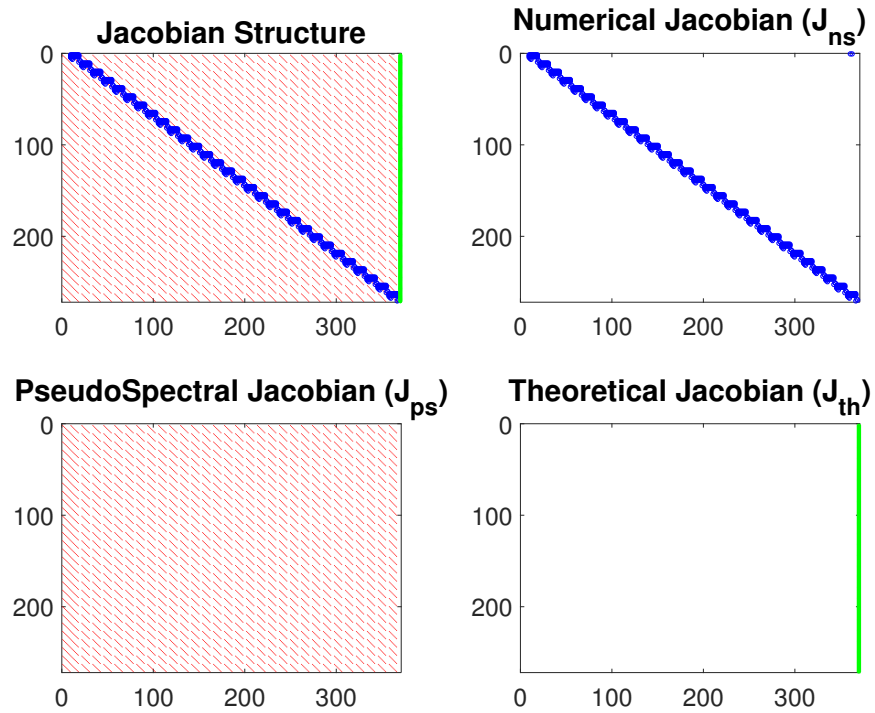
$$\mathcal{J}_n = -\frac{t_f - t_0}{2} \begin{bmatrix} \frac{\partial J}{\partial x_0} & \frac{\partial J}{\partial x_1} & \frac{\partial J}{\partial u_1} & \cdots & \frac{\partial J}{\partial x_N} & \frac{\partial J}{\partial u_N} \\ \frac{\partial f_{c1}}{\partial x_0} & \frac{\partial f_{c1}}{\partial x_1} & \frac{\partial f_{c1}}{\partial u_1} & \cdots & \frac{\partial f_{c1}}{\partial x_N} & \frac{\partial f_{c1}}{\partial u_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_{cN}}{\partial x_0} & \frac{\partial f_{cN}}{\partial x_1} & \frac{\partial f_{cN}}{\partial u_1} & \cdots & \frac{\partial f_{cN}}{\partial x_N} & \frac{\partial f_{cN}}{\partial u_N} \\ \frac{\partial g_1}{\partial x_0} & \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial x_N} & \frac{\partial g_1}{\partial u_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial g_N}{\partial x_0} & \frac{\partial g_N}{\partial x_1} & \frac{\partial g_N}{\partial u_1} & \cdots & \frac{\partial g_N}{\partial x_N} & \frac{\partial g_N}{\partial u_N} \end{bmatrix} O_{[N \cdot n_s + 1] \times 1} \quad (4.30)$$

Thus, the number of elements needed to be computed for  $\mathcal{J}_{n_s}$  using complex step differentiation decreases substantially.

Final contribution comes from the final time  $t_f$ . If the final time is also an optimization variable,  $\mathcal{J}_t$  is computed as follows

$$\mathcal{J}_t = -\frac{1}{2} \begin{bmatrix} 0 \\ f_{c,1} \\ f_{c,2} \\ \dots \\ f_{c,N} \end{bmatrix} \quad (4.31)$$

If the final time  $t_f$  is fixed,  $\mathcal{J}_t = 0$ . Figure 4.8 represents the structure of the  $\mathcal{J}$  and its contributors. Finally resulting sparse NLP is solved either by using SQP method or Interior Point Method. The solvers approximates the Hessian matrix numerically using BFGS method<sup>83</sup>.



**Figure 4.8:** Resulting Structure of the Jacobian

The solution of the NLP gives the values of states  $x_i$  and control inputs  $u_i$  ( $i = 1, \dots, N$ ) at the collocation points. The values are post-processed by transforming all the values into time domain  $[t_0, t_f]$ . In addition, the state trajectories are computed by a suitable RK propagation method by integrating the control input  $u(t)$ . SPARTAN computes the three contributors to the Jacobian of the NLP separately and increases the computational efficiency of the discretization<sup>81</sup>. Linear and nonlinear automatic

#### 4. DISCRETIZATION OF OPTIMAL CONTROL PROBLEM

---

scaling techniques are also applied for NLPs with states and constraints with several orders of magnitude<sup>78</sup>. In addition, several discretization methods are used during the computation of the numerical derivatives. In the solution of the associated NLPs for the computation of reference RS, dual numbers are used despite the computational complexity.

# 5

## Set Computations

In the previous chapter, the main algorithm for computing the RS was introduced. As discussed previously, the method relies on finding a feasible trajectory between the initial states and the desired target final state. The state space is discretized using uniform grid points. In this chapter, grid points are projected onto the plane defined by downrange and crossrange on the Moon surface to find the maximum attainable landing area of the Lunar lander. In this way a subset of the RS is approximated. This eliminates the need to compute the RS in all dimensions of the state space, while the information about some states might not include crucial information. Moreover, extra effort to project the whole RS onto a lower dimensional subset of state space is avoided.

In this chapter RS is computed for different scenarios with propellant-to-reach and time-to-reach maps. First, the developed method is tested on two example problems from the literature. Later on, a reference RS is computed with fixed final time considering equality and inequality constraints mentioned before. The effect of bounds for the angular rate and thrust on RS is demonstrated. In the final scenario, the simplified dynamical model is used for the computation of the RS.

In order to verify the solution of NLP, the control inputs obtained from the solution are propagated. The spectral convergence of the PSM methods is tabulated with the required computation time.

### 5.1 Solution of an Optimal Trajectory

The RS is approximated by the union of final states obtained from solution of a set of problems. Therefore, obtaining a feasible trajectory without violating the constraints is crucial. Recall the OCP in Eq. (2.10) together with the nonlinear dynamical system with state vector in Eq. (3.37) and control vector in Eq. (3.38). Initial and final conditions on the states are defined in Table 5.1 and controls in Table 5.2.

## 5. SET COMPUTATIONS

---

**Table 5.1:** State Information for the example OCP

	$x_0$	$x_f$	$x_l$	$x_u$
$\dot{d}$ (m/s)	5	0	free	free
$\dot{h}$ (m/s)	-19	0	free	free
$\dot{c}$ (m/s)	0	0	free	free
$d$ (m)	0	0	free	free
$h$ (m)	300	0	0	free
$c$ (m)	0	0	free	free
$\beta$ ( $^\circ$ )	-86.02	-90	-90	90
$\chi$ ( $^\circ$ )	0	free	0	180
$m$	0.5397	$\geq 0.5$	0.5000	0.5397

**Table 5.2:** Control Input Information for the example OCP

	$u_0$	$u_f$	$u_l$	$u_u$
$T_m$	0	free	0	4
$\omega_\beta$ ( $^\circ$ /s)	0	free	-15	15
$\omega_\chi$ ( $^\circ$ /s)	0	free	-15	15

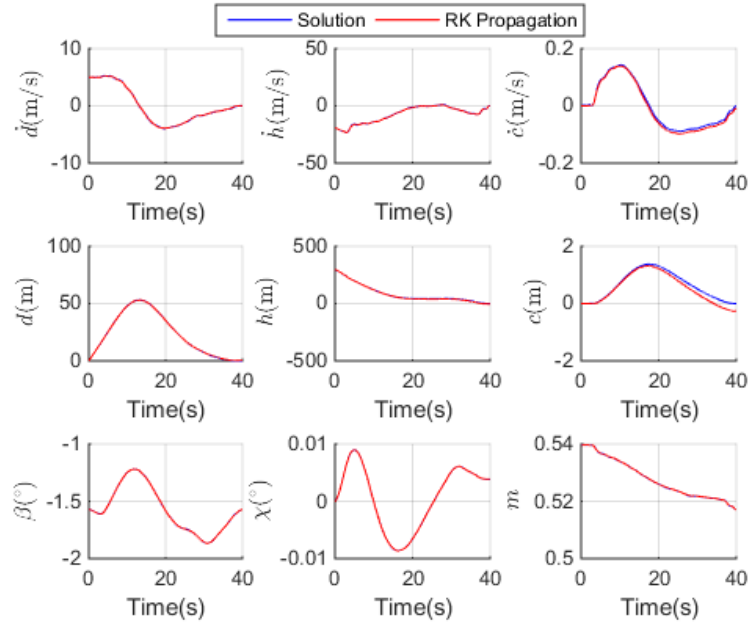
where  $x_0$  define the initial states at  $t = t_0$ ,  $x_f$  define the final states at  $t = t_f$ ,  $x_l$  and  $x_u$  define the upper and lower bounds on the states. Similarly  $u_0$  define the initial control input at  $t = t_0$ ,  $u_f$  define the final control input at  $t = t_f$ ,  $u_l$  and  $u_u$  define the upper and lower bounds on the control inputs. The values are taken from Arslantas (2016) *et.al*<sup>7</sup>. After discretizing the OCP in Eq. (2.10) with PSM and solving the resulting NLP for a fixed final time ( $t_f = 40$  s), state trajectories are shown in Figure 5.1 and control input in Figure 5.2 respectively. The results show that the spacecraft performs the landing maneuver without attaining sub-surface levels in altitude, satisfying the safe landing condition without any violations in constraints.

In order to compute the RS, the same problem is solved with a different objective function targeting a specific landing point on the Moon surface. There is a need for error metrics to assess the quality of the solution. After solving the NLP of the associated OCP in Eq. (2.10), state information and control inputs for each grid point  $g_s \in G_s$  are obtained. Results are validated by propagating the control inputs by 4<sup>th</sup> order variable-step-sized Runge-Kutta integration and comparing them with state trajectories from the solution of the NLP. The comparison is done by measuring the Euclidean distance between the final downrange and crossrange obtained by these two different approaches. In other words, it makes a comparison between the solution of NLP and the numerical integration scheme. Figure 5.1 shows the state trajectories obtained by propagating the control inputs.

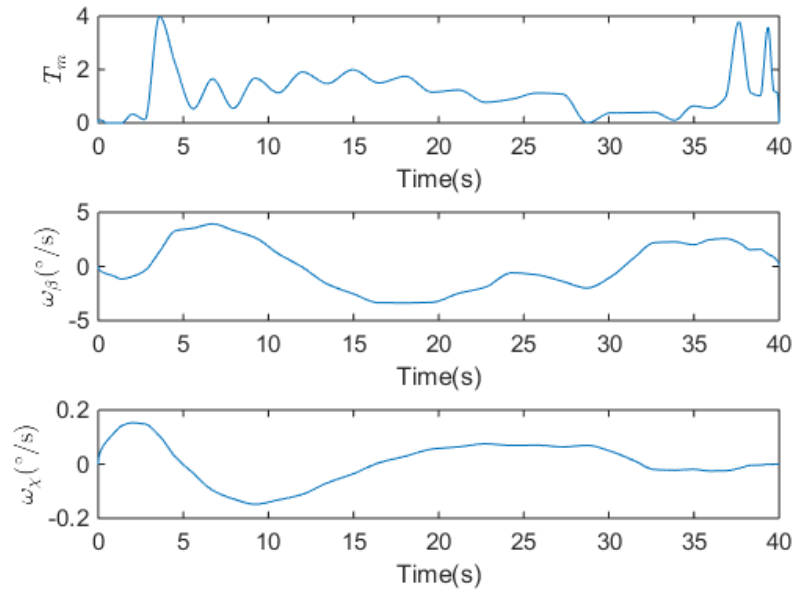
One advantage of PSM is that for smooth problems if the number of discretization points increases, the solution of the NLP and the states obtained by propagating the control inputs tend to converge. However, due to the increased size of the problem, the



## 5.1 Solution of an Optimal Trajectory



**Figure 5.1:** State Trajectories after solving Eq. (2.10). The control inputs are propagated and state trajectories are compared.



**Figure 5.2:** Control Inputs obtained by solving Eq. (2.10).

## 5. SET COMPUTATIONS

---

computational time increases accordingly.

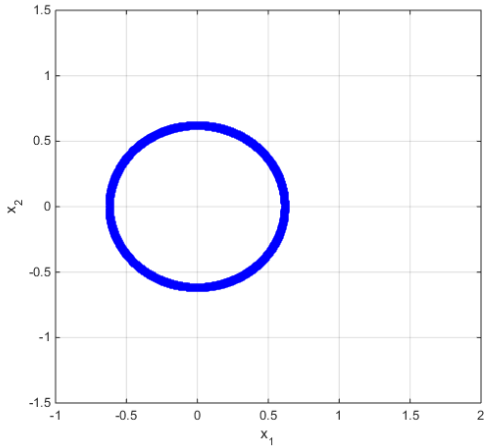
### 5.2 Reachable Sets for Sample Problems

The following examples are considered for toy problems<sup>30</sup>. The true RS of the examples are either known analytically or reference solutions are obtained by exhaustive search. In each example the state space is discretized and NLP is solved for every grid point.

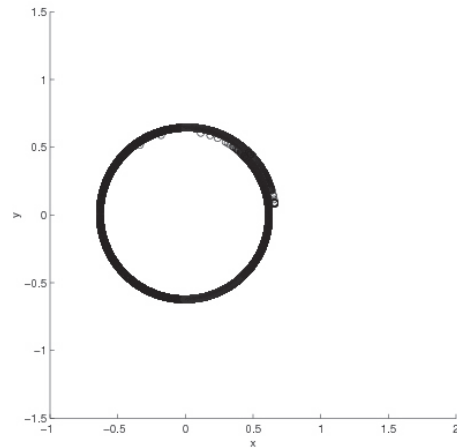
The first example is constructed such a way that the RS is a circle. The dynamical equations, initial values and equality constraints for the problem is as follows

$$\begin{aligned}\dot{x} &= 8(a_1x(t) + a_2y(t) - 2a_2y(t)u(t)) \\ \dot{y} &= 8(-a_2x(t) + a_1y(t) + 2a_1x(t)u(t)) \\ x(0) &= y(0) = 2 \\ u(t) &\in [-1, 1], t \in [0, 1]\end{aligned}$$

where  $a_1 = \sigma^2 - 1$ ,  $a_2 = \sigma\sqrt{1 - \sigma^2}$ , and  $\sigma = 0.9$ .



(a) RS Solution by the Developed Main Algorithm.



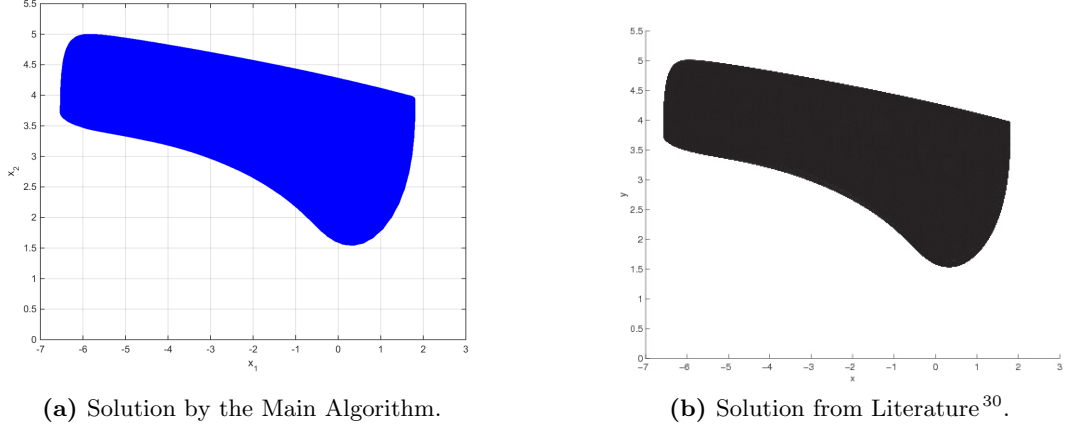
(b) RS Solution from Literature<sup>30</sup>.

**Figure 5.3:** Comparison of RS for the Kenderov Problem.

Figure 5.3a shows the results obtained by the main RS computation algorithm. The result is inline with the reference set illustrated in Figure 5.3b. The second example is an electric circuit and is known as Rayleigh problem. The nonlinear control problem is

$$\begin{aligned}\dot{x} &= y(t) \\ \dot{y} &= -x(t) + y(t)(1.4 - 0.14y(t)^2) + 4u(t) \\ x(0) &= y(0) = -5 \\ u(t) &\in [-1, 1], t \in [0, 2.5].\end{aligned}$$

### 5.3 Reachable Sets for Nominal Dynamical Model



**Figure 5.4:** Comparison of RS for the Rayleigh Problems.

As shown in Figures 5.4a-5.4b, the results obtained by the proposed method are similar with respect to the example problems. Since the method described in the literature has a different strategy for gridding the state space, a comparison in terms of computation time is not included. The proposed algorithm is able to approximate non-convex RS with equality constraints and inequality constraints on the states and control inputs. In addition, final time can also be included as a free parameter to the definition of the OCP. This allows computation of time-to-reach maps for a given dynamical nonlinear system.

### 5.3 Reachable Sets for Nominal Dynamical Model

In this section, the main algorithm is used to compute RS for different scenarios. The candidate landing area is discretized and mapped onto crossrange and downrange coordinates. The RS is computed for fixed and free final time. In case of free final time, the distance between uniform grid points is 250 m and for fixed final time ( $t_f = 40$  s) the distance is 50 m. The number of nodes for the discretization of the OCP in Eq. (2.10) is 50. For the case study, the number of elements of the Jacobian matrix is  $451 \times 709$  and  $451 \times 710$  for the problems with fixed final time and free final time respectively. The coordinates of the discretized region on the surface of the Moon is obtained by trial cases with a coarse grid and the size of grid is minimized considering the dimensions of the RS.

In all scenarios, the constraints defined in Table 3.2 and Table 3.3 are valid. In some scenarios, the effect of the bounds on the control inputs are illustrated. In these cases, the bounds are defined explicitly. For the RS computation, the solutions are validated with RK propagation. In addition, safe landing condition is checked for the solutions that constitute the approximated RS. After the results are obtained, propellant-to-reach or time-to-reach maps are generated.

Figure 5.5 shows the RS for  $t_f = 40$  s and Figure 5.6 shows the maximum attainable

## 5. SET COMPUTATIONS

---

landing area with free final time. As predicted, the landing area with free final time is larger compared to the scenario when  $t_f$  is fixed. Due to the initial velocity in downrange, the reachable set is not symmetric with respect to crossrange axis. It is also clear that, propellant required to reach points near the boundary of the RS is higher than the points in the interior of the RS.

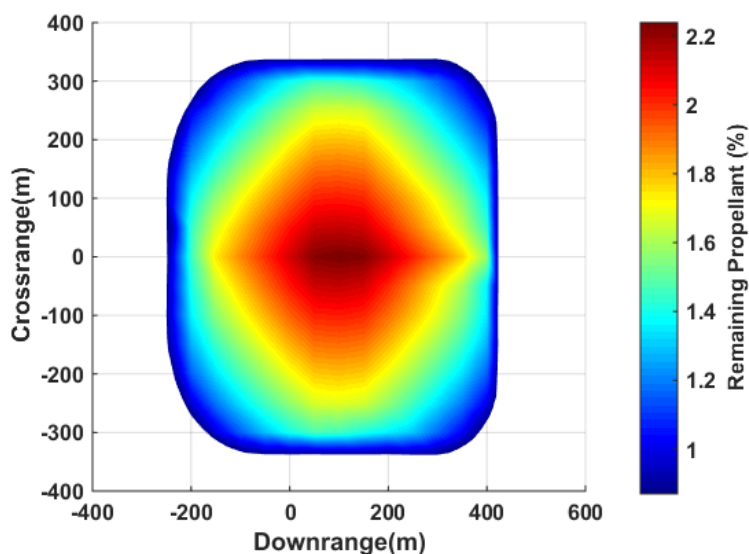


Figure 5.5: Reachable Set with Fuel Cost -  $t_f = 40$  (s).

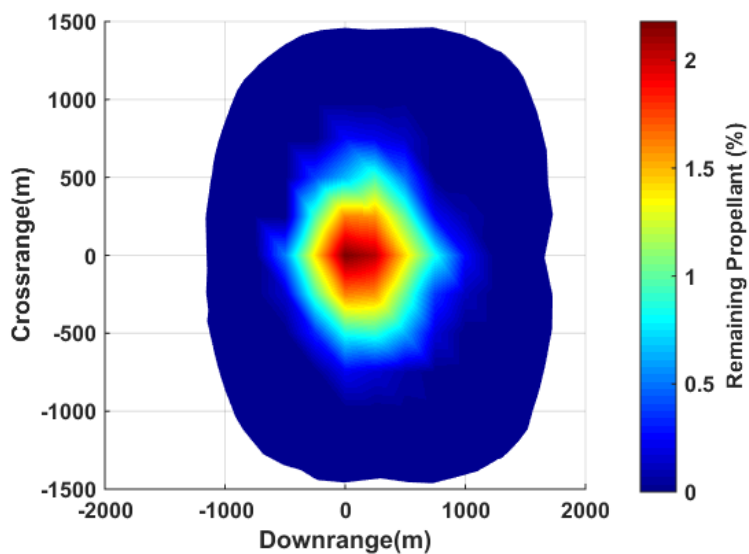
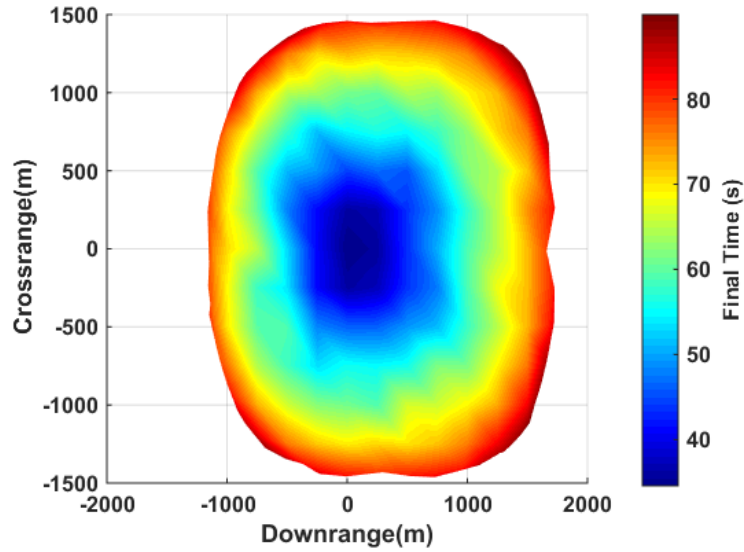


Figure 5.6: Reachable Set with Fuel Cost -  $t_f = \text{free}$ .

### 5.3 Reachable Sets for Nominal Dynamical Model

For the results, solutions for the family of NLPs are obtained with specific objective function given in Eq. (2.10), and all the following statements are related to the specific set of problems. Figure 5.7 shows the time required to reach to the points inside the RS. Same comments are valid for the final time as in the case of final propellant. It takes more time to achieve a successful landing on the points near the boundary of the RS than on the points interior of the set. In this case, the size of the RS is determined with the amount of final propellant. For the points on the border of the RS with free final time, all the propellant is consumed.



**Figure 5.7:** Reachable Set with Time Cost.

For the first scenario, maximum and minimum values for downrange and crossrange respectively are  $[-249 \text{ m}, 423 \text{ m}]$  and  $[-338 \text{ m}, 338 \text{ m}]$ . The extreme points for the scenario with free final time are  $[-1161 \text{ m}, 1729 \text{ m}]$  and  $[-1462 \text{ m}, 1462 \text{ m}]$  for downrange and crossrange. As there is no initial velocity in the crossrange direction, the RS is symmetric with respect to this axis.

Figure 5.8 shows the evolution of the states and the landing tunnel. The landing tunnel shows the evolution of the crossrange and downrange with propellant cost for the lunar lander to land safely on any points inside the computed RS.

#### RS with different initial states

The algorithm is used to compute RS with different initial velocities. Figure 5.9 shows the RS with  $\dot{d}_0 = \dot{c}_0 = 0 \text{ (m/s)}$  for fixed final time, while initial values of other states are not changed. In this case, RS exhibits a symmetric behavior in downrange and crossrange coordinates. The maximum and minimum values for downrange and crossrange respectively are  $[-341 \text{ m}, 334 \text{ m}]$  and  $[-338 \text{ m}, 338 \text{ m}]$ .

## 5. SET COMPUTATIONS

---

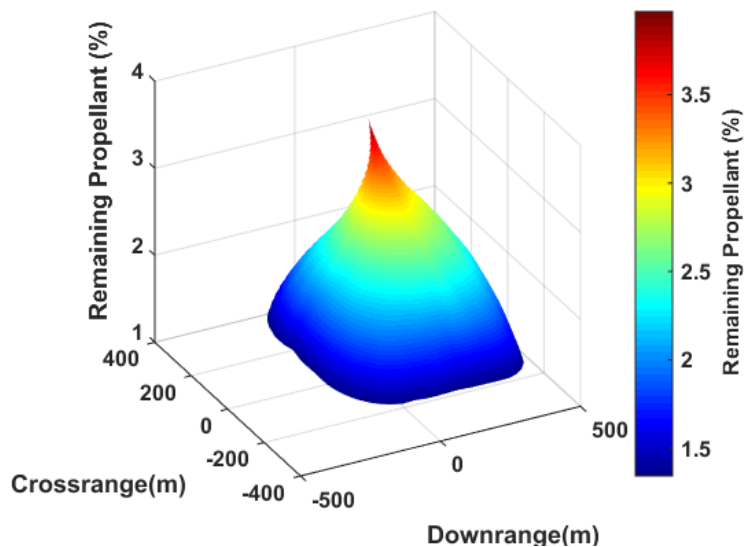


Figure 5.8: Evolution of the States.

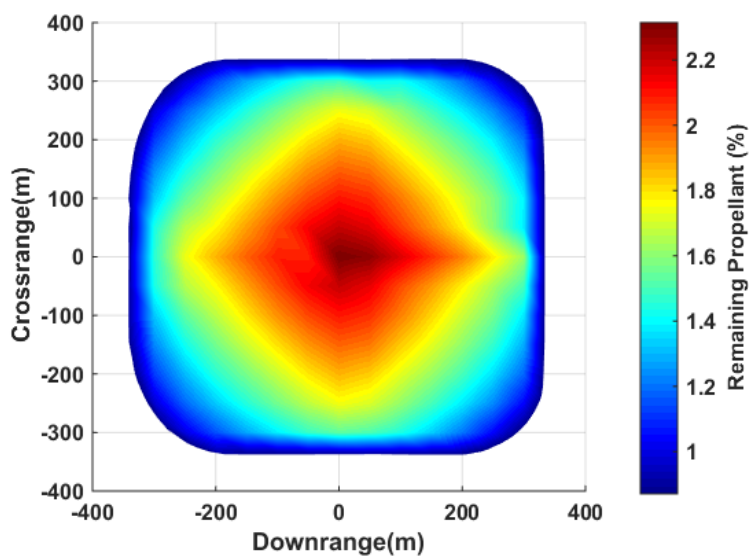


Figure 5.9: Reachable Set with  $\dot{d}_0 = \dot{c}_0 = 0$  (m/s).

Figure 5.10 shows the RS with  $\dot{d}_0 = \dot{c}_0 = 5$  (m/s). In this scenario, RS is shifted along positive downrange and crossrange coordinates w.r.t. the case with nominal initial states. Similarly, the maximum and minimum values for downrange and crossrange respectively are [-249 m, 423 m] and [-265 m, 406 m].

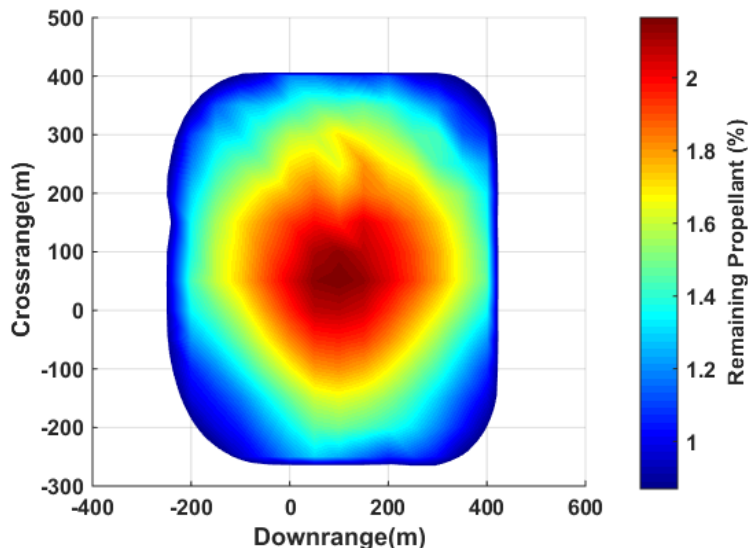


Figure 5.10: Reachable Set with  $\dot{d}_0 = \dot{c}_0 = 5$  (m/s).

### RS with different bounds on the constraints

In this case the bounds on the constraints are relaxed. Initially the angular rates were defined in the interval of  $[-2^\circ/\text{s}, 2^\circ/\text{s}]$ . If the angular rates are changed to  $[-15^\circ/\text{s}, 15^\circ/\text{s}]$  the RS as illustrated in Figure 5.11. As shown from the Figure 5.11, the spacecraft has more agility in the rotational dynamics, yielding to a more circular landing area. The maximum and minimum values for downrange and crossrange respectively are  $[-462 \text{ m}, 723 \text{ m}]$  and  $[-595 \text{ m}, 595 \text{ m}]$

Another important effect is the maximum available thrust. In addition to angular rates, the box constraint on the non-dimensional thrust  $T_m$  is modified from  $[0.778, 1, 222]$  to  $[0, 5]$ . The RS computed with this assumption is shown in Figure 5.12. As illustrated, the size of the RS changes considerably. In this case, the maximum and minimum values for downrange and crossrange is  $[-1662 \text{ m}, 1859 \text{ m}]$  and  $[-1761 \text{ m}, 1761 \text{ m}]$ .

## 5.4 Reachable Sets for Simplified Dynamical Model

In the final scenario, the RS is computed for slightly modified nonlinear dynamical system. The RCS is omitted and the thrust is provided with the throttable main engine. The simplified dynamical system is defined with state vector in Eq. (3.37) and control vector in Eq. (3.38). The bounds on the angular rates and thrust is the same with the previous scenario. As shown in Figure 5.13, the maximum and minimum values for downrange and crossrange is  $[-1727 \text{ m}, 1917 \text{ m}]$  and  $[-1823 \text{ m}, 1823 \text{ m}]$  with a slight increase in the area of the RS.

The coordinates of the RS for different scenarios are summarized in Table 5.3.

## 5. SET COMPUTATIONS

---

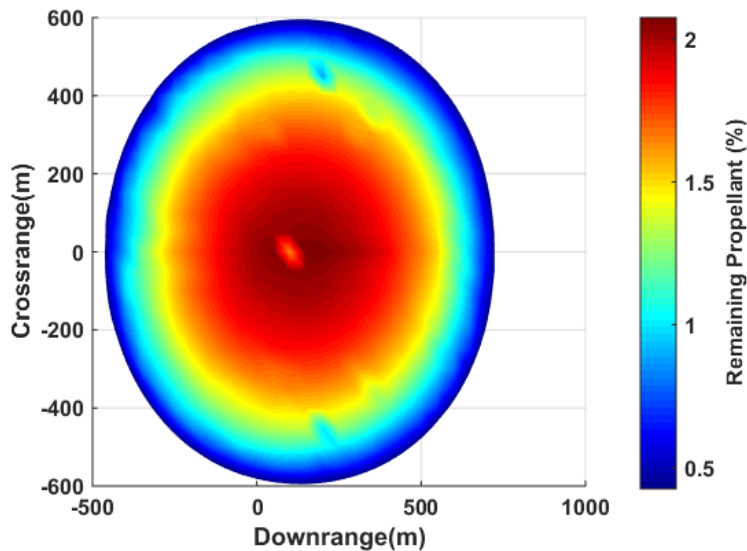


Figure 5.11: RS with  $-15^\circ/\text{s} < \omega < 15^\circ/\text{s}$ .

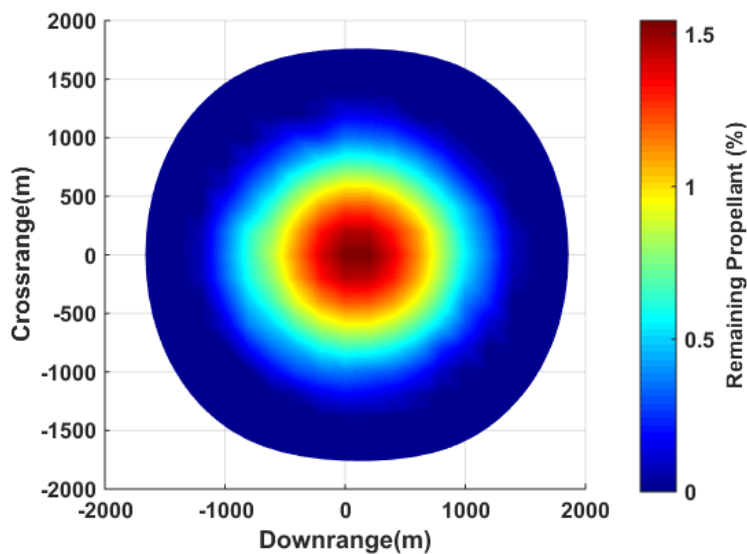


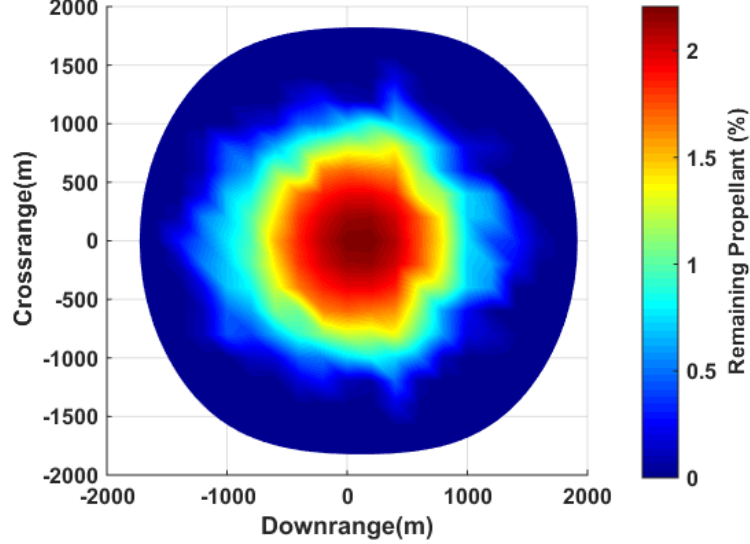
Figure 5.12: RS with Relaxed Angular Rates and  $T_m$ .

It can be concluded that:

- The size of the RS depends on the initial states, constraints on the control inputs, final time and the dynamical model.
- If the final time is a free parameter to the optimization problem, the size of the RS increases substantially. In this case, the limiting factor for the size of the RS



## 5.4 Reachable Sets for Simplified Dynamical Model



**Figure 5.13:** RS with Simplified Dynamical Model.

**Table 5.3:** Summary of RS Coordinates for Different Scenarios.

	Downrange (m)		Crossrange (m)	
	Min.	Max.	Min.	Max.
Nominal RS	-249	423	-338	338
RS with free $t_f$	-1161	1729	-1462	1462
RS with $\dot{d}_0 = \dot{c}_0 = 0$ m/s	-341	334	-338	338
RS with $\dot{d}_5 = \dot{c}_0 = 5$ m/s	-249	423	-265	406
RS with $-15(^{\circ}/s) < \omega < 15(^{\circ}/s)$	-462	723	-595	595
RS with relaxed angular rates and $T_m$	-1662	1859	-1761	1761
RS for simplified dynamical model	-1727	1917	-1823	-1823

is the amount of remaining propellant.

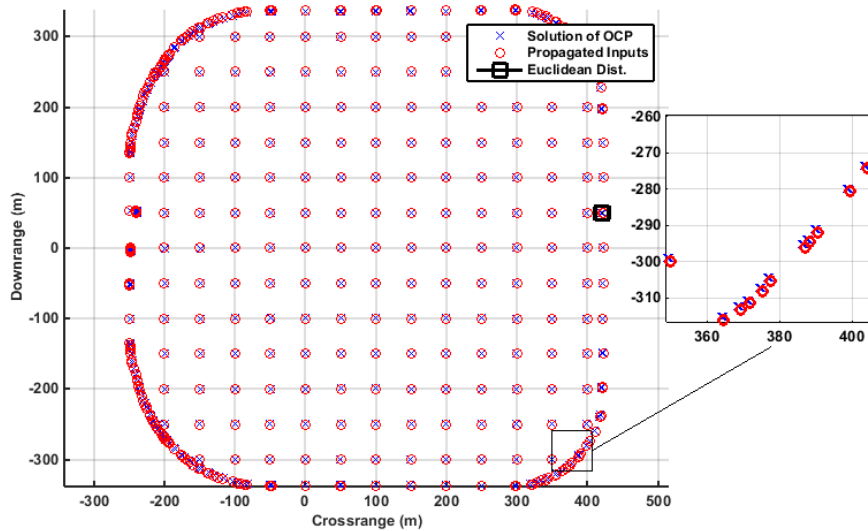
- When the allowed rates for pitch and yaw is increased, the agility of the lander increases. This results in a more circular RS. It can be deduced that, the rectangular-like RS in downrange-crossrange plane is due to the rate limits. In the previous scenarios, rate limits were dominant factor on the shape of the RS.
- Available maximum thrust is yet another important factor for the size of the RS. The safe landing area increases with increasing  $T_m$ . However it should be noted that there might be limitations for the allowed minimum thrust level for throttleable engines.
- When the RCS is omitted from the dynamics of the lander, the shape and the size of the RS does not change marginally. The components of main thrust vector

## 5. SET COMPUTATIONS

provide enough force to steer the lander to regions with RCS.

### 5.5 Validation of the Results

As previously discussed, control inputs obtained by solving NLPs are propagated with Runge-Kutta integration methods and compared with state information to assess the quality of the solution. For the first scenario with  $t_f = 40$  s, the Euclidean distance between two methods is 1.01 m as shown in Figure 5.14. Similarly, in the second scenario where  $t_f$  is defined as free parameter, the distance is 9.38 m as in the Figure 5.15. For both cases, the maximum error occurs at the boundary of the RS. It can be concluded that, it is numerically more demanding to find a solution for the points close to the boundary of the RS.



**Figure 5.14:** The Euclidean Distance for RS with  $t_f = 40(s)$ .

The number of nodes for discretization of the associated OCP is also another important parameter for the accuracy of the computed RS. Table 5.4 shows the accuracy of the solution based on number of discretization nodes and computation time for each case. The computations are performed using an Intel i7 Quad Core CPU operating at 3.40 Ghz and 32 GB of RAM. The RS computed with 90 nodes is illustrated in Figure 5.16.

As the number of nodes for the discretization of the OCP increases, the Euclidean distance between the solution of RS and final states obtained by propagating the control inputs is decreasing. On the other hand, as the size of the problem increases due to the increased number of nodes, the computation time also increases. As expected, there is a compromise between the accuracy of the solution and the computation time.

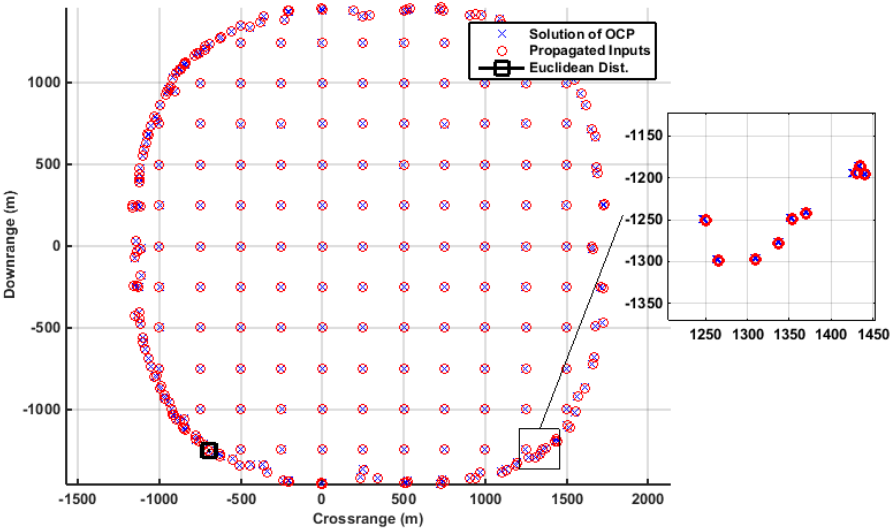


Figure 5.15: The Euclidean Distance for RS with free  $t_f$ .

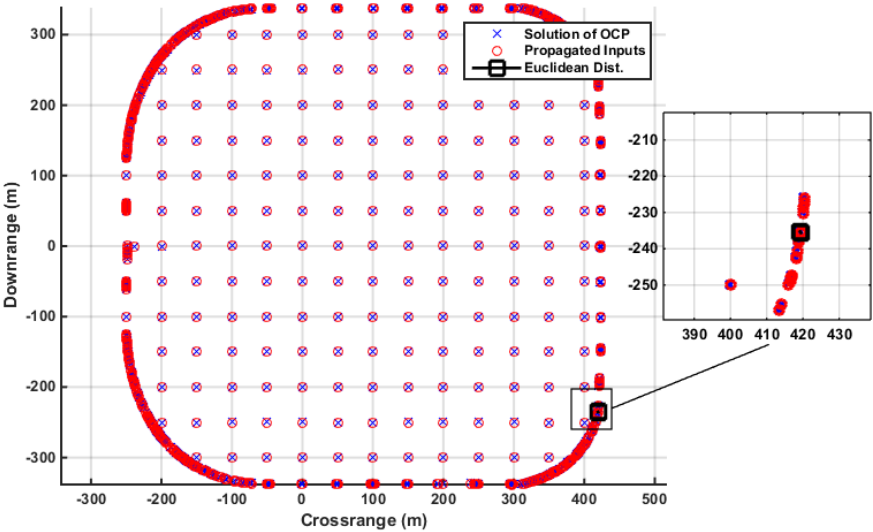


Figure 5.16: The Euclidean Distance for RS with free  $t_f$  and 90 Nodes for Discretizing the OCP.

When 90 nodes are used for the discretization of the problem, the computation of RS takes approximately 92 hours with an accuracy of 0.3513 m between the solution obtained by solving the NLP, and the states obtained by propagating the control inputs. On the other hand, the computation takes approximately 14 hours with an accuracy

## 5. SET COMPUTATIONS

---

**Table 5.4:** Accuracy of the RS Computation Based on Number of Discretization Nodes.

Number of Nodes	Computation Time	Euclidean Distance (m)
30	14hr 12m 59s	2.5703
40	18hr 51m 58s	1.4655
50	28hr 13m 43s	1.0179
60	32hr 13m 19s	0.7821
70	47hr 54m 29s	0.5915
80	77hr 48m 29s	0.4147
90	91hr 51m 29s	0.3513

of 2.5703 m when 30 nodes are used to discretize the problem. In these scenarios the maximum error for the solution of the problem is much smaller than the discretization of the landing area (50 m-250 m) in downrange-crossrange plane. Therefore, in order to be conservative 50 nodes are used for the discretization of the OCP to compute the reference RS set, which is used in the subsequent chapters.

## 6

# Polynomial Guidance

In order to find the reachable set for moon landing problem, the candidate landing area is discretized in crossrange-downrange plane by equidistant rectangular grid. For each grid point, an OCP is formulated to find a feasible trajectory between the initial point of the maneuver to the respective grid point without violating the safe landing condition. Later on, the continuous OCP is transcribed using nonuniform discretization points in time to obtain an associated NLP. Finally, the NLP is solved using available solvers satisfying optimality conditions.

The major drawback of this approach is the computation time. In order to find reachable set, all the grid points are tested if they are reachable. Moreover, the verification of the solution is provided by just propagating the control inputs obtained from the solving the NLP problem. Therefore, there is a need for comparing the proposed guidance method, which is used to approximate the reachable set, with another algorithm and compare the performance of both methods.

In literature there are various methods for the pinpoint landing (PPL) problem. The basic PPL guidance problem is defined as finding a trajectory for the lunar lander between any given state at the beginning of terminal landing maneuver to a desired terminal state (e.g, candidate grid point) without violating any state constraints (e.g., reaching the target without flying subsurface) and control constraints (e.g., actuator saturation) with the available fuel.

There are various approaches to obtain numerical and approximate solutions for fuel optimal PPL guidance problem. Sostaric and Rea (2005)<sup>64</sup> converted the guidance problem into a constrained parameter optimization problem using both direct collocation and direct multiple shooting methods and numerical solutions are obtained for various scenarios. Najson and Mease (2005)<sup>84</sup> developed an analytical method by approximating the original problem by solving a simpler optimal control problem without fuel considerations in the objective functional. Aıkmee and Ploen (2005)<sup>71</sup> showed that the fully constrained minimum-fuel guidance problem can be reduced to a finite-dimensional convex-programming problem by convexification and interior point methods are used to find the optimal solution with guaranteed convergence to any desired level of accuracy for given state and control constraints.

## 6. POLYNOMIAL GUIDANCE

---

The last method is also used in landing phase of the Mars Science Laboratory and proven to be successful. In order to keep the computations simple, more importance could be put on finding a feasible solution, rather than a fuel-optimal solution.

One of the proven guidance methods is the polynomial based trajectory generation which was used for Apollo missions. The Apollo guidance algorithms were developed to be executed with a limited computational capability to accomplish a critical task of sending men to Moon and return them safely to Earth.

### 6.1 Apollo Guidance Algorithm

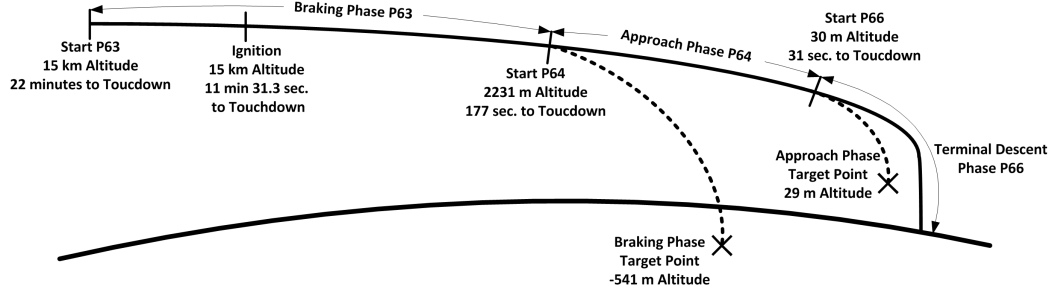
For the Apollo missions, the lunar landing guidance consists of several phases. In the braking and approach phase the aim is to reduce the velocity of the vehicle from orbital speed. Once these phases are completed, the velocities in the downrange, crossrange and altitude are close to their final values. During the terminal landing phase, the G&C system steers the vehicle from low gate to the landing point. The Apollo landing is compromised of the approach phase, braking phase and the terminal descent phase. These phases use this same guidance algorithm with differences in the reference constants.

The lunar descent is a nominally-planar trajectory in which the three phases as illustrated in Figure 6.1 are:

- The braking phase (or P63) is initiated manually about 10 minutes before the normal ignition time. In this phase the time and attitude for ignition is computed. Latern, the descent propulsion system (DPS) is ignited. Finally, the Lunar Module (LM) is brought into the required initial conditions the approach phase.
- Approach phase (P64) guidance begins with initial conditions consisting of 2.2 km altitude and 7.5 km ground range from the landing point. LM is almost hovers at the top of the landing point. In this phase, the commander directs the lunar guidance computer (LGC) to land at any pre-defined landing point until initiation of the terminal descent phase. The pilot visually identifies the current landing site and then steers the vehicle onto a trajectory coincident with another visually selected landing site. This procedure is continues as the pilot commands incremental displacements until the states for the terminal descent is achieved.
- The terminal descent phase (P66) begins at around 30 m altitude and 11 m ground range from the landing site, or it may be initiated by the commander during the approach phase. In this phase only velocities are controlled. The forward and lateral velocity components are nulled to have just a vertical approach to the lunar surface. This is done autonomously due to the removing sand blocks the view of the commander. Altitude rate is controlled incrementally until touchdown.

In terms of a vector polynomial function of target referenced time, the reference trajectory is defined such that it satisfies a two point boundary value problem with a

## 6.1 Apollo Guidance Algorithm



**Figure 6.1:** Apollo Mission Descent Guidance Phases and Targets.

total of five degrees of freedom for each of the three components. It can be defined as:

$$\text{RRG} = \text{RTG} + \text{VTG}T_{go} + \text{ATG}\frac{T_{go}^2}{2} + \text{JTG}\frac{T_{go}^3}{6} + \text{STG}\frac{T_{go}^4}{12} \quad (6.1)$$

where RRG is the position vector on the reference trajectory at the current (negative) time-to-go  $T_{go}$  and RTG, VTG, ATG, JTG, and STG, are the targeted position, velocity, acceleration, jerk, and snap. These states are all given in a guidance reference frame that has its origin at the specified landing point.

Because of the quartic form for the approach phase reference trajectory, fifteen constraints must be specified (five coefficients in three dimensions). First, the trajectory is confined to two dimensions, thereby reducing the unknown variables to ten. Then, according to Klumpp (1971)<sup>63</sup>, if the approach phase target point, velocity, and time are selected along with a phase inception position and time and an additional midpoint position, velocity, and time, there exists a closed form solution for the remaining ten unknown variables that define RTG, VTG, ATG, JTG, and STG.

If we change the jerk and snap with some constants the acceleration profile for the vehicle is defined as

$$a(t) = c_0 + c_1t + c_2t^2 \quad (6.2)$$

where  $c_i \in \mathbb{R}$  are coefficients to be determined. Integrating Eq. (6.2) twice yields

$$\begin{aligned} v(t) &= v_0 + c_0t + \frac{1}{2}c_1t^2 + \frac{1}{3}c_2t^3 \\ r(t) &= r_0 + v_0t + \frac{1}{2}c_0t^2 + \frac{1}{6}c_1t^3 + \frac{1}{12}c_2t^4 \end{aligned} \quad (6.3)$$

with  $r_0$  and  $v_0$  are the initial position and the velocity of the spacecraft respectively. For given boundary conditions  $r(t_f) = r_f, v(t_f) = \dot{r}_f$  and  $a(t_f) = \ddot{r}_f$ , the coefficients

## 6. POLYNOMIAL GUIDANCE

---

are solved yielding

$$\begin{aligned}
 c_0 &= \ddot{r}_f - \frac{6}{t_f}(\dot{r}_f - \dot{r}_0) + \frac{12}{t_f^2}(r_f - r_0 - \dot{r}_0 t_f) \\
 c_1 &= -\frac{6}{t_f}\ddot{r}_f + \frac{30}{t_f^2}(\dot{r}_f - \dot{r}_0) - \frac{48}{t_f^3}(r_f - r_0 - \dot{r}_0 t_f) \\
 c_2 &= \frac{6}{t_f^2}\ddot{r}_f - \frac{24}{t_f^3}(\dot{r}_f - \dot{r}_0) + \frac{36}{t_f^4}(r_f - r_0 - \dot{r}_0 t_f)
 \end{aligned} \tag{6.4}$$

Therefore, the trajectory defining the vehicle position can be found for the given boundary conditions. A significant drawback of the quadratic Apollo guidance law is that it does not contain any free parameters. During the Apollo missions, the objective was to find a trajectory by fixing final time and having a linear acceleration profile.

A line search can be performed over  $t_f \in [t_{min}, t_{max}]$  and the trajectory with lowest propellant cost can be evaluated at discrete times. The propellant optimal  $t_f^*$ , is then can be used in the Apollo guidance law. However, this method is not implemented in the construction of the reachable set, as the aim is to find a feasible set, then finding the set with different final times.

### 6.2 Extended Apollo Guidance Algorithm

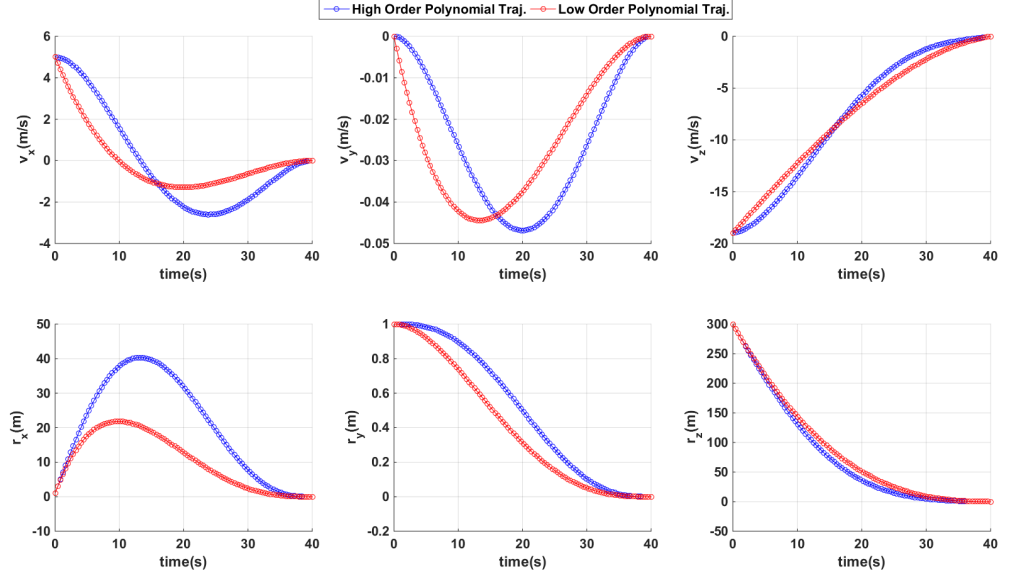
As discussed in the previous section, the guidance algorithm can be used for a known initial position, velocity and final acceleration, velocity and position. Compared to the requirements one has to make sure that the initial acceleration is also taken into account. Therefore, the degree of the acceleration profile is increased one more, such that it satisfies boundary conditions related to initial acceleration. The comparison of two trajectory profiles is shown in Figure 6.2. The low order polynomial trajectory does not take into account the initial acceleration of the vehicle. On the other hand, the high order polynomial trajectory satisfies the given initial acceleration constraint. This is needed during the comparison w.r.t the reference RS, since the acceleration in three orthogonal direction is already defined.

Increasing the order polynomial for the acceleration in 6.2 yields

$$\begin{aligned}
 a(t) &= c_0 + c_1 t + c_2 t^2 + c_3 t^3 \\
 v(t) &= v_0 + c_0 t + \frac{1}{2}c_1 t^2 + \frac{1}{3}c_2 t^3 + \frac{1}{4}c_3 t^4 \\
 r(t) &= r_0 + v_0 t + \frac{1}{2}c_0 t^2 + \frac{1}{6}c_1 t^3 + \frac{1}{12}c_2 t^4 + \frac{1}{20}c_3 t^5
 \end{aligned} \tag{6.5}$$

Known values are  $a_0, v_0, r_0, a_f, v_f, r_f$  and  $t_f$ . Solving for coefficients  $c_0, c_1, c_2, c_3$





**Figure 6.2:** Comparison of High Order and Low Order Polynomial Trajectories.

yields

$$\begin{aligned}
 c_0 &= a_0 \\
 c_1 &= \frac{3(20r_f - 20r_0 - 8v_f t_f - 12v_0 t_f + a_f t_f^2 - 3a_0 t_f^2)}{t_f^3} \\
 c_2 &= -\frac{6(30r_f - 30r_0 - 14v_f t_f - 16v_0 t_f + 2a_f t_f^2 - 3a_0 t_f^2)}{t_f^4} \\
 c_3 &= \frac{10(12r_f - 12r_0 - 6v_f t_f - 6v_0 t_f + a_f t_f^2 - a_0 t_f^2)}{t_f^5}
 \end{aligned} \tag{6.6}$$

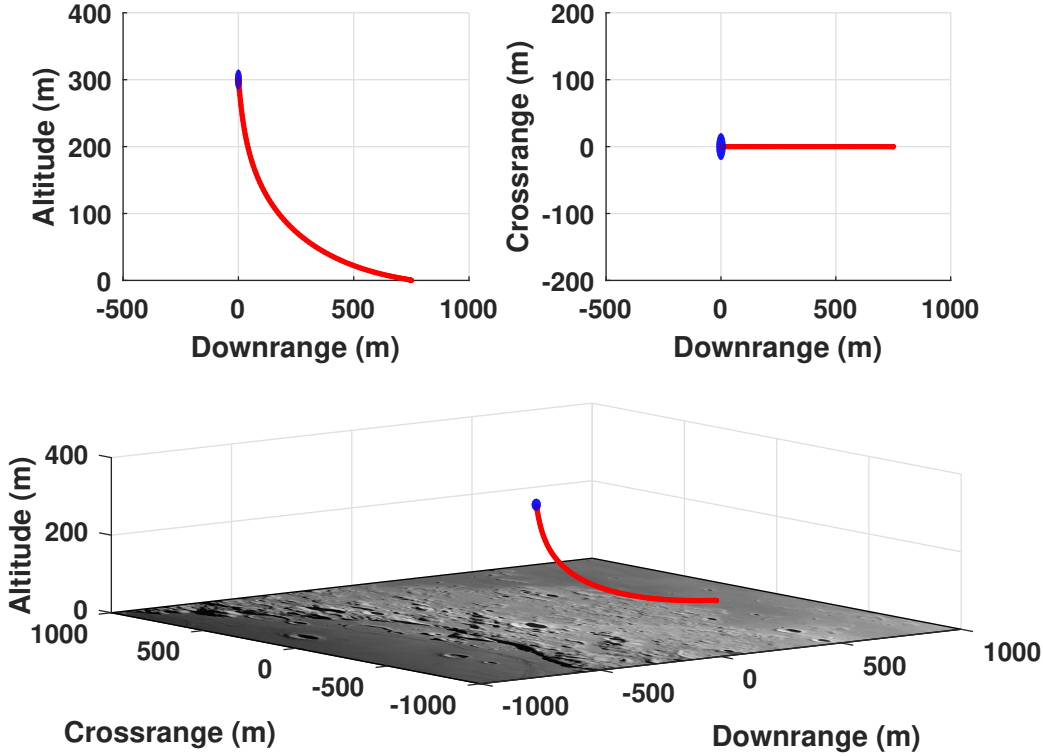
The acceleration, velocity and position as a function of time are obtained by substituting the coefficients found in Eq. (6.6) into Eq. (6.5). The state trajectories for a sample landing scenario are shown in Figure 6.3 and Figure 6.4.

### 6.3 Lyapunov based PD Controller

Once the target trajectory has been created, a tracking controller must be designed. The main assumption in designing the reference trajectory using polynomials is that the spacecraft is a point mass. Therefore the mass of the vehicle is not incorporated in the equations of the motion. In order to eliminate this discrepancy, there is a need for feedback controller in addition to the tracking controller.

During the Apollo missions, the flight computer was not able to execute the necessary

## 6. POLYNOMIAL GUIDANCE



**Figure 6.3:** Sample Trajectory - Downrange/Crossrange/Altitude. The initial starting point for the maneuver is highlighted.

control actions. In order to overcome this problem, further simplifications are included in the controller design. However, there are various control methods available for simulation purposes. The aim of the section is to introduce a closed-form solution for the guidance and control of the lander.

For this purpose a Lyapunov based PD controller is designed. This controller is implemented for a similar soft landing problem by Chomel<sup>85</sup>. The candidate Lyapunov function is taken from this reference. This method guarantees the convergency of the states to the target trajectory. The aim of the proof is to show that the altitude  $h$ , downrange  $d$ , and crossrange  $c$ , all converge to their respective target states denoted by  $h_{ref}$ ,  $d_{ref}$ ,  $c_{ref}$ . After the tracking controller is designed, the nonlinear dynamical system is inverted to obtain the controller inputs that lead to the given reference trajectories. The performance of this exact linearization will be demonstrated on tracking sample trajectories for the terminal landing phase. It should also be noted that polynomial based closed loop system assumes flat surface with non-rotating reference frame. Therefore the substitution can be done in local coordinates for  $x \rightarrow d$ ,  $y \rightarrow c$ ,  $z \rightarrow h$ .

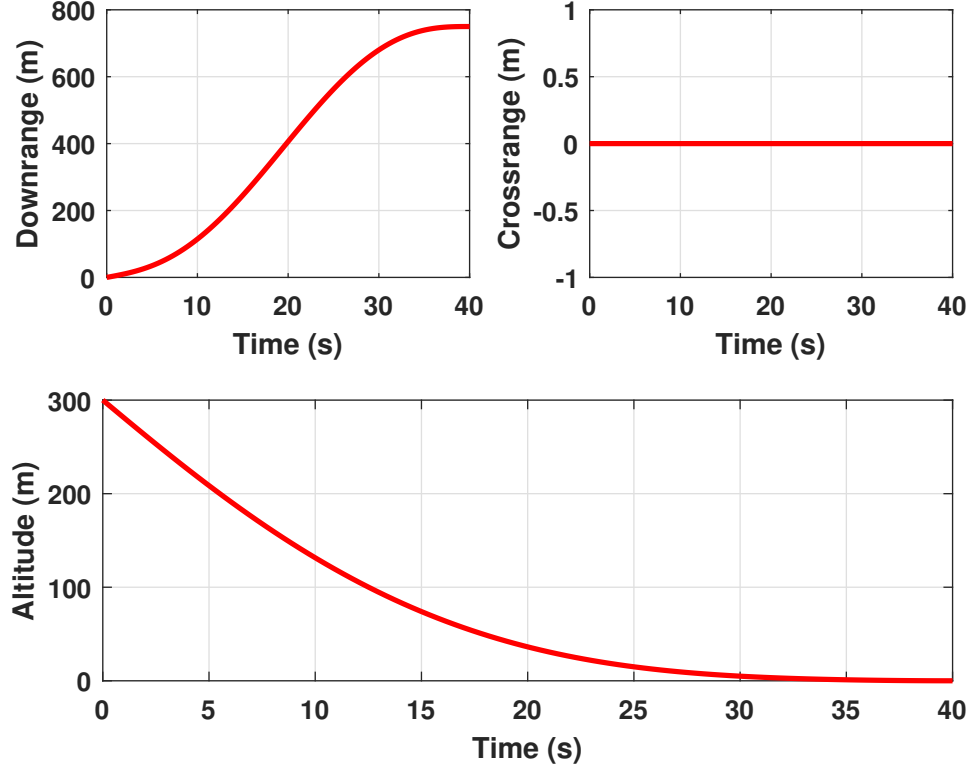


Figure 6.4: Sample Trajectory - Downrange/Crossrange/Altitude vs. Time.

The Lyapunov candidate function chosen for this application is

$$\begin{aligned}
 V = & \frac{1}{2}[\dot{h}_{err}^2 + \dot{d}_{err}^2 + \dot{c}_{err}^2 + \frac{1}{\gamma_h}k_h^2 + \frac{1}{\gamma_d}k_d^2 + \frac{1}{\gamma_c}k_c^2] + \\
 & \frac{\beta_h A}{\lambda_h} \ln \cosh \lambda_h h_{err} + \frac{\beta_d A}{\lambda_d} \ln \cosh \lambda_d d_{err} + \frac{\beta_c A}{\lambda_c} \ln \cosh \lambda_c c_{err}
 \end{aligned} \tag{6.7}$$

where

$$\begin{aligned}
 h_{err} &= h - h_{ref} & d_{err} &= d - d_{ref} & c_{err} &= c - c_{ref} \\
 \dot{h}_{err} &= \dot{h} - \dot{h}_{ref} & \dot{d}_{err} &= \dot{d} - \dot{d}_{ref} & \dot{c}_{err} &= \dot{c} - \dot{c}_{ref} \\
 \ddot{h}_{err} &= \ddot{h} - \ddot{h}_{ref} & \ddot{d}_{err} &= \ddot{d} - \ddot{d}_{ref} & \ddot{c}_{err} &= \ddot{c} - \ddot{c}_{ref}
 \end{aligned} \tag{6.8}$$

and the variables  $k_h$ ,  $k_d$ , and  $k_c$  are tuning functions, the value  $A$  is a positive value for the maximum expected acceleration. The gains  $\lambda_h$ ,  $\lambda_d$ ,  $\lambda_c$ ,  $\gamma_h$ ,  $\gamma_d$ , and  $\gamma_c$  are positive and  $0 \leq \beta_h, \beta_d, \beta_c \leq 1$ .

Lyapunov function  $V$  is a positive ( $V \geq 0$ ) with these definitions (considering  $\cosh x \geq 1$  so  $\ln \cosh x \geq 0$  for all  $x$ ). The tuning functions  $k_h$ ,  $k_d$ , and  $k_c$  increase the complexity of the controller, but the performance is well balanced by the proportional

## 6. POLYNOMIAL GUIDANCE

---

control with the derivative control. First step is to show that the Lyapunov function satisfies stability conditions. Taking the derivative of the Lyapunov function yields

$$\begin{aligned}\dot{V} = & \dot{h}_{err}(\ddot{h}_{err} + \beta_h A \tanh \lambda_h h_{err}) + \frac{1}{\gamma_h} k_h \dot{k}_h + \\ & \dot{d}_{err}(\ddot{d}_{err} + \beta_d A \tanh \lambda_d d_{err}) + \frac{1}{\gamma_d} k_d \dot{k}_d + \\ & \dot{c}_{err}(\ddot{c}_{err} + \beta_c A \tanh \lambda_c c_{err}) + \frac{1}{\gamma_c} k_c \dot{k}_c.\end{aligned}\quad (6.9)$$

If the control law makes the following true

$$\begin{aligned}\ddot{h}_{err} + \beta_h A \tanh \lambda_h h_{err} &= -(1 - \beta_h) A \tanh \lambda_h (\dot{h} + k_h h_{err}) \\ \ddot{d}_{err} + \beta_d A \tanh \lambda_d d_{err} &= -(1 - \beta_d) A \tanh \lambda_d (\dot{d} + k_d d_{err}) \\ \ddot{c}_{err} + \beta_c A \tanh \lambda_c c_{err} &= -(1 - \beta_c) A \tanh \lambda_c (\dot{c} + k_c c_{err})\end{aligned}\quad (6.10)$$

where  $\lambda_{h2}$ ,  $\lambda_{d2}$  and  $\lambda_{c2}$  are additional positive control gains then

$$\begin{aligned}\dot{V} = & -(\dot{h}_{err} + k_h h_{err})(1 - \beta_h) A \tanh \lambda_{h2} (\dot{h}_{err} + k_h h_{err}) \\ & + k_h \left( \frac{1}{\gamma_h} \dot{k}_h + h_{err} (1 - \beta_h) \right) A \tanh \lambda_{h2} (\dot{h}_{err} + k_h h_{err}) \\ & - (\dot{d}_{err} + k_d d_{err})(1 - \beta_d) A \tanh \lambda_{d2} (\dot{d}_{err} + k_d d_{err}) \\ & + k_d \left( \frac{1}{\gamma_d} \dot{k}_d + d_{err} (1 - \beta_d) \right) A \tanh \lambda_{d2} (\dot{d}_{err} + k_d d_{err}) \\ & - (\dot{c}_{err} + k_c c_{err})(1 - \beta_c) A \tanh \lambda_{c2} (\dot{c}_{err} + k_c c_{err}) \\ & + k_c \left( \frac{1}{\gamma_c} \dot{k}_c + c_{err} (1 - \beta_c) \right) A \tanh \lambda_{c2} (\dot{c}_{err} + k_c c_{err})\end{aligned}\quad (6.11)$$

If the tuning function is implemented such that

$$\begin{aligned}\dot{k}_h &= -\gamma_h h_{err} (1 - \beta_h) A \tanh \lambda_{h2} (\dot{h}_{err} + k_h h_{err}) \\ & \quad - \gamma_h h_{err} \tanh k_h h_{err} \\ \dot{k}_d &= -\gamma_d d_{err} (1 - \beta_d) A \tanh \lambda_{d2} (\dot{d}_{err} + k_d d_{err}) \\ & \quad - \gamma_d d_{err} \tanh k_d d_{err} \\ \dot{k}_c &= -\gamma_c c_{err} (1 - \beta_c) A \tanh \lambda_{c2} (\dot{c}_{err} + k_c c_{err}) \\ & \quad - \gamma_c c_{err} \tanh k_c c_{err}\end{aligned}\quad (6.12)$$

then

$$\begin{aligned}\dot{V} = & -(\dot{h}_{err} + k_h h_{err} (1 - \beta_h) A \tanh \lambda_{h2} (\dot{h}_{err} + k_h h_{err})) - k_h h_{err} \tanh k_h h_{err} \\ & - (\dot{d}_{err} + k_d d_{err} (1 - \beta_d) A \tanh \lambda_{d2} (\dot{d}_{err} + k_d d_{err})) - k_d d_{err} \tanh k_d d_{err} \\ & - (\dot{c}_{err} + k_c c_{err} (1 - \beta_c) A \tanh \lambda_{c2} (\dot{c}_{err} + k_c c_{err})) - k_c c_{err} \tanh k_c c_{err}\end{aligned}\quad (6.13)$$

meaning that  $\dot{V} \leq 0$  and  $V$  is a non-increasing function (any function  $x \tanh x \geq 0$ ). It has been shown that the Lyapunov candidate function  $V$  is positive and non-increasing if the gains  $\lambda_{h1}, \lambda_{d1}, \lambda_{c1}, \lambda_{h2}, \lambda_{d2}, \lambda_{c2}, \gamma_h, \gamma_d$  and  $\gamma_c$  are positive and  $0 \leq \beta_h, \beta_d, \beta_c \leq 1$ .

The condition for  $V$  to be bounded is that the each term of  $V$  must be also bounded. The second condition for  $V$  to be a suitable Lyapunov function is that the error states asymptotically approach zero as a function of time such that

$$\dot{h} \rightarrow \dot{h}_{ref}, \dot{d} \rightarrow \dot{d}_{ref}, \dot{c} \rightarrow \dot{c}_{ref}.$$

The comprehensive proof for the above conditions is shown by Chomel (2007)<sup>85</sup>. After designing the tracking controller, recall the equations of motion of the lander for the translational motion with three control inputs. The main reason for the reduction of number of control inputs is that, during the nonlinear conversion of the dynamical equations, the solution is not unique. However, if a polynomial based trajectory is generated, three control inputs can be found from the inversion of the nonlinear dynamical system which steer the spacecraft for the provided reference trajectory.

$$\mathbf{T}_{MCI} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} -\cos(\beta - \frac{d}{r})(T_m) \\ -\sin(\beta - \frac{d}{r})(T_m) \\ T_m \sin \chi \end{pmatrix} \quad (6.14)$$

$$\begin{pmatrix} \ddot{d} \\ \ddot{h} \\ \ddot{c} \end{pmatrix} = \begin{pmatrix} \frac{r}{m\tilde{h}c\tilde{c}}(-T_1c\tilde{d} + T_2s\tilde{d}) + 2\dot{d}(\frac{\dot{c}}{r}\tilde{t}\tilde{c} - \frac{\dot{h}}{\tilde{h}}) \\ \frac{1}{m}[(-T_1s\tilde{d} - T_2c\tilde{d})c\tilde{c} - T_3s\tilde{c}] + [(\dot{d}c\tilde{c})^2 + \dot{c}^2]\frac{\tilde{h}}{r^2} - \frac{MG}{(\tilde{h})^2} \\ \frac{r}{m(\tilde{h})}[(T_1s\tilde{d} + T_2c\tilde{d})s\tilde{c} - T_3c\tilde{c}] - \frac{\dot{d}^2}{r}s\tilde{c}\tilde{c} - \frac{2\dot{c}\dot{h}}{\tilde{h}} \end{pmatrix} \quad (6.15)$$

$$\begin{aligned} \tilde{h} &= h + r, & s\tilde{d} &= \sin \frac{d}{r}, & c\tilde{d} &= \cos \frac{d}{r} \\ s\tilde{c} &= \sin \frac{c}{r}, & c\tilde{c} &= \cos \frac{c}{r}, & \tilde{t}\tilde{c} &= \tan \frac{c}{r} \end{aligned}$$

Solving for  $T_m, \beta, \chi$  in Eq. (6.15) for given  $d, h, c$  and their respective derivatives yields

$$\begin{aligned} \beta &= \arctan(N_1/N_2) + d/r \\ T_m &= \sqrt{N_1^2 + N_2^2 + N_3^2} \\ \chi &= \arcsin(-N_3/T_m) \end{aligned} \quad (6.16)$$

## 6. POLYNOMIAL GUIDANCE

---

where

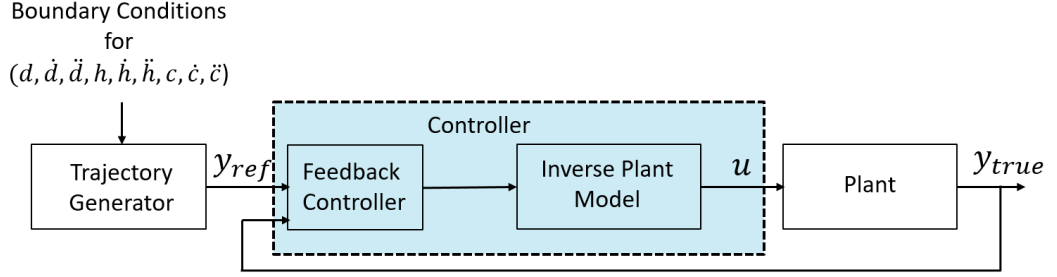
$$\begin{aligned}
N_1 &= \frac{-m}{r^2(h+r)^2} \cos\left(\frac{c}{r}\right) \left( (h+r)^2 \cos\left(\frac{d}{r}\right) \left( r(2\dot{d}\dot{h} + \ddot{d}(h+r)) - 2\dot{c}\dot{d}(h+r) \tan\left(\frac{c}{r}\right) \right) \right. \\
&\quad \left. + \sin\left(\frac{d}{r}\right) \left( -\dot{c}^2(h+r)^3 - \dot{d}^2(h+r)^3 + r^2(\text{GM} + \ddot{h}(h+r)^2) - r(h+r)^2 \right) \right. \\
&\quad \left. \left( 2\dot{c}\dot{h} + \ddot{c}(h+r) \arctan\left(\frac{c}{r}\right) \right) \right) \\
N_2 &= \frac{m}{r^2(h+r)^2} \cos\left(\frac{c}{r}\right) \sin\left(\frac{d}{r}\right) \left( (h+r)^2 \left( r(2\dot{d}\dot{h} + \ddot{d}(h+r)) - 2\dot{c}\dot{d}(h+r) \tan\left(\frac{c}{r}\right) \right) \right. \\
&\quad \left. + \cot\left(\frac{d}{r}\right) \left( \dot{c}^2(h+r)^3 + \dot{d}^2(h+r)^3 - r^2(\text{GM} + \ddot{h}(h+r)^2) \right) \right. \\
&\quad \left. + r(h+r)^2 (2\dot{c}\dot{h} + \ddot{c}(h+r)) \tan\left(\frac{c}{r}\right) \right) \\
N_3 &= \frac{-m}{r^2(h+r)^2} \sin\left(\frac{c}{r}\right) \left( \left( -\dot{c}^2(h+r)^3 + r^2(\text{GM} + \ddot{h}(h+r)^2) \right) \right. \\
&\quad \left. + r(h+r)^2 (2\dot{c}\dot{h} + \ddot{c}(h+r)) \right)
\end{aligned} \tag{6.17}$$

Hence we use the Lyapunov-based PD controller

$$\begin{aligned}
\ddot{d}_{err} &= -\phi_d A \tanh \alpha_d \theta_{d_1} \\
&\quad - (1 - \phi_d) A \tanh \theta_{d_2} (\dot{d}_{err} + k_d d_{err}) \\
\ddot{h}_{err} &= -\phi_h A \tanh \alpha_h \theta_{h_1} \\
&\quad - (1 - \phi_h) A \tanh \theta_{h_2} (\dot{h}_{err} + k_h h_{err}) \\
\ddot{c}_{err} &= -\phi_c A \tanh \alpha_c \theta_{c_1} \\
&\quad - (1 - \phi_c) A \tanh \theta_{c_2} (\dot{c}_{err} + k_c c_{err})
\end{aligned} \tag{6.18}$$

with tuning functions

$$\begin{aligned}
\dot{k}_d &= -\gamma_d d_{err} (1 - \phi_d) A \tanh \theta_{d_2} (\dot{d}_{err} + k_d d_{err}) \\
&\quad - \gamma_d d_{err} \tanh k_d d_{err} \\
\dot{k}_h &= -\gamma_h h_{err} (1 - \phi_h) A \tanh \theta_{h_2} (\dot{h}_{err} + k_h h_{err}) \\
&\quad - \gamma_h h_{err} \tanh k_h h_{err} \\
\dot{k}_c &= -\gamma_c c_{err} (1 - \phi_c) A \tanh \theta_{c_2} (\dot{c}_{err} + k_c c_{err}) \\
&\quad - \gamma_c c_{err} \tanh k_c c_{err}
\end{aligned} \tag{6.19}$$



**Figure 6.5:** Block diagram of the closed-loop system.

Previously it was proved that the following statements holds

$$\begin{aligned}
 h_{err} &= h - h_{ref} & d_{err} &= d - d_{ref} & c_{err} &= c - c_{ref} \\
 \dot{h}_{err} &= \dot{h} - \dot{h}_{ref} & \dot{d}_{err} &= \dot{d} - \dot{d}_{ref} & \dot{c}_{err} &= \dot{c} - \dot{c}_{ref} \\
 \ddot{h}_{err} &= \ddot{h} - \ddot{h}_{ref} & \ddot{d}_{err} &= \ddot{d} - \ddot{d}_{ref} & \ddot{c}_{err} &= \ddot{c} - \ddot{c}_{ref}
 \end{aligned} \tag{6.20}$$

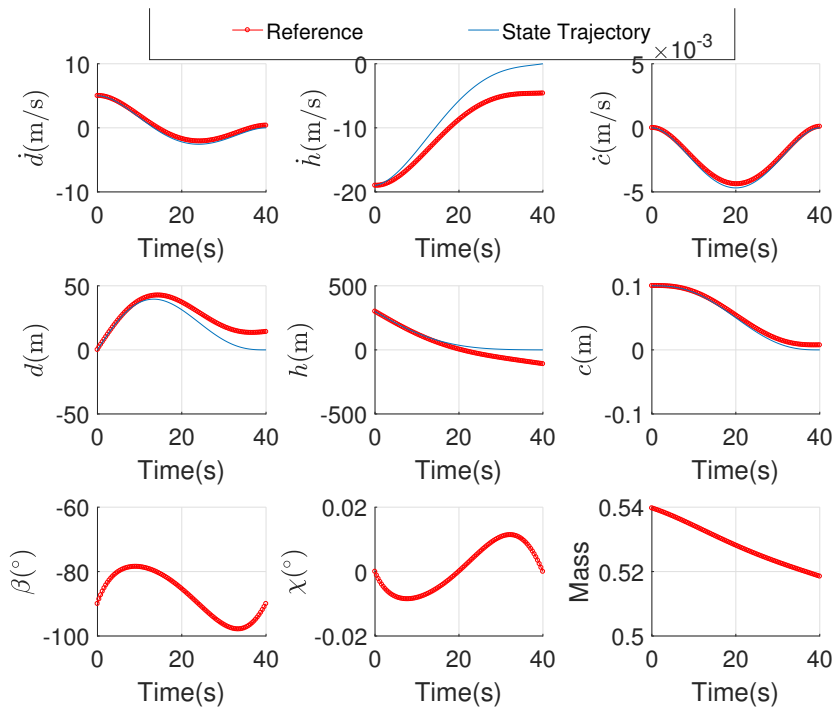
where  $d_{ref}$ ,  $h_{ref}$ ,  $c_{ref}$  denote the reference trajectories. Figure 6.5 shows the block diagram of the feedback controller with model inversion.

For reference trajectories, ignoring the curvature and rotation of the Moon and working in local coordinates, the downrange-crossrange-altitude and their respective derivatives are substituted with local  $x - y - z$  coordinates, where  $x$ ,  $v_x$ ,  $a_x$  are the position, velocity and acceleration in local  $x - y - z$ -coordinate system. Therefore the control inputs as a function of time can be obtained by using these variables in Eq. (6.16).

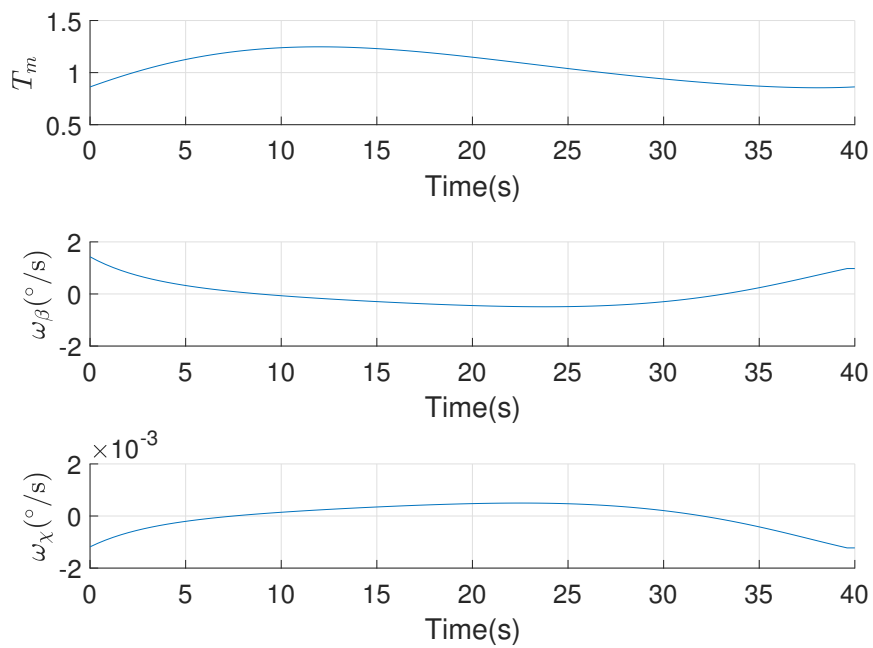
## 6.4 Polynomial Guidance Based RS

Figure 6.6 shows the state trajectories and Figure 6.8 control inputs for the open loop system in a sample scenario. The reference trajectories are generated using polynomials. However, there is no controller which attenuates the discrepancy due to the assumptions used during inversion of the nonlinear dynamical system. Final time is fixed with  $t_f = 40$  s. The reference trajectory is generated for a point mass, and the total mass of the spacecraft is decreasing due to propellant consumption. Figure 6.7 shows the state error for this scenario. As illustrated, the deviation of the states from the reference trajectory is increasing with time.

## 6. POLYNOMIAL GUIDANCE

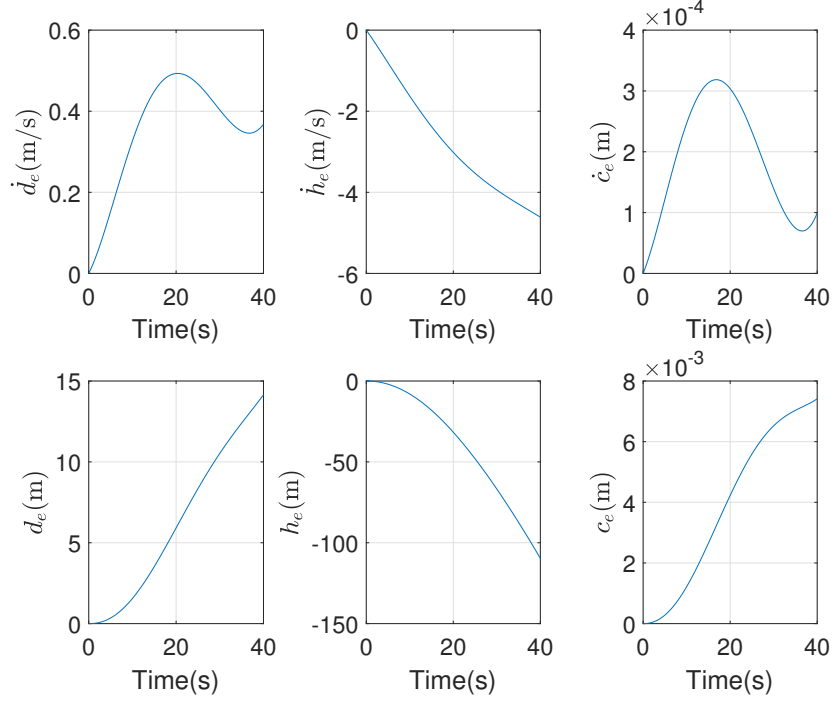


**Figure 6.6:** States - w/o Feedback Controller for Polynomial Reference Trajectory.



**Figure 6.8:** Controls - w/o Feedback Controller for Polynomial Reference Trajectory.





**Figure 6.7:** Tracking Error - w/o Feedback Controller for Polynomial Reference Trajectory.

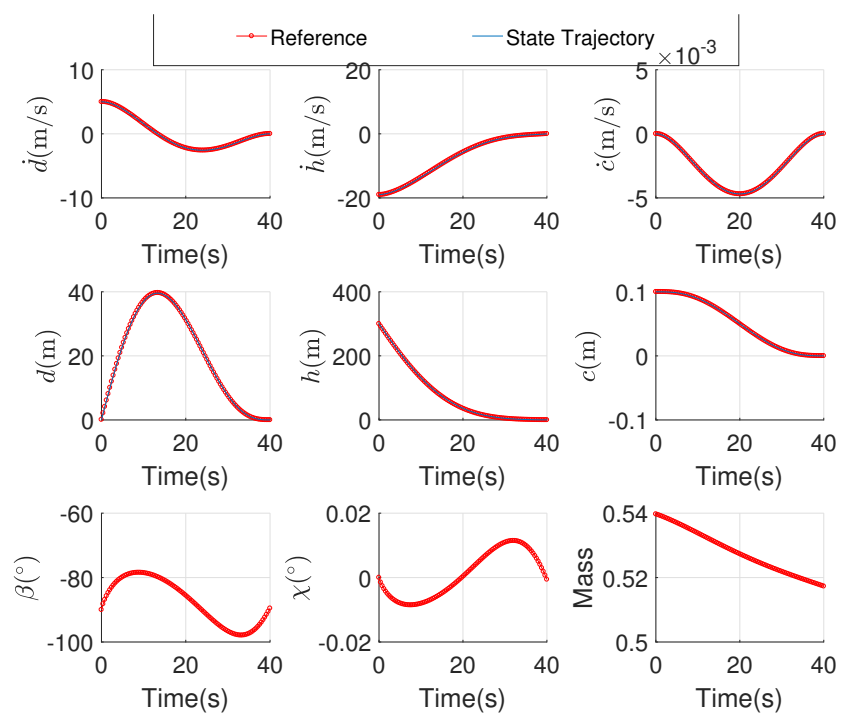
The following gain set in Table 6.1 is used for the tracking controller after the associated values are tuned. Figure 6.9 shows the state trajectories and Figure 6.11 control inputs for the closed loop system in the same sample scenario. The spacecraft tracks the same reference trajectory generated in the previous case. As shown in Figure 6.10, the controller is able to attenuates the discrepancy due to the propellant consumption. Maximum error is experienced in downrange with  $|d_e|_{\max} < 0.15$  m.

**Table 6.1:** Gains for the Lyapunov based PD Controller.

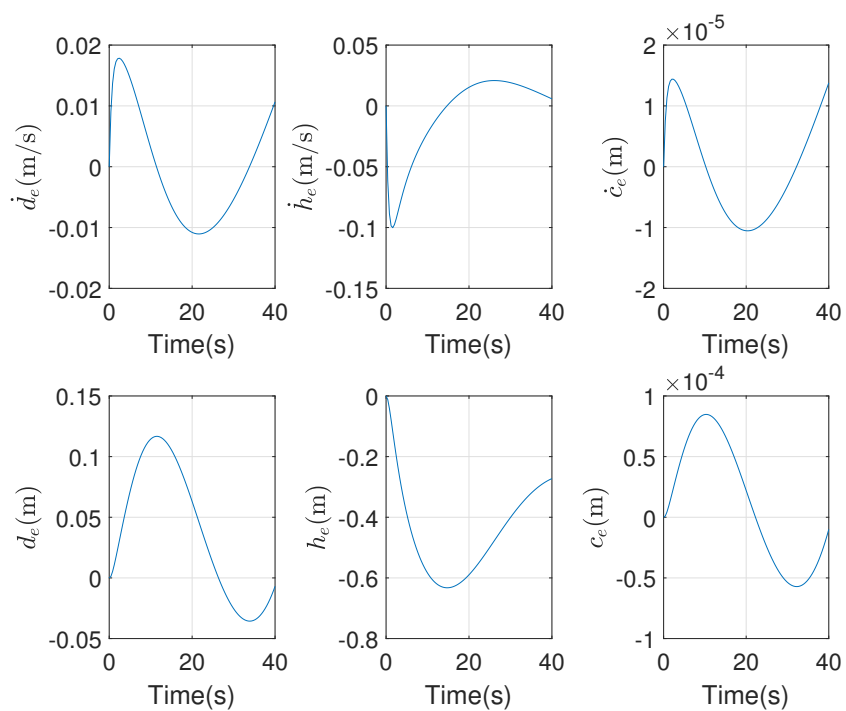
$\alpha_{h_1}$	$\alpha_{d_1}$	$\alpha_{c_1}$	$\alpha_{h_2}$	$\alpha_{d_2}$	$\alpha_{c_2}$	$\alpha_{\beta_h}$	$\alpha_{\beta_d}$	$\alpha_{\beta_c}$	$\alpha_{\gamma_h}$	$\alpha_{\gamma_d}$	$\alpha_{\gamma_c}$	$\alpha_{k_h(t_0)}$	$\alpha_{k_d(t_0)}$	$\alpha_{k_c(t_0)}$
1	1	1	2	2	2	0.25	0.25	0.25	1e-3	1e-3	1e-3	1e-2	1e-2	1e-2

In this approach, the reference trajectories are generated using modified Apollo guidance method by increasing the order of the polynomials to satisfy the initial acceleration. Error due to model inversion is attenuated by Lyapunov based PD controller as illustrated with the simulation results. However, compared to the reference RS computation, this method does not considers final time as an optimization variable. Instead, time of the simulation needs to be defined by previously. In order to compare the results, RS computation with fixed final time is considered for the reference set.

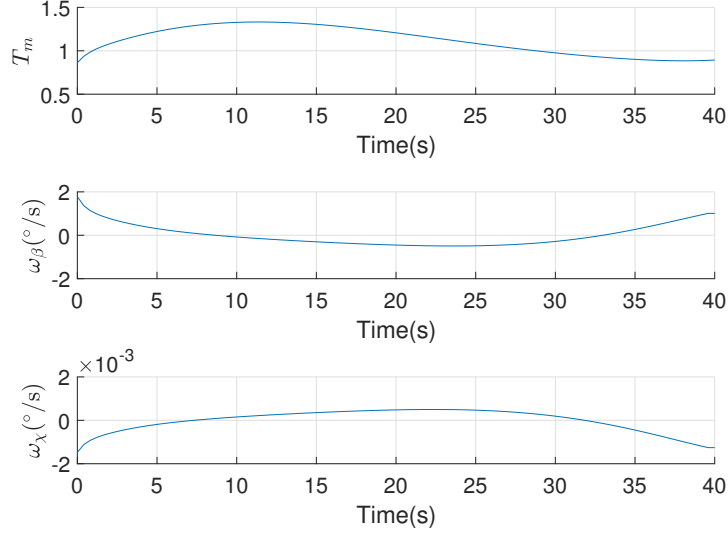
## 6. POLYNOMIAL GUIDANCE



**Figure 6.9:** States with Feedback Controller for Polynomial Reference Trajectory.



**Figure 6.10:** Tracking Error with Feedback Controller for Polynomial Reference Trajectory.



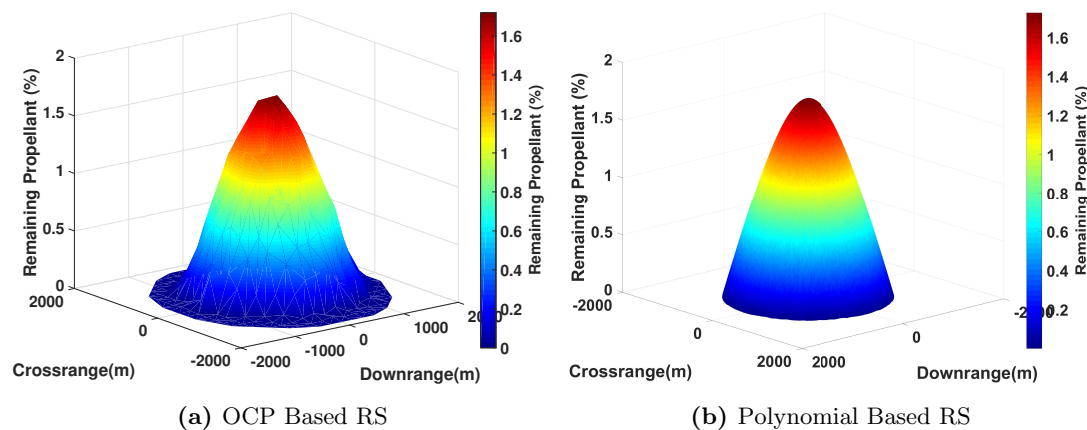
**Figure 6.11:** Controls with Feedback Controller for Polynomial Reference Trajectory.

## 6.5 Comparison of Reachable Sets

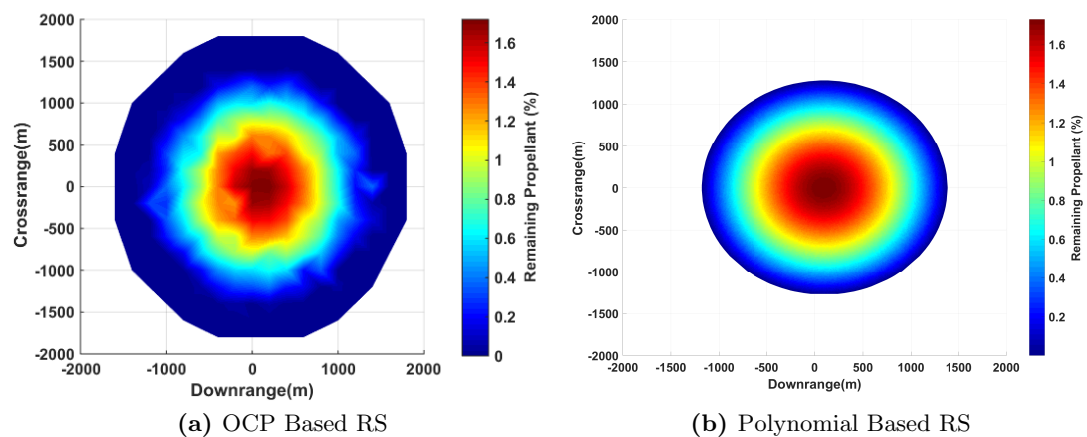
In order to make a comparison between the RS obtained by solving NLP and closed form solution using polynomial guidance and Lyapunov based PD controller, the landing area of the spacecraft is discretized. For the case of OCP based RS, the reference RS is obtained by providing better initial guess for the points at the boundary of the RS. In addition, in order to be consistent with the subsequent sections, the solutions which are not sufficiently close to discretization points on the landing surface are excluded.

The control inputs  $\omega_\beta$  and  $\omega_\chi$  are obtained by taking derivative of the  $\beta$  and  $\chi$  after solving Eq. 6.16 for the polynomial based RS. One disadvantage of the inversion based closed loop system is the active constraints. The solutions in which the control inputs exceed the allowed limits, or saturate are excluded during the RS. Since the method does not rely on solution of a optimization problem, the computational demand is less compared to OCP based RS. Because of the computationally expensive solutions, the target landing region for OCP based RS is discretized with a distance of 250 m between the uniform grid points. For the polynomial based RS the discretization error between the candidate landing region is 10 m. Figure 6.12a-6.12b and Figures 6.13a-6.13b shows the comparison of OCP and polynomial guidance based RS with propellant to reach cost. As shown in the figures, the results for the maximum value of the objective function is close to each other. On the other hand as illustrated in the Figure 6.14, the area that can be reachable according to OCP based solution is larger than polynomial based RS with a ratio of 48.4% in favor of the OCP based algorithm.

## 6. POLYNOMIAL GUIDANCE



**Figure 6.12:** Comparison of OCP and Polynomial Guidance Based RS



**Figure 6.13:** Comparison of OCP and Polynomial Guidance Based RS

The simulation results constitute an example of how the selected guidance and control algorithm changes the size of the maximum attainable area of the lander. Both methods compute the RS with the same initial states, landing scenario, and limits on the control inputs. However, the guidance methods were different in both cases. In the reference RS computation, only an open-loop system is considered without any controller. In the latter scenario, the controller attenuates the model inversion error and tracks the reference trajectory generated by polynomials.

As expected, the cost of computation for the optimal control based RS is more compared to the polynomial based RS, leading to a larger attainable area. Despite the interactive simulation times with the polynomial approach, the computations were not real-time capable. Moreover, the constraints for the polynomial approach are not

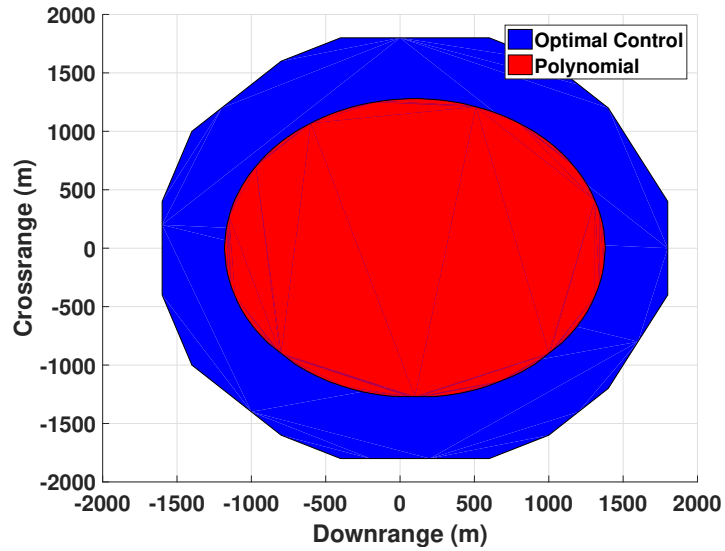


Figure 6.14: Area Comparison for Polynomial and OCP Based RS

considered explicitly during the computations. In other words, the numerical results which does not satisfy the given constraints are omitted in the final RS. On the other hand, the optimal control approach computes the trajectories taking these constraints into account explicitly.

## Part III

# Uncertainties and Parameter Variation

# 7

## Monte Carlo Analysis and Quality Assessment

This chapter contains a description of the Monte Carlo analysis and RS computation with uncertainties. During the mission phase, the on-board sensors are not able to obtain measurements without any error. Moreover, there might be some variations in the system parameters. Therefore, there is a need for robust reachable maps which incorporates these uncertainties and variations. In order to show the change of size and other characteristics of the RS with the uncertainties, MC simulations are run considering different scenarios. The uncertainties for the initial states and the system parameters are included as error sources during the simulations. A Riccati-equation based controller is designed to track the reference trajectories and attenuate the initial errors with a cost of additional propellant consumption.

After the Monte Carlo simulations, the success rate for each corresponding point inside the nonconvex RS is evaluated by counting the number of simulations that satisfies successful landing condition with respect to total number of simulations. The results are illustrated as probability of successful landing projected onto Moon surface.

In order to make a comparison of the results for uncertainties, different metrics are defined using ellipses. Finally, the sensitivity of the final landing region w.r.t. initial state errors are tabulated. It should be noted that, these ellipses are different from the confidence ellipses for testing the robustness of guidance and control algorithms for planetary landing.

### 7.1 Uncertainties

In order to perform MC analysis, the source of errors needs to be identified with reasonable perturbation values. The source of errors could be depending on actuator, sensors, environmental factors or parameters of the dynamical system. For the terminal landing phase of lunar landing, these errors can be generalized but not limited to

- Thruster misalignment, position offset and variation

## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT

---

- Specific impulse,  $I_{sp}$
- Initial dry mass/propellant amount at terminal landing phase
- Measurement error in position and velocity
- Measurement error in attitude of the spacecraft and angular rates
- Dispersion on true CoM location
- Navigational errors (inertia bias, delays)
- Dispersion of the gravitational acceleration on Moon

In order to evaluate performance, the source of errors are considered separately. Reference values from the literature are compared and similar figures are used for perturbation values.

### Position and Velocity Error

It is assumed that an optical navigation system is used for the estimation of position and velocities during terminal landing as illustrated previously in Figure 1.5.  $1-\sigma$  position and velocity error are defined as  $\pm 25$  m and  $\pm 0.4$  m/s at an altitude of 2000 m by Lunghi et. al (2015)<sup>86</sup>. Wibben and Furfaro (2016)<sup>87</sup> considers standard deviation 50 m for the position 0.5 m/s for the velocity error where the landing maneuver starts from an altitude of 11,500 m. In addition to initial uncertainties for the states, navigation errors can also be included in the analysis. It can be assumed that the navigation error decreases linearly with the distance from the surface of the Moon<sup>88</sup>. However, since the initial altitude is relatively low compared to reference values used in the literature, the navigation errors are considered to be negligible and are not included in the MC analysis. Instead, fixed values for the errors is considered and included as uncertainty for the initial states.

### Error for the Angular Rates

In literature, the perturbation values for the flight path angle and crossing angle is  $1.15^\circ$  for standard deviation with a mean value of  $90^\circ$ <sup>89</sup>. However, MC analysis is performed for a lander with an altitude of 3000 m. In this section, the values for the uncertainties on the initial attitude of the vehicle are  $\pm 2^\circ$  deviation from the mean values of  $0^\circ$  for yaw and  $-86.02^\circ$  for the pitch angles, respectively.

### Thrust Related Errors

Thrust related errors could be characterized with misaligning of the thrust, variations in the thrust level and SFC. MC analysis only considers the SFC has an uncertainty margin of  $\pm 2\%$  variation from the nominal value considering the results from literature<sup>90</sup>.



### Error for the Mass of the Vehicle

The error in mass could be characterized in three different sources. The uncertainty in the initial mass of the spacecraft, the uncertainty in the amount of available propellant and the propellant consumed per unit thrust generated. In the MC simulations, the change of total mass of the vehicle is small compared to the reference values from the literature due to the fact that only the terminal landing phase is considered<sup>91</sup>. For the initial mass, the dynamical equations were propagated backwards in time for each specific point inside the RS with an assumption of %5 more propellant consumption. The worst case resulted in approximately %2.95 more propellant requirement. During the MC analysis, the initial uncertainty for the mass is taken  $\pm\%5$  of the nominal value.

In summary, a Gaussian distribution is assumed for the initial state error with mean and standard deviation values given in Table 7.1. The histogram of distribution of initial state uncertainties are illustrated in Figure 7.1 for a sample of 500 simulations.

**Table 7.1:** List of Uncertainties for MC Analysis.

	Mean	$3\sigma$
$\Delta\dot{d}$ (m/s)	5	2
$\Delta\dot{h}$ (m/s)	-19	2
$\Delta\dot{c}$ (m/s)	0	2
$\Delta d$ (m)	0	30
$\Delta h$ (m)	300	30
$\Delta c$ (m)	0	30
$\Delta\beta$ ( $^\circ$ )	-86.02	2
$\Delta\chi$ ( $^\circ$ )	0	2
$\Delta m$	0.5397	%5
$\Delta\sigma_{\text{SFC}}$	5e-4	%2

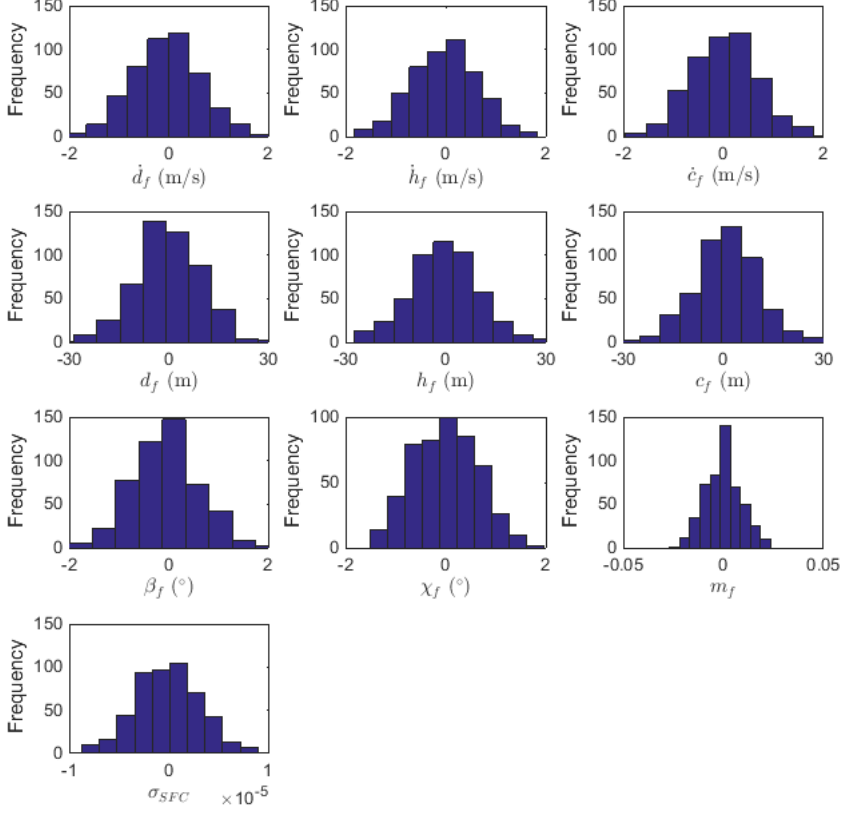
## 7.2 Feedback Controller

After defining the source of errors, a tracking controller is designed for the reference trajectories. In this chapter, different from the previous comparison study, the feedback controller attenuates the initial state error and error due to the system parameter variation. In the previous chapter, the Lyapunov based controller is implemented to attenuate the error due to the model inversion. In this section, the trajectory control of the lunar lander consist of a combination of feedforward and feedback controls as illustrated in Figure 7.2.

For the purpose of the design of the feedback controller the dynamics of the lunar lander are represented by the nonlinear state space model in Eq. (2.1). Assuming that the aforementioned state space representation is analytic and that a real landing trajectory is close to the optimal trajectory  $\mathbf{x}(t) = \hat{\mathbf{x}}(t) + \delta\mathbf{x}(t)$ , the linearization of Eq.

## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT

---



**Figure 7.1:** Histogram of Uncertainties.

(2.1) along  $\hat{\mathbf{x}}(t)$  and  $\hat{\mathbf{u}}(t)$  yields

$$\dot{\delta \mathbf{x}}(t) = \mathbf{A}(t)\delta \mathbf{x}(t) + \mathbf{B}(t)\delta \mathbf{u}(t) \quad (7.1)$$

with  $\mathbf{A}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}, \hat{\mathbf{u}}} \in \mathbb{R}^{n \times n}$  and  $\mathbf{B}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\hat{\mathbf{x}}, \hat{\mathbf{u}}} \in \mathbb{R}^{n \times m}$ ,  $t \in [t_0, t_f]$  and corresponding initial conditions. Considering a time variant control law for the linear time variant dynamical system leads to

$$\begin{aligned} \delta \mathbf{u}(t) &= -\mathbf{K}(t)\delta \mathbf{x}(t) \\ \mathbf{u}(t) &= \hat{\mathbf{u}}(t) + \delta \mathbf{u}(t), \quad t \in [t_0, t_f]. \end{aligned} \quad (7.2)$$

The changes in the dynamics of the lunar lander during a landing maneuver along a reference trajectory are supposed to be slow compared to the sampling rate of the discretized version of the OCP in Eq. (2.10). For this reason the gain matrix of the feedback controller  $\mathbf{K}(t)$  is calculated only at each point  $t_i$ ,  $i \in I$  with  $I = 1, \dots, l$  of the

**Table 7.2:** Gains for the States.

	$\dot{d}$	$\dot{h}$	$\dot{c}$	$d$	$h$	$c$	$\beta$	$\chi$
<b>Q</b>	5	5	5	5	5	5	50	5

**Table 7.3:** Gains for the Control Inputs.

	$T_m$	$\omega_\beta$	$\omega_\chi$
<b>R</b>	1e4	1e4	1e4

control discretization by minimization of the cost function

$$J(\delta \mathbf{x}, \delta \mathbf{u}) = \int_0^{\infty} \delta \mathbf{x}(\tau)^T \mathbf{Q}(t_i) \delta \mathbf{x}(\tau) + \delta \mathbf{u}(\tau)^T \mathbf{R}(t_i) \delta \mathbf{u}(\tau) d\tau \quad (7.3)$$

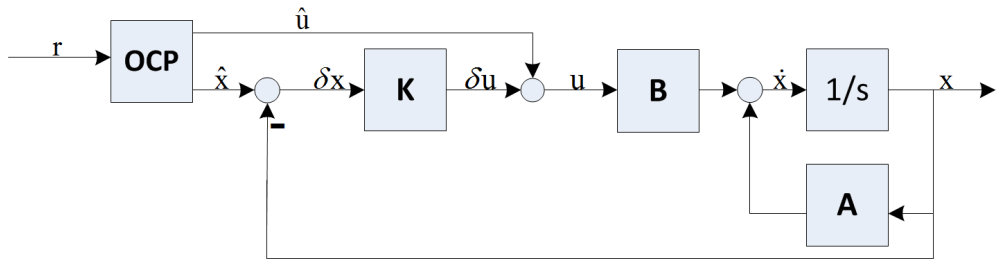
for all  $i \in I$ . Assuming the stability of the underlying system, and positive semi-definite  $\mathbf{Q}(t_i)$  and  $\mathbf{R}(t_i)$  for all  $i \in I$ , the feedback matrix that solves the OCP is given as a function

$$\mathbf{K}(t_i) = \mathbf{R}(t_i)^{-1} \mathbf{B}(t_i)^T \mathbf{S}(t_i), \quad i \in I \quad (7.4)$$

of the unique stabilizing solution  $\mathbf{S}(t_i)$  of the algebraic Riccati equation

$$\mathbf{A}(t_i)^T \mathbf{S}(t_i) + \mathbf{S}(t_i) \mathbf{A}(t_i) - \mathbf{S}(t_i) \mathbf{B}(t_i) \mathbf{R}(t_i)^{-1} \mathbf{B}(t_i)^T \mathbf{S}(t_i) + \mathbf{Q}(t_i) = 0, \quad i \in I \quad (7.5)$$

such that the eigenvalues of  $\mathbf{A}(t_i) - \mathbf{B}(t_i) \mathbf{K}(t_i)$ ,  $i \in I$  are negative. A linear interpolation has been applied to the  $\mathbf{K}(t_i)$ ,  $i \in I$  for time-continuous control law.



**Figure 7.2:** Block Diagram of the Closed Loop System.

The main goal for the feedback control is to achieve the final landing point by following the reference trajectory while attenuating initial state errors due to uncertainties. The diagonal weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$  are kept constant during the simulations for each reference trajectories. The final time of the reference trajectories are 40 s and the number of controllers are  $i = 10$  for the duration of the maneuver.

During the MC simulations, controller gains defined in Table 7.2 and Table 7.3 are used for the MC simulations.

## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT

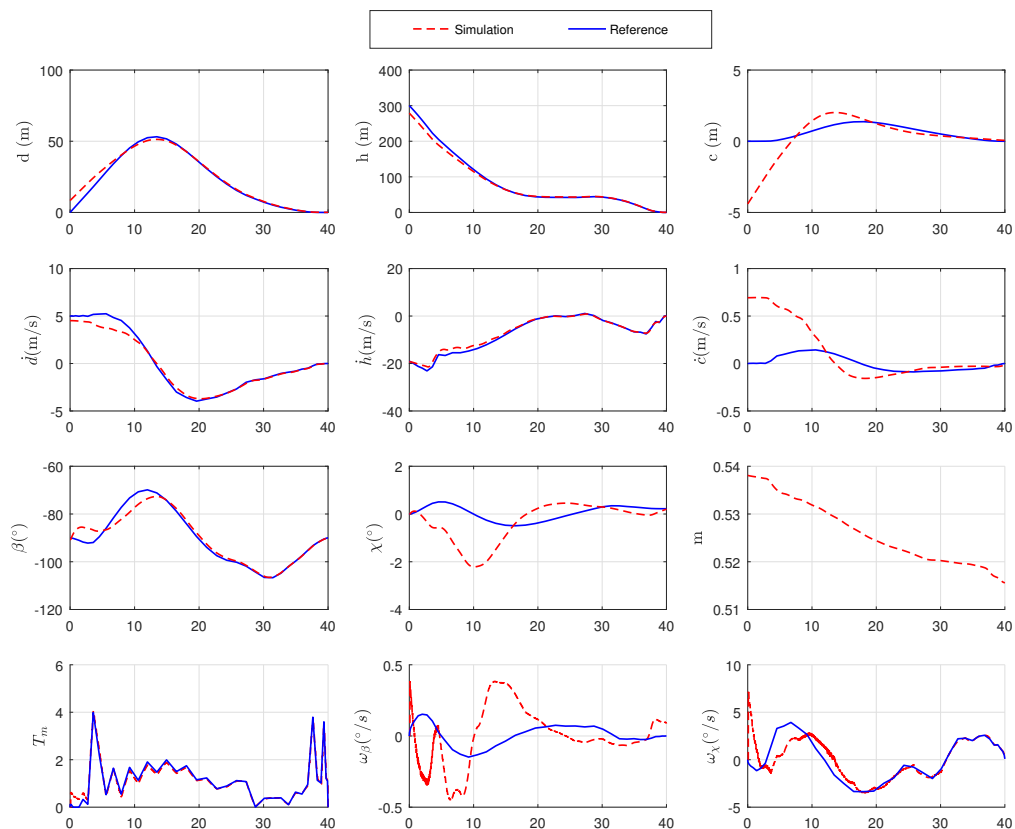


Figure 7.3: Simulation for the Tracking Problem.

Figure 7.3 shows the simulation of the closed loop system for a specific point inside the RS with initial state error. Similarly, Figure 7.4 shows the discrepancy between the reference trajectory and the simulation. It can be seen from the plots that the controller is able to track the given reference trajectory by attenuating the initial state error.

### 7.3 Monte Carlo Simulations

The uncertainties for the initial state and the specific fuel consumption are included as error sources during the simulations. Table 7.1 shows the range of the error for each state and specific fuel consumption. The tracking controller tries to follow the reference trajectories that are obtained in the reference RS. The cost of attenuating the disturbance and uncertainty is the amount of extra propellant consumed by the main engine. After the Monte Carlo simulations, the results that satisfy the safe landing conditions are filtered. The success rate for each corresponding point inside the RS is evaluated by the number of simulations that achieves a successful landing with respect

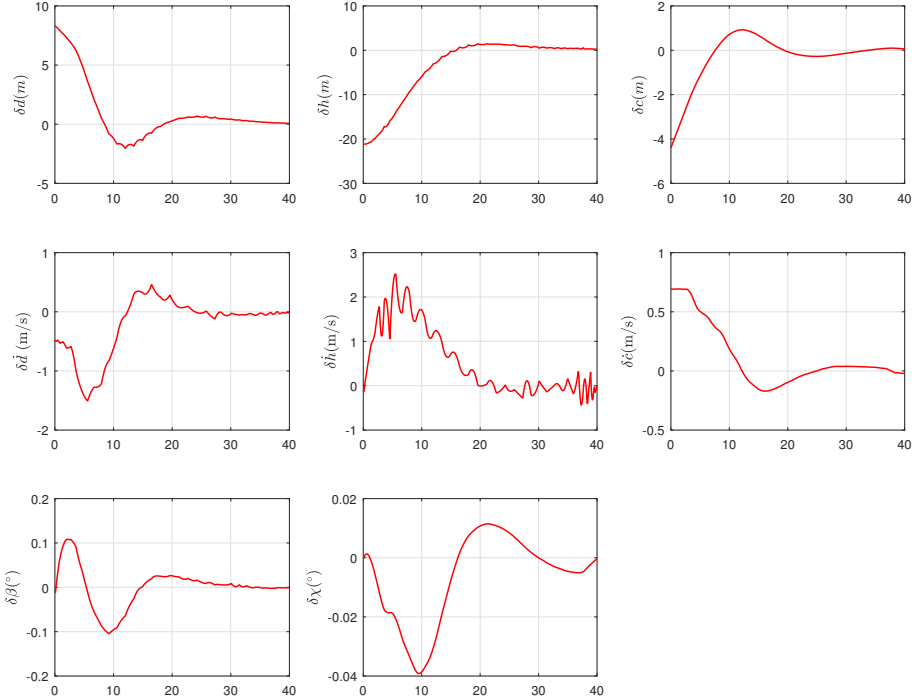


Figure 7.4: State Error Between the Reference Trajectory and the Simulation.

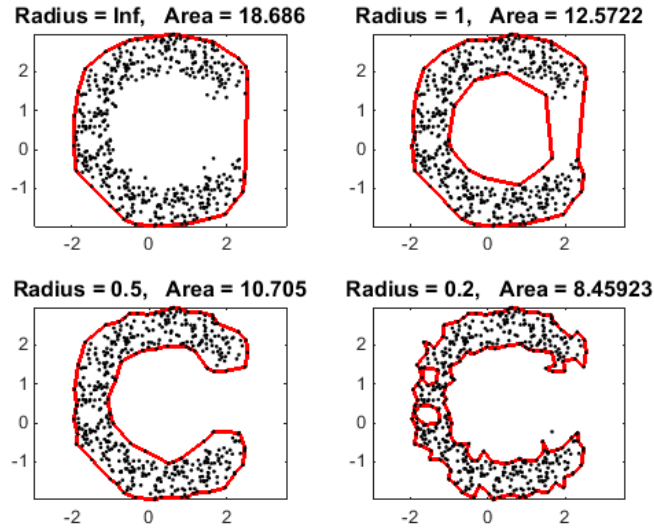
to total number of simulations. The region which the lander achieve a safe landing for all MC simulations is called the 100% successful attainable landing area of the lunar lander.

The borderlines that define the nonconvex hull of 100% successful attainable landing area are obtained using alpha shapes. Alpha shapes are a family of piecewise linear simple curves associated with the shape of a finite point set. The nonconvex region is generated by point pairs that can be touched by an empty disc of radius alpha. In Figure 7.5, the nonconvex hull of point set is obtained using alpha shapes with different radius. In this case, the point set represents the discrete landing points inside 100% successful attainable landing area.

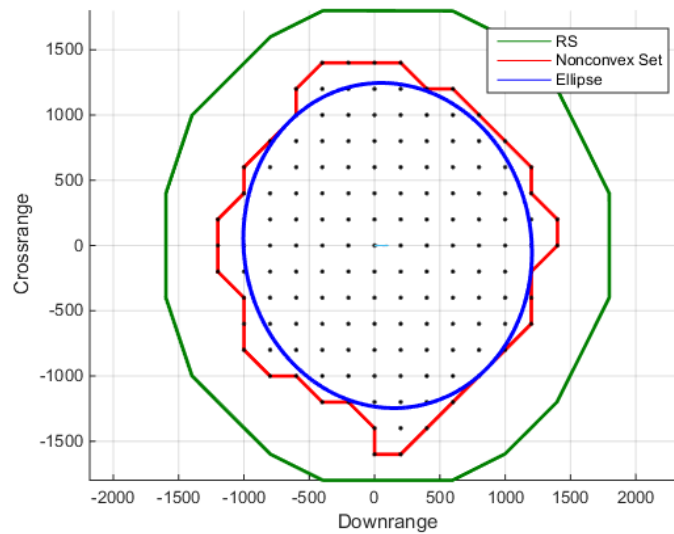
After the nonconvex region is defined, an ellipse with the maximum area is fitted inside the nonconvex set to evaluate the ratios of the reachable area, the successful landing area and area of the ellipse. Figure 7.6 shows the comparison of the reference reachable landing area without the controller, with the controller and the ellipse that fits inside the nonconvex successful landing area.

## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT

---



**Figure 7.5:** Representation of the Nonconvex Region by Alpha Shapes.



**Figure 7.6:** Representation of the Maximum Area Ellipse.

### Probabilistic Attainable Maps

Probabilistic attainable areas are obtained for each uncertainty source as illustrated with Figures 7.7-7.16. The successful landing area including initial errors for all states is illustrated in Figure 7.17.

Table 7.4 shows the results of the MC analysis as well as center, semi-major and

### 7.3 Monte Carlo Simulations

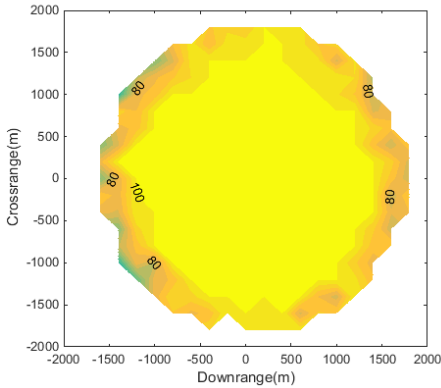


Figure 7.7: RS with Uncertainty in  $\dot{d}_0$ .

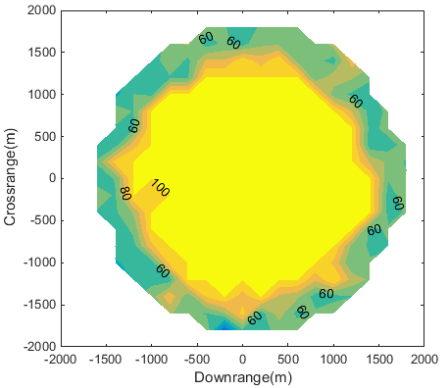


Figure 7.8: RS with Uncertainty in  $\dot{h}_0$ .

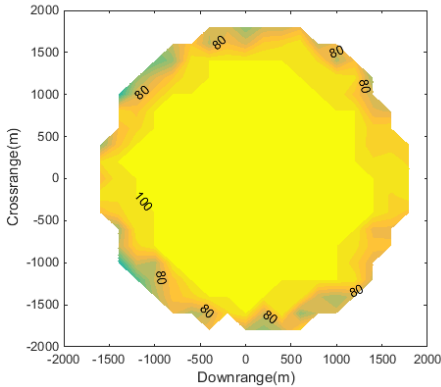


Figure 7.9: RS with Uncertainty in  $\dot{c}_0$ .

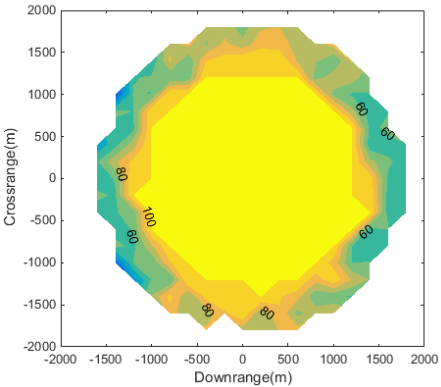


Figure 7.10: RS with Uncertainty in  $d_0$ .

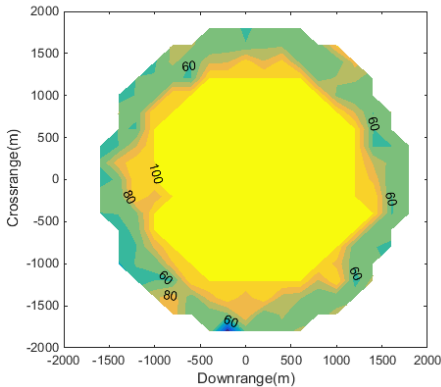


Figure 7.11: RS with Uncertainty in  $h_0$ .

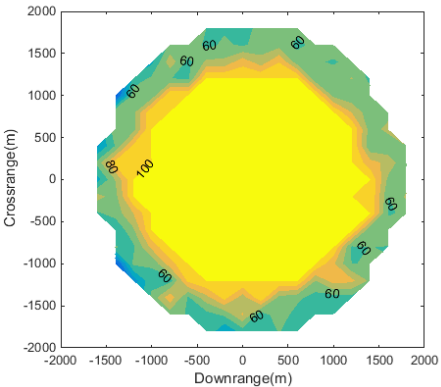
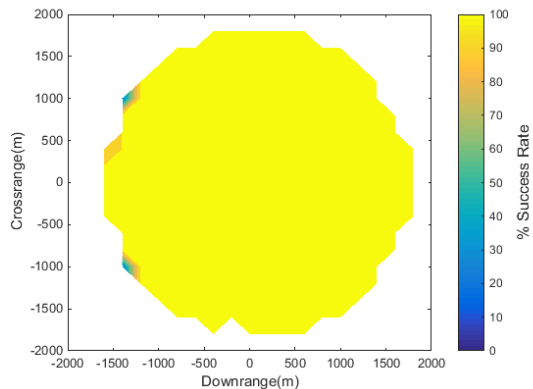
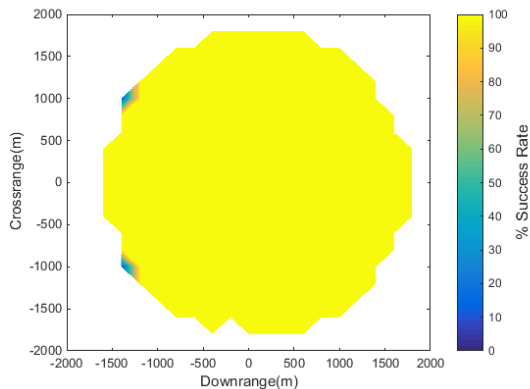


Figure 7.12: RS with Uncertainty in  $c_0$ .

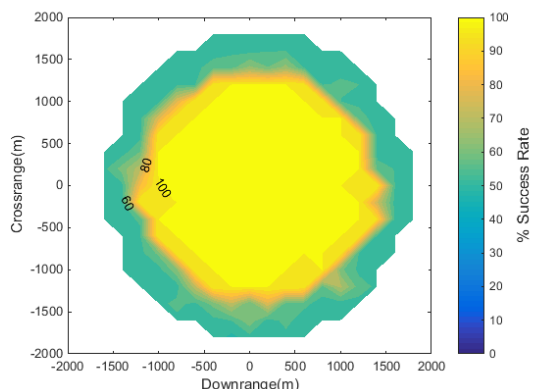
## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT



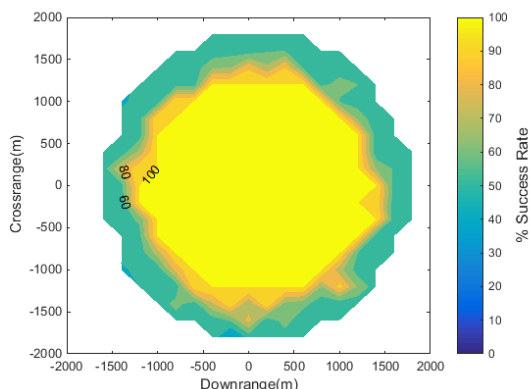
**Figure 7.13:** RS with Uncertainty in  $\beta_0$ .



**Figure 7.14:** RS with Uncertainty in  $\chi_0$ .



**Figure 7.15:** RS with Uncertainty in  $m_0$ .



**Figure 7.16:** RS with Uncertainty in SFC.

semi-minor axis and orientation of the ellipse that fits inside the nonconvex 100% successful attainable landing area. For each specific case, 500 Monte Carlo simulations are performed using a fixed structure of the feedback controller with the same gains. The borderlines that define the nonconvex hull of the successful landing area are obtained using alpha shapes. Finally, an ellipse with the maximum area is fitted inside the nonconvex set to evaluate the ratios of the reachable area, the successful landing area and the area of the ellipse. The parameters of the ellipse are discretized, and the maximum area is computed by a brute force method. In order to obtain the ellipse with the maximum area, the area of each ellipse using discrete values are calculated. Additional Figures for the 100% successful attainable landing area are included in Appendix B.

It is seen that, when uncertainties are introduced the 100% attainable landing area gets smaller due to the amount of extra propellant consumed to stabilize the plant. The change of the successful landing area is most sensitive to the mass of the spacecraft or the amount of propellant considering the area ratios. As the uncertainty increases for



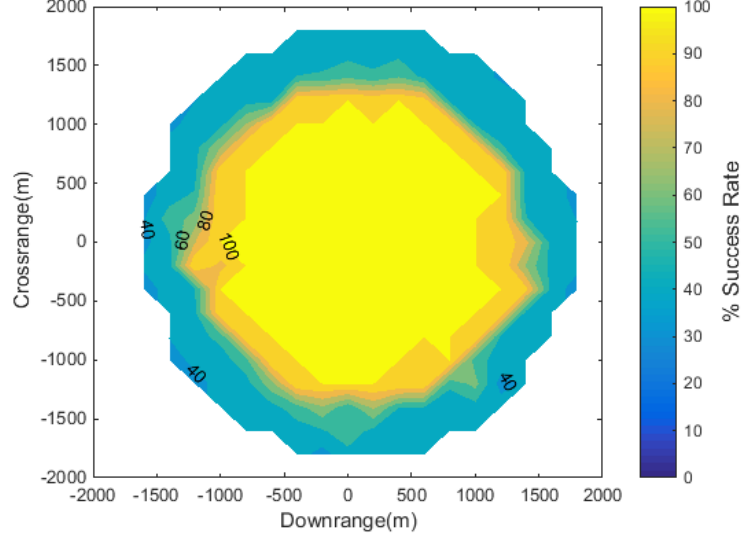


Figure 7.17: RS with Uncertainty in All States.

Table 7.4: Results of the MC Analysis and Properties of the Ellipse.

	Center	Axis	Orientation	Area Ratio	Ellipse Ratio
$\Delta \dot{d}$ (m/s)	[100 -50]	[1100 1350]	90	57.09	47.05
$\Delta \dot{h}$ (m/s)	[200 0]	[1000 1200]	0	48.20	38.02
$\Delta \dot{c}$ (m/s)	[100 0]	[1100 1250]	0	54.87	43.57
$\Delta d$ (m)	[100 0]	[1100 1200]	-10	45.98	41.82
$\Delta h$ (m)	[200 0]	[1000 1200]	0	45.59	38.02
$\Delta c$ (m)	[100 0]	[1100 1200]	0	46.79	41.81
$\Delta \beta$ (°)	[100 50]	[1500 1700]	0	95.62	80.81
$\Delta \chi$ (°)	[100 0]	[1600 1600]	10	96.03	81.12
$\Delta m$	<b>[100 100]</b>	<b>[900 1050]</b>	<b>0</b>	<b>38.50</b>	<b>29.92</b>
$\Delta \sigma_{\text{SFC}}$	[100 0]	[1100 1200]	0	46.79	41.82
Total	[100 0]	[900 950]	0	33.84	27.06

the initial mass, the points along the boundary of the reachable set is not attainable by the lander during Monte Carlo simulations. Although the tracking controller tracks the reference trajectory, the successful landing condition is not satisfied in some cases. Specific fuel consumption is yet another important parameter affecting the area of the success region. For the cases, where the main engine consumes more propellant per unit thrust, the available propellant is not enough to steer the vehicle to the desired landing point.

As the initial error for the velocity along downrange is introduced, the successful landing region becomes smaller along crossrange, leading to a slender successful landing

## 7. MONTE CARLO ANALYSIS AND QUALITY ASSESSMENT

---

region as illustrated in Figure 7.7. Similarly, the initial error for the velocity along crossrange results in a smaller successful landing area, with unattainable regions close to the maximum and minimum crossrange of the reachable set in Figure 7.9. The initial error for the rate of altitude has a symmetric behavior in crossrange and downrange coordinates as shown in Figure 7.8. Although initial position error decreases the success rate of the landing, similar attainable landing maps are obtained in Figures 7.10 and 7.12. The results for the position error is also verified from the properties of the ellipses inside the nonconvex 100% successful landing region in Table 7.4.

On the other hand, the initial error for the attitude of the spacecraft is not very critical in Figure 7.11. The success rate of landing decreases slightly for some points on the boundary of the reachable set. However, similar to the discussion for the properties of the ellipses, the mass and the specific fuel consumption of the vehicle is decreasing the success rate of landing considerably in Figure 7.15 and Figure 7.16. Finally, for the cases in with uncertainty in all states, the success rate of the landing decreases substantially with a symmetrical behaviour as illustrated in Figure 7.17.

Successful landing is less sensitive to the initial errors for the attitude of the lander. The ratio of 100% successful attainable area for pitch angle and reference attainable area without initial state error is 95.62% as illustrated in Figure 7.13. Similarly, the ratio of 100% successful attainable area for yaw angle and reference attainable area without initial state error is 96.03% as illustrated in Figure 7.14. On the other hand, the lander is most sensitive to changes of initial mass for a successful landing. The ratio of the attainable area with respect to successful attainable area is 38.50%. As expected the smallest area is obtained when the uncertainties for all states are introduced during MC simulations. Similarly area ratios for reference RS and 100% successful attainable area is 33.84%.

The orientation of the ellipse does not change for all the states except for the initial error for downrange velocity. In this case the ellipse is elongated across crossrange direction. It can be concluded that if the uncertainty in the initial downrange velocity increases, the limits of the 100% successful attainable landing area in terms of downrange get approach to the center of the ellipse.

The MC analysis could be used see the effect of controller parameters on the size of the successful attainable landing area within the presence of uncertainties and disturbances. The parameters could be optimized to maximize the 100% successful attainable landing area or increase the likelihood of success of the mission for specific uncertainties.

## 8

# Reachable Sets using Unscented Transform

In the previous chapter, RS with the uncertainty in the initial states are computed. The quality metrics shows the sensitivity of the uncertainty for the final RS.

In this chapter, unscented transform is used to obtain probabilistic RS incorporating the uncertainties and parameter variations. The unscented transformation is a novel method for predicting means and covariances for nonlinear systems<sup>92</sup>. The basic principle of unscented transform is that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function. A set of points named sigma ( $\sigma$ ) points are chosen for a given sample mean and covariance. These points undergo a nonlinear transformation, resulting in a cloud of transformed points as illustrated in Figure 8.1. The mean and the covariances are approximated using the transformed points. This method captures the first two moments of the Taylor Series expansion of the covariance for Gaussian inputs.

Although the method resembles to Monte Carlo methods, the fundamental difference is the methodology for selecting sample points. Different from Monte Carlo methods, unscented transform defines the samples according to a specific, deterministic algorithm. This in return reduces the number of simulations required to capture the behavior of the RS within the presence of uncertainties. Therefore, unscented transform obtains similar results like MC analysis using less number of simulations.

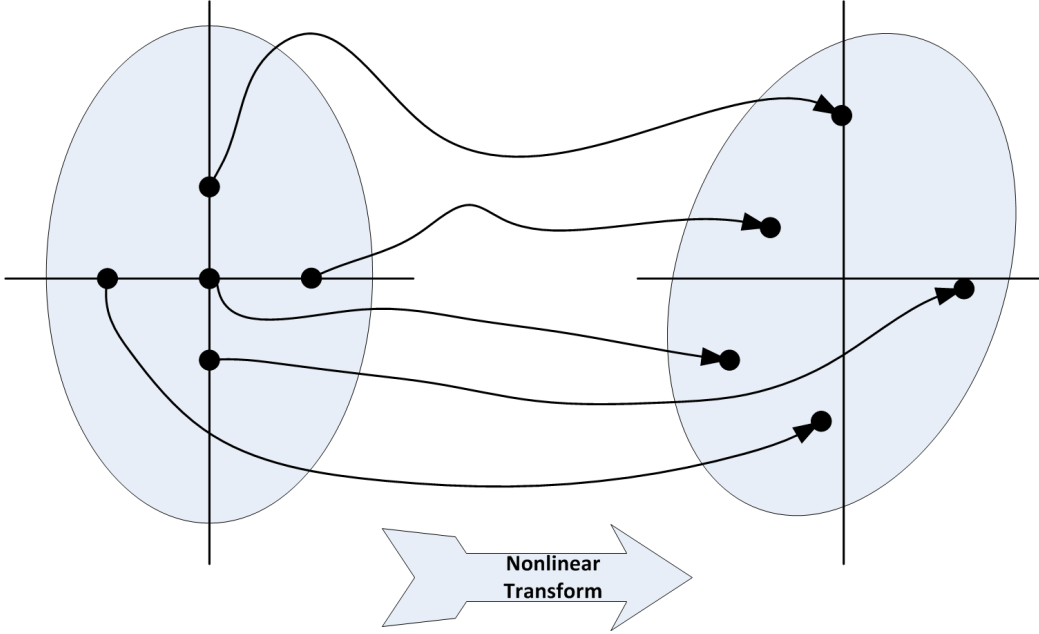
## 8.1 Unscented Transform

Consider  $\mathbf{x}$  be an  $n$ -dimensional random variable with mean  $\bar{\mathbf{x}}$  and covariance  $\mathbf{P}_{xx}$ . The nonlinear transformation of  $\mathbf{x}$  gives a second random variable  $\mathbf{y}$ , with the following relation

$$\mathbf{y} = g(\mathbf{x}) \tag{8.1}$$

The aim is to obtain the mean  $\bar{\mathbf{y}}$  and covariance  $\mathbf{P}_{yy}$  of  $\mathbf{y}$  after the transformation.

## 8. REACHABLE SETS USING UNSCENTED TRANSFORM



**Figure 8.1:** The Concept of Unscented Transform Using  $\sigma$  Points.

First step is to approximate  $\bar{\mathbf{x}}$  and  $\mathbf{P}_{xx}$  using  $\sigma$  points. The unscaled unscented transform uses a minimum of  $n + 1$  points to match the mean and the covariance of  $n$ -dimensional variable. The unscaled transformation has two disadvantages. As the number of dimension increases, the radius of sphere which encloses the  $\sigma$  points increases. In addition, the calculated points are asymmetrically distributed about the origin, leading to high order effects such as the skew more significant.

A scaling method is used to overcome dimensional effects by calculating the transformation of a scaled set of  $\sigma$  points<sup>93</sup>. In this method, first a matrix  $\mathcal{X}$  of  $2n + 1$  *sigma* vectors  $\mathcal{X}_i$  is constructed using

$$\begin{aligned}\mathcal{X}_0 &= \bar{\mathbf{x}} \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left( \sqrt{(n + \lambda)\mathbf{P}_{xx}} \right)_i \quad i = 1, \dots, n \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left( \sqrt{(n + \lambda)\mathbf{P}_{xx}} \right)_{i-n} \quad i = n + 1, \dots, 2n\end{aligned}\tag{8.2}$$

and the associated weights  $\mathcal{W}_i$  are obtained with

$$\begin{aligned}\mathcal{W}_0^{(1)} &= \lambda / (n + \lambda) \\ \mathcal{W}_0^{(2)} &= \lambda / (n + \lambda) + (1 - \alpha^2 + \beta) \\ \mathcal{W}_i^{(1)} &= \mathcal{W}_i^{(2)} = 1 / (2(n + \lambda)) \quad i = 1, \dots, 2n\end{aligned}\tag{8.3}$$

where  $\lambda = \alpha^2(n + \kappa) - n$  is a scaling parameter. The spread of  $\sigma$  points around  $\mathbf{x}$  are determined by positive  $\alpha \in (0, 1]$ .  $\kappa$  is the non-negative secondary scaling parameter

---

## 8.2 Approximation of Multivariate Probability Density Function

and is usually set to 0. The prior knowledge of the distribution of  $\mathbf{x}$  is incorporated by  $\beta$ . For Gaussian distribution, the error in higher order terms in the Taylor Series expansion of the covariance is minimized for  $\beta = 2$ .

The term  $\left(\sqrt{(n + \lambda)\mathbf{P}_{xx}}\right)_i$  is the  $i^{\text{th}}$  row of the matrix square root and is computed via diagonalization. The matrix square root can be also obtained by numerically efficient and stable methods such as Cholesky decomposition.

Finally,  $\sigma$  vectors  $\mathcal{X}_i$  are propagated with the following nonlinear transformation

$$\mathcal{Y}_i = g(\mathcal{X}_i) \quad i = 0, \dots, 2n \quad (8.4)$$

A weighted sample mean and covariance of the *sigma* points are used to approximate the statistics of  $\mathbf{y}$  using

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2n} \mathcal{W}_i^{(1)} \mathcal{Y}_i \quad (8.5)$$

$$\mathbf{P}_{yy} \approx \sum_{i=0}^{2n} \mathcal{W}_i^{(2)} (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T. \quad (8.6)$$

Thus, after a given mean  $\bar{\mathbf{x}}$  and covariance  $\mathbf{P}_{xx}$  undergoes a nonlinear transformation, the resulting mean  $\bar{\mathbf{y}}$  and covariance  $\mathbf{P}_{yy}$  can be approximated. The mean and covariance of  $\mathbf{x}$  are captured up to the second order. This means that the approximated mean and covariance of  $\mathbf{y}$  are also calculated up to second order. The method could be further exploited to include information from higher orders, leading to better accuracy.

The resulting  $\sigma$  points for different parameters  $\kappa$  and  $\alpha$  are given in Figures 8.2-8.3 together with  $1\sigma$  contour plot. The increase in these tuning parameters results in  $\sigma$  point set to diverge from the mean.

Figure 8.4 shows the histogram of final altitude for a sample scenario obtained by Monte Carlo simulations. In these simulations, the tuning parameters are taken as  $\kappa = 0.1$  and  $\alpha = 1$ . The simulations are performed for the closed loop system with an uncertainty on the initial altitude. The dispersion of the final altitude is approximated where  $\sigma$  points undergo the same nonlinear transformation as in the case of MC simulations. Initial  $\sigma$  point set is defined by proper selection of  $\kappa$  and  $\alpha$  in Eq.(8.3). These points are selected to be on the 1-2-3  $\sigma$  confidence ellipses of the initial altitude. The results shows that when the initial  $\sigma$  points are close to the mean, the approximation of the final dispersion is better as expected.

## 8.2 Approximation of Multivariate Probability Density Function

Univariate probability distribution is described for a single random variable with mean and variance. The multivariate normal distribution in  $n$  dimensions, on the other hand is characterized by two sets of parameters, namely a mean vector of  $\bar{\mathbf{y}}$ , and a covariance

## 8. REACHABLE SETS USING UNSCENTED TRANSFORM

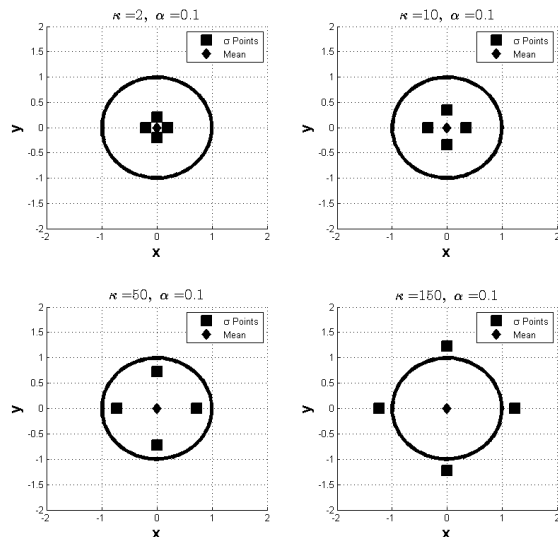


Figure 8.2: Variation of  $\kappa$  for Determining  $\sigma$  Points.

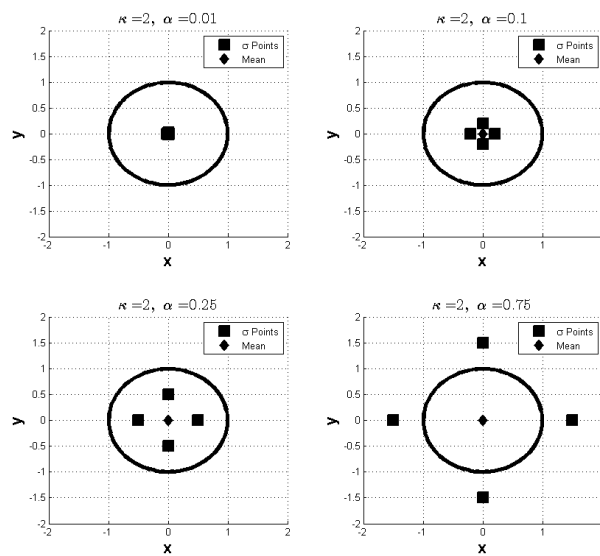
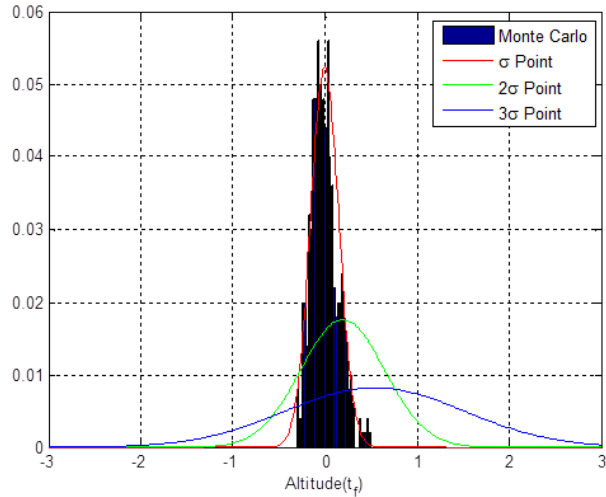


Figure 8.3: Variation of  $\alpha$  for Determining  $\sigma$ -Points.

matrix  $\mathbf{P}_{yy}$ . The probability density function for multivariate normal distribution is expressed with

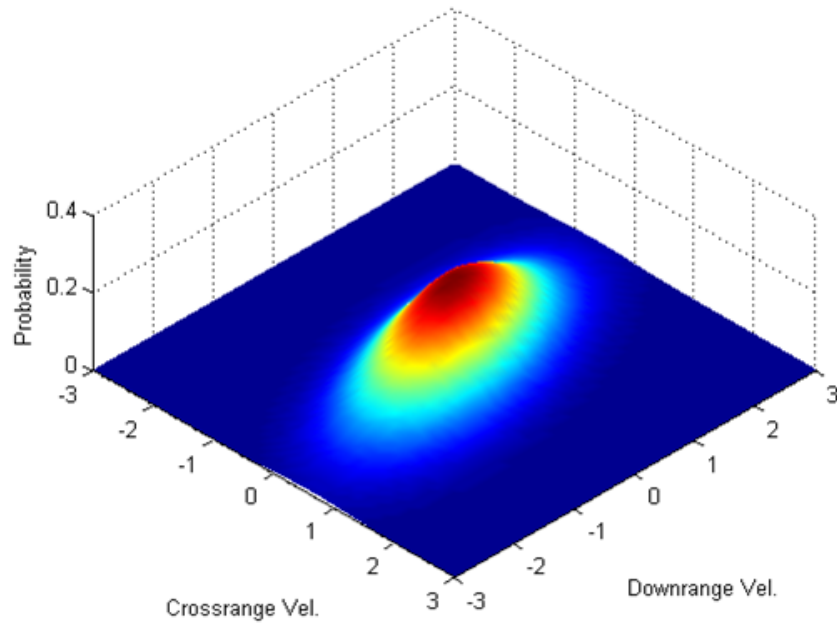
$$p(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{P}_{yy}|}} \exp \left[ -\frac{(\mathbf{y} - \bar{\mathbf{y}})^T \mathbf{P}_{yy}^{-1} (\mathbf{y} - \bar{\mathbf{y}})}{2} \right] \quad (8.7)$$

## 8.2 Approximation of Multivariate Probability Density Function



**Figure 8.4:** Variation of Probability Density Function with  $\sigma$ -Points.

Figure 8.5 shows the multivariate probability distribution of final landing velocities in downrange and crossrange coordinates approximated by using  $\sigma$  points as described previously and using the relation in Eq. (8.7).



**Figure 8.5:** Probability Distribution of Final Velocities in Downrange and Crossrange.

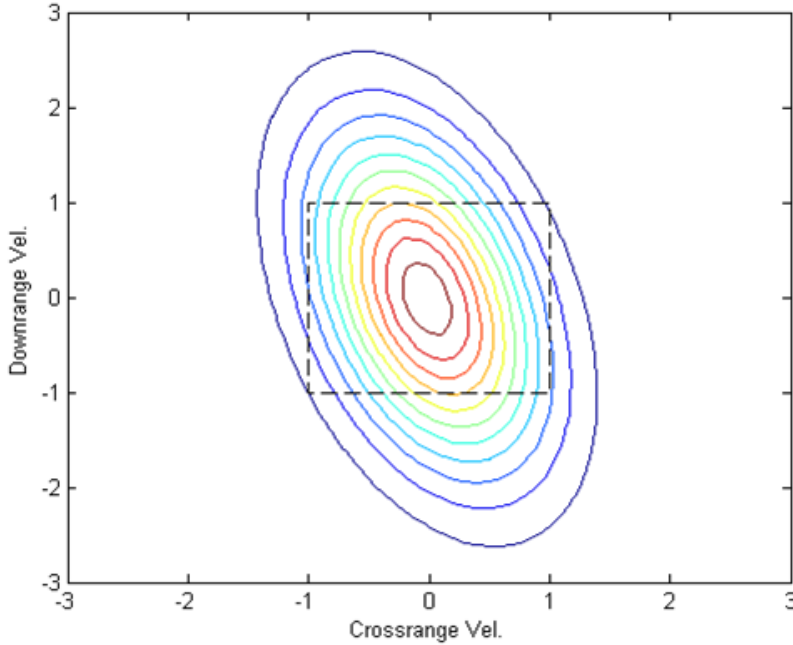
## 8. REACHABLE SETS USING UNSCENTED TRANSFORM

---

Similarly the multivariate normal distribution function is expressed by

$$\Phi(\mathbf{a}, \mathbf{b}, \mathbf{P}_{yy}) = \frac{1}{\sqrt{|\mathbf{P}_{yy}|}(2\pi)^n} \int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_n}^{b_n} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{P}_{yy}^{-1} \mathbf{y}\right) d\mathbf{y} \quad (8.8)$$

where  $-\infty \leq a_i < b_i \leq \infty$  for all  $i$ , and  $\mathbf{P}_{yy}$  is positive semi-definite symmetric  $n \times n$  matrix. Several suitable integration routines can be applied to compute cumulative probability over regions in Eq. (8.8)<sup>94</sup>. After the proper bounds are defined for the region which satisfy safe landing conditions as illustrated in Figure 8.6, the probability of a safe landing can be computed with the Algorithm 1.



**Figure 8.6:** Safe Landing Constraints of Final Velocities in Downrange and Crossrange.

Figures 8.7-8.8-8.9 show the results obtained by unscented transform based probability maps, when the initial  $\sigma$  points are selected to be on 1-2-3  $\sigma$  confidence ellipses for all states. The selection of the initial  $\sigma$  point set has an effect on the quality of the approximation for probability of safe landing. As the dispersion increases, some of the points inside the RS does not satisfy safe landing conditions, leading to lower chance of probability for these points.

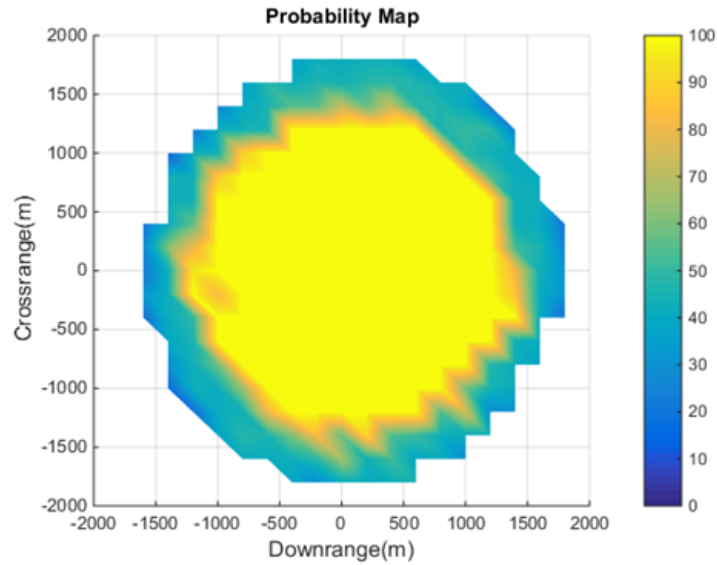
In order to compare with the results from MC analysis, the maps which shows the probability of successful landing is also computed using the same uncertainties in Chapter 7. The simulations for unscented transform only considers the uncertainties for the initial states and parameters. The number of elements for  $\mathbf{x}$  is 10, and the required number of simulations for each point inside the RS is 21, whereas for MC



## 8.2 Approximation of Multivariate Probability Density Function

### Algorithm 1 Reachable Sets using Unscented Transform

- 1: **for**  $i \leq n_d$  **do**
- 2:     **for**  $j \leq n_c$  **do**
- 3:         Calculate  $\mathcal{X}_i$  and associated weights  $\mathcal{W}_i$  using 8.2 and 8.3
- 4:         Propagate  $\mathcal{X}_i$  with nonlinear transformation  $g(\cdot)$
- 5:         Approximate resulting covariance  $\mathbf{P}_{yy}$  and mean  $\bar{\mathbf{y}}$  using relations 8.5 and 8.6
- 6:         Construct probability distribution for current point  $g_{ij}$  inside RS with 8.7
- 7:         Compute the probability of the region defined by the bounds  $\mathbf{a-b}$  and expression in 8.8
- 8:     **end for**
- 9: **end for**
- 10: Draw contour plots inside the RS with the calculated probability of successful landing



**Figure 8.7:** Probability Map with Unscented Transform ( $1\sigma$ )

500 simulations are performed. The list of uncertainties for the states with mean and standard deviations are listed in Table 7.1. The  $\sigma$  points undergo the same nonlinear transformation as in the case of MC analysis.

Figures 8.10a-8.10b compares the result for successful landing obtained by MC analysis and using unscented transform. In both scenarios, in order to eliminate numerical deficiencies, the final altitude for safe landing is considered as  $(-1 < h_f < 1$  m). Although there is a slight difference for the probability maps, considering the computation requirement for unscented transform, the results could be considered as satisfactory. Moreover, the unscented transform could be used as an initial estimating

## 8. REACHABLE SETS USING UNSCENTED TRANSFORM

---

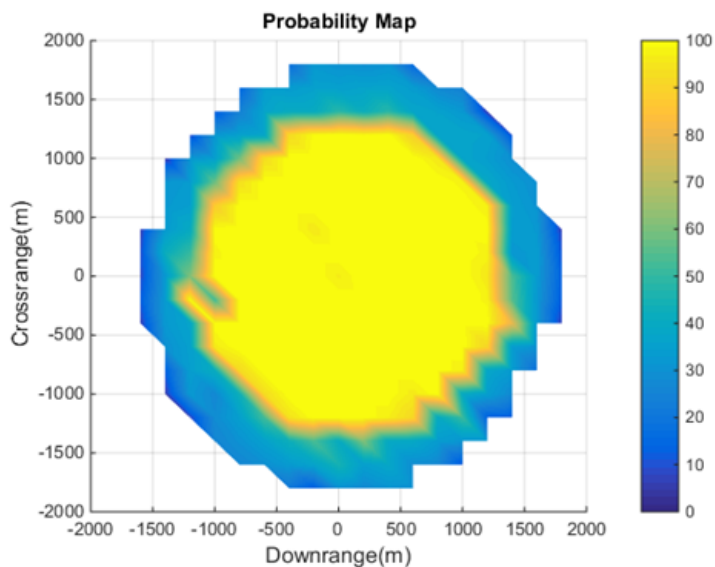


Figure 8.8: Probability Map with Unscented Transform ( $2\sigma$ )

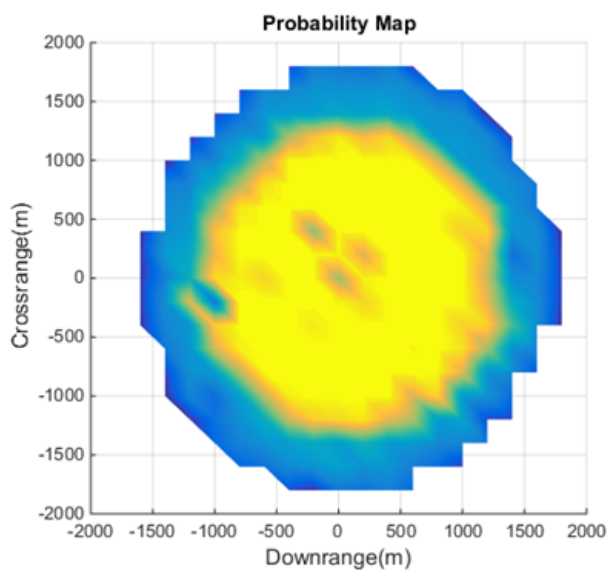
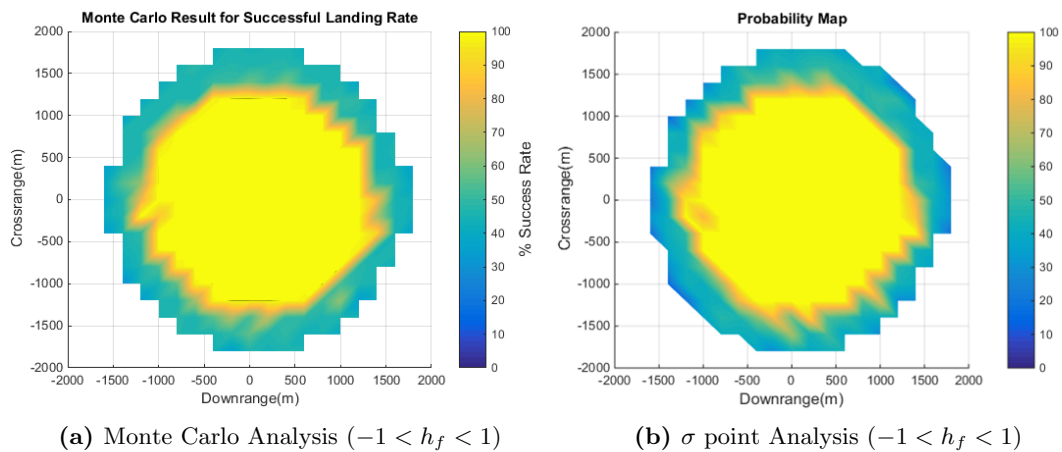


Figure 8.9: Probability Map with Unscented Transform ( $3\sigma$ )

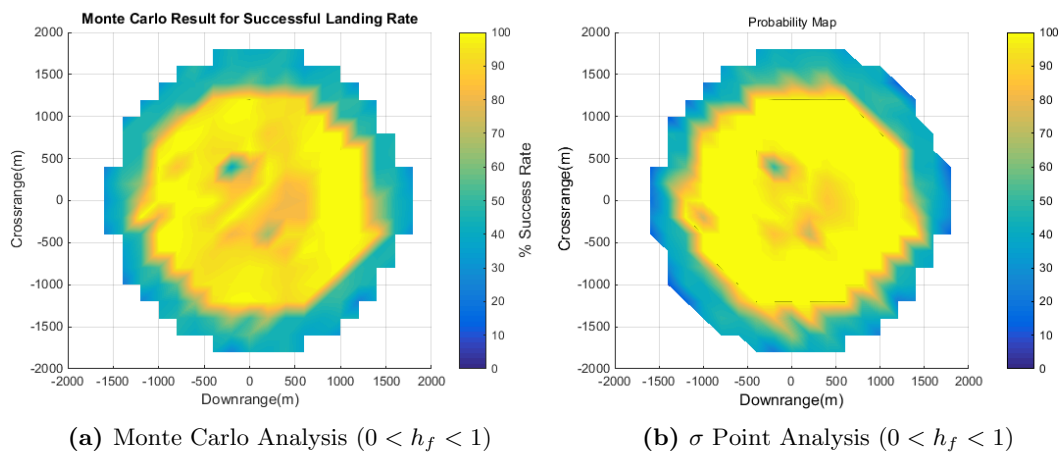
tool for the nonlinear dynamical system with uncertainties, where running simulations are computationally demanding.

Finally, the results are illustrated in Figure 8.11a and Figure 8.11b for the case when the final altitude for safe landing is considered as ( $0 < h_f < 1$  m). In this case, it is observed that, some of the points inside the RS is not reachable during both for MC analysis and maps obtained by unscented transform. The histogram of final states

## 8.2 Approximation of Multivariate Probability Density Function



**Figure 8.10:** Comparison of MC and  $\sigma$  Point Analysis



**Figure 8.11:** Comparison of MC and  $\sigma$  Point Analysis

shows that, the safe landing condition eliminates some of the results because of the hard constraint for final altitude. If this condition is relaxed slightly, more reasonable results are obtained.

## 9

# Set Interpolation

This section introduces the method for interpolating optimal trajectories with different initial conditions or with different system parameters. The main objective is to evaluate interpolation error for RS. Different RS are computed with different initial states. These RS are interpolated and resulting RS is compared with the computed RS of the dynamical system.

As discussed previously, the set of all final states of the feasible trajectories constitute the approximated RS. These trajectories are interpolated by univariate interpolation methods to obtain interpolated trajectories. State information from interpolated trajectories are used to obtain interpolated RS. Two scenarios with the associated amount of perturbation for initial states are described. Hausdorff distance is used as an error metric to compare the interpolated RS w.r.t. reference RS.

The interpolation uses Piecewise cubic Hermite splines for the trajectory synthesis. In order to solve the resulting interpolation problems, the B-spline basis is used. The method relies on interpolation of optimal trajectories described by Lockner (2014)<sup>70</sup> *et al.* In this method four different interpolation methods are introduced with the characteristics of each method. The requirements for the boundary conditions of the interpolated function is also explained.

### 9.1 Univariate Interpolation

During the interpolation trajectories from the two distinctive RS are used. Although an extension of the method is available for multivariate interpolation, the interpolated RS are obtained using univariate interpolation. Consider the spline space  $\mathbb{S}_{k,t}$  is defined by

$$\mathbb{S}_{k,t} = \text{span}\{B_{1,k}, \dots, B_{m,k}\} \quad (9.1)$$

where  $B_{1,k} = B_{1,k,t}$  denotes the  $i^{\text{th}}$  B-spline of order  $k$  for a given nondecreasing knot vector  $t = (t_i)_{i=1}^{m+k}$ . An element  $s \in \mathbb{S}_{k,t}$  is called a spline of order  $k$  with knots  $t$  and is denoted as

$$s(x) = \sum_{i=1}^m c_i B_{i,k}(x) \quad (9.2)$$

with real coefficients  $(c_i)_{i=1}^m$ . Each given set of  $k, t, m$  describes an interpolation method in the form Eq. (9.2). For a given data  $(x_i, f_i)_{i=1}^n$  with  $a = x_1 < x_2 < \dots < x_n = b$ , the coefficients  $(c_i)_{i=1}^m$  are computed such that

$$\begin{aligned} s(x_i) &= f_i, \quad i = 1, \dots, n \\ f &\in C^1[a, b] \end{aligned} \tag{9.3}$$

Afterwards the evaluation of Eq. (9.2) at a point  $x \in \mathbb{R}$  is carried out using De Boor's algorithm<sup>95</sup> and method described in reference<sup>70</sup> for univariate interpolation.

### Piecewise Cubic Hermite Interpolation

Piecewise cubic Hermite interpolation is used to synthesize the optimal trajectories. Since the information about derivatives  $f'_i$  is available, an interpolant which has a continuous first derivative can be constructed. The piecewise cubic Hermite interpolant is the solution of the following problem. Let the data  $(x_i, f_i, f'_i)_{i=1}^n$  with  $a = x_1 < x_2 < \dots < x_n = b$  be given. A function  $s$  is found that is a cubic polynomial on each subinterval  $[x_i; x_{i+1}]$  and interpolates the data in the sense that

$$s(x_i) = f_i \text{ and } s'(x_i) = f'_i, \quad i = 1, \dots, n \tag{9.4}$$

If  $k = 4$  and  $t$  as

$$t = (t_i)_{i=1}^{2n+4} = (x_1, x_1, x_1, x_1, x_2, x_2, \dots, x_{n-1}, x_{n-1}, x_n, x_n, x_n, x_n) \tag{9.5}$$

the unique solution  $s \in \mathbb{S}_{4,t} \subset C^1[a, b]$  of the above problem is given by

$$s(x) = \sum_{i=1}^{2n} c_i B_{i,4}(x) \tag{9.6}$$

with

$$c_{2i-1} = f_i - \frac{1}{3}h_{i-1}f'_i, \quad c_{2i} = f_i - \frac{1}{3}h_{i-1}f'_i, \quad i = 1, \dots, n$$

where  $h_i = x_{i+1} - x_i$  and  $x_0 := x_1, x_{n+1} := x_n$

The derivatives of  $(f'_i)_{i=1}^n$  can be estimated in the following way. The derivatives  $f'_1$  and  $f'_n$  are defined as the slope at  $x_1$  or  $x_n$  of the parabola that interpolates  $f$  at  $x_1, x_2, x_3$  or  $x_{n-2}, x_{n-1}, x_n$ . A short calculation gives

$$f'_i := \begin{cases} \frac{(2h_1+h_2)s_1-h_1s_2}{h_1+h_2}, & i = 1 \\ \frac{h_{i-1}s_i+h_is_{i-1}}{h_{i-1}+h_i}, & i = 2, \dots, n-1 \\ \frac{(2h_{n-1}+h_{n-2})s_{n-1}-h_{n-1}s_{n-2}}{h_{n-2}+h_{n-1}}, & i = n \end{cases} \tag{9.7}$$

where  $s_i = \frac{f_{i+1}-f_i}{h_i}$ . The cubic Hermite interpolation with this choice for derivatives is known as cubic Bessel interpolation. The approximation order of the cubic Hermite interpolant with exact derivatives is  $\mathcal{O}(h^4)$  for  $f \in C^4[a, b]$ .

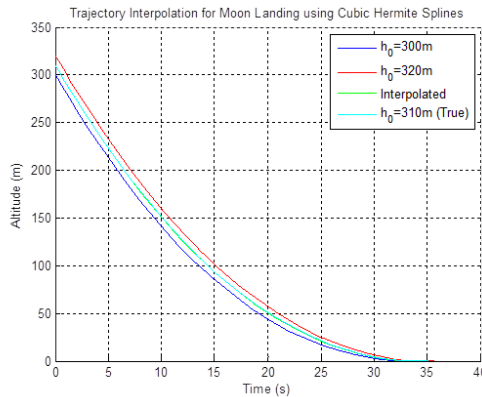
## 9.2 Trajectory Interpolation based Set Interpolation

Based on a pre-calculated grid of optimal solutions of a perturbed optimal control problem, it is described how an approximated solution to an actual given perturbation can be constructed in real-time, by using multivariate interpolation methods. In addition, relevant properties are proved, which carry over from the optimal solutions to the approximate solution. Furthermore, it is stated how an approximate solution can be efficiently computed numerically.

The algorithm considers perturbed optimal control problems with perturbations  $p \in \mathbb{R}^d$ . The task is to determine a control  $u(t;p) \in \mathbb{R}^m$  and resulting state  $x(t;p) \in \mathbb{R}^n$  in a given time interval  $[0; t_f]$ , which minimizes a given objective function, subject to the system of differential equations  $f$ , the initial and terminal conditions  $x_f$  and state and control constraints.

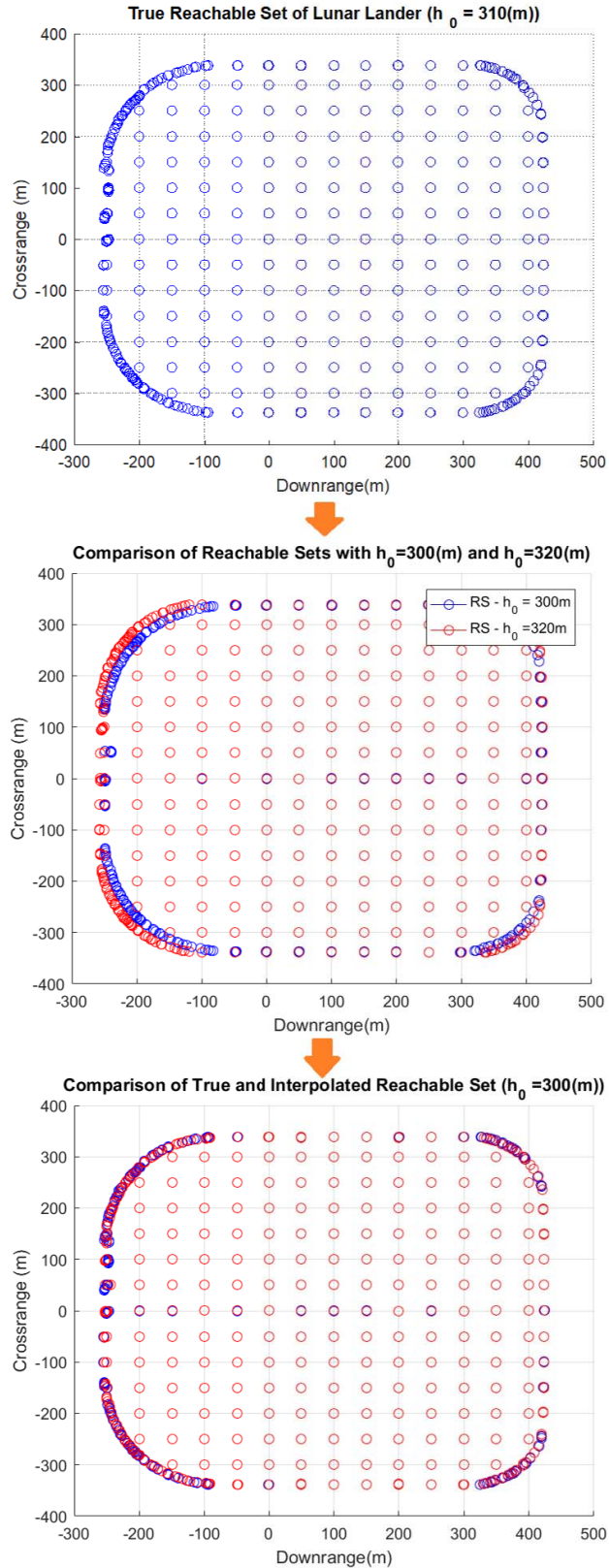
After the solution of the NLP associated to the OCP is obtained using two initial states, these values are interpolated using the univariate interpolation method with cubic Hermite interpolation. In this section, optimal trajectories have the same final time. However, Lockner (2014)<sup>70</sup> *et al.* also includes a scaling algorithm for interpolating optimal trajectories with different final time.

Figure 9.1 shows optimal trajectories with initial altitude  $h_1(t_0) = 300$  m,  $h_2(t_0) = 310$  m,  $h(t_0) = 320$  m. Trajectories with  $h(t_0) = 300$  m and  $h(t_0) = 320$  m are interpolated to obtain interpolated trajectory with  $h(t_0) = 310$  m. Interpolated and optimal trajectories are compared in the same figure. In the set interpolation, as shown in Figure 9.2, two different optimal trajectories with different initial conditions are interpolated to obtain interpolated RS. These two optimal trajectories are the solution to OCP in Eq. (2.10) for the same grid point. For the interpolation either states or control inputs can be used. In the case of interpolation of the control inputs, the states can be propagated with Runge-Kutta integration method. In this section, the interpolation is performed only using state information from the optimal trajectories.



**Figure 9.1:** Trajectory Interpolation with Different Initial States.

## 9.2 Trajectory Interpolation based Set Interpolation



**Figure 9.2:** Interpolation of RS with different Initial States and Comparison w.r.t. a Reference RS.

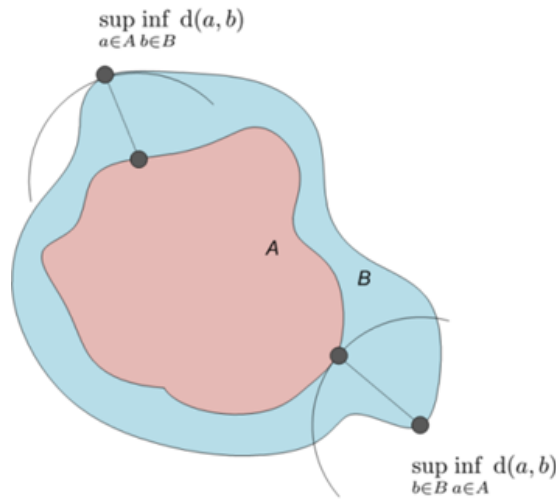
## 9. SET INTERPOLATION

---

### Hausdorff Distance

After the set interpolation there is a need for error metrics to assess the quality of the solution. The Hausdorff distance  $d_H$  measures the distance between two subsets of a metric space.

$$d_H(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} \quad (9.8)$$



**Figure 9.3:** Concept of Hausdorff Distance

The Hausdorff distance is the maximum of all the distances from a point in one set to the closest point in the other set as shown in Figure 9.3.

In order to use Hausdorff distance to determine the interpolation error, RS is computed for scenarios with two different initial states. The lower initial value  $x_l(t_0)$  and upper initial value  $x_u(t_0)$  are used for the computation of the RS. For the third case mean value  $x_m(t_0) = (x_u(t_0) + x_l(t_0))/2$  is obtained by interpolating the state trajectories by bivariate polynomial interpolation. As discussed earlier, either states are directly interpolated or interpolated control inputs are used to integrate equation of motions to obtain states. The interpolation considers the values of the states at the same collocation point and hence time.

A small and large deviation set is used to show the HD between interpolated and reference RS. The bounds on the deviations are considering the discussion in Monte Carlo analysis. The subsequent figures shows the results for the scenarios for small deviations. Appendix B includes the results for large deviation set. Table 9.1 summarizes the initial states, Hausdorff distance between the two interpolated sets and the results after interpolation. Similarly Table 9.2 includes the summary of data set used and the results for large deviations of the initial states.



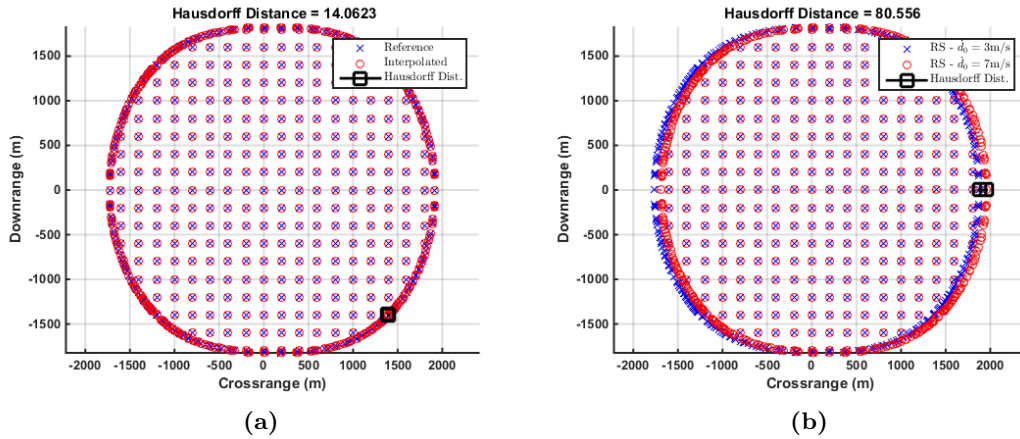
## 9.2 Trajectory Interpolation based Set Interpolation

**Table 9.1:** Results of Set Interpolation with Associated Hausdorff Distance for  $\Delta s$  Perturbations

	Initial Value	HD Before Interpolation	HD Between Reference RS
$\dot{d}$ (m/s)	$\pm 2$	80.55	14.06
$\dot{h}$ (m/s)	$\pm 2$	23.84	5.99
$\dot{c}$ (m/s)	$\pm 2$	80.79	16.29
$d$ (m)	$\pm 15$	29.87	5.31
$h$ (m)	$\pm 15$	3.47	0.84
$c$ (m)	$\pm 15$	29.86	5.15
$\beta$ ( $^\circ$ )	$\pm 2$	20.79	3.24
$\chi$ ( $^\circ$ )	$\pm 2$	20.22	3.44
$m$ (%)	$\pm 2$	1142.95	129.76

**Table 9.2:** Results of Set Interpolation with Associated Hausdorff Distance for  $2\Delta s$  Perturbations

	Initial Value	HD Before Interpolation	HD Between Reference RS
$\dot{d}$ (m/s)	$\pm 4$	160.99	28.72
$\dot{h}$ (m/s)	$\pm 4$	46.70	12.42
$\dot{c}$ (m/s)	$\pm 4$	161.14	36.39
$d$ (m)	$\pm 30$	59.87	14.96
$h$ (m)	$\pm 30$	6.59	1.68
$c$ (m)	$\pm 30$	59.73	10.99
$\beta$ ( $^\circ$ )	$\pm 4$	42.63	5.27
$\chi$ ( $^\circ$ )	$\pm 4$	40.86	6.37
$m$ (%)	$\pm 4$	2433.84	294.94



**Figure 9.4:** Set Interpolation with  $\dot{d}_0 = 3\text{m/s}$  and  $\dot{d}_0 = 7\text{m/s}$ .

## 9. SET INTERPOLATION

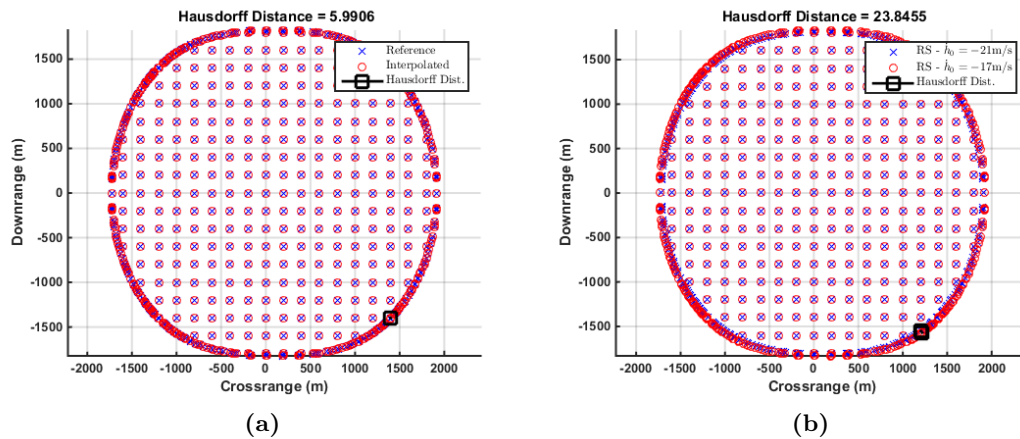


Figure 9.5: Set Interpolation with  $\dot{h}_0 = -21\text{m/s}$  and  $\dot{h}_0 = -17\text{m/s}$ .

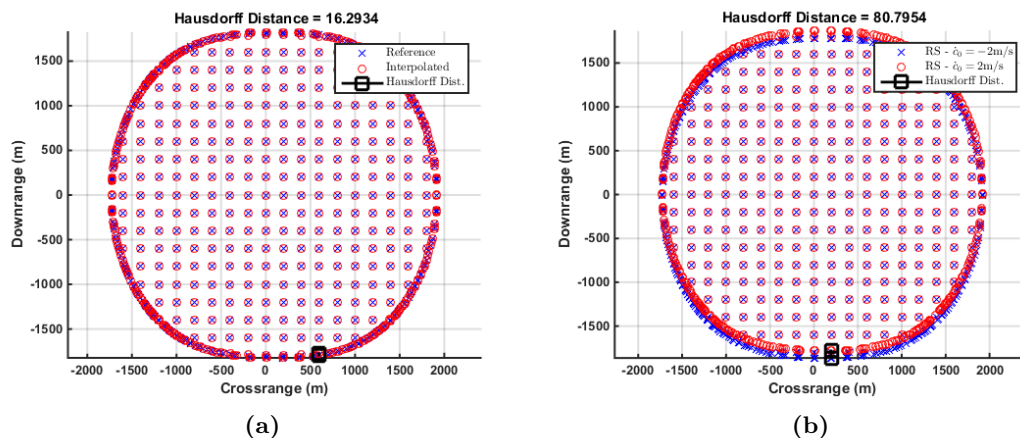


Figure 9.6: Set Interpolation with  $\dot{c}_0 = -2\text{m/s}$  and  $\dot{c}_0 = 2\text{m/s}$ .

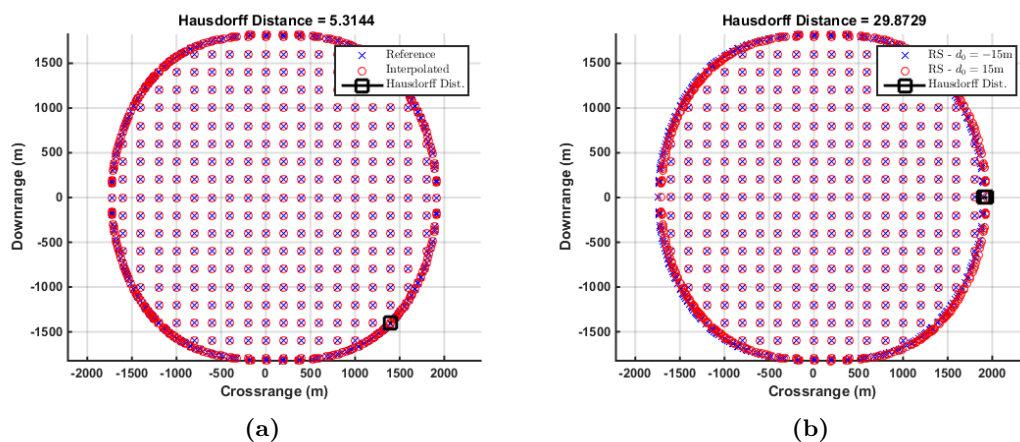


Figure 9.7: Set Interpolation with  $d_0 = -15\text{m}$  and  $d_0 = 15\text{m}$ .

## 9.2 Trajectory Interpolation based Set Interpolation

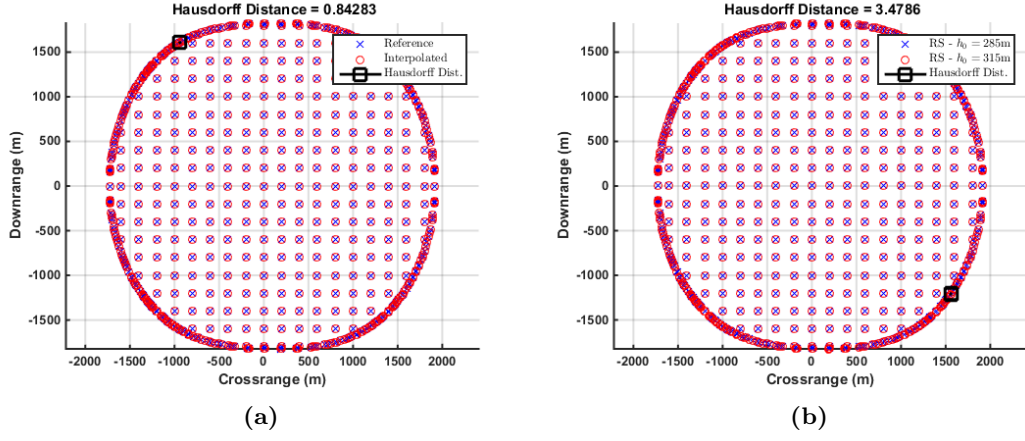


Figure 9.8: Set Interpolation with  $h_0 = 285\text{m}$  and  $h_0 = 315\text{m}$ .

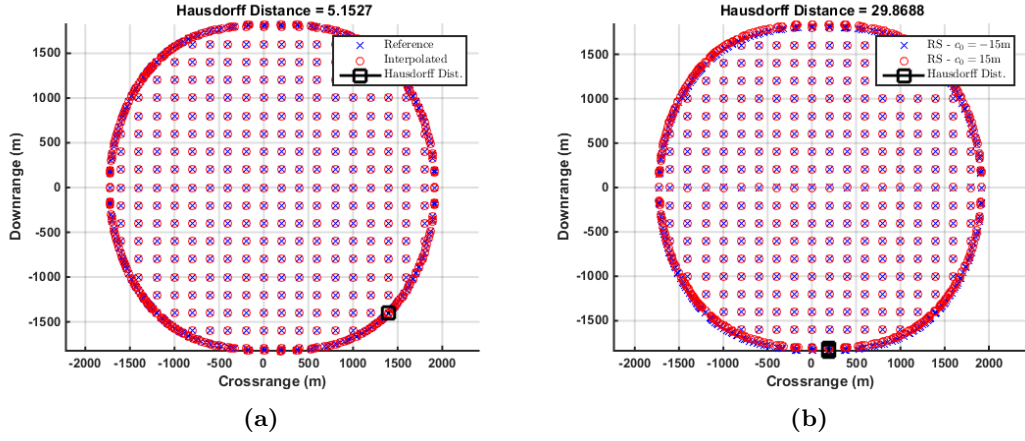


Figure 9.9: Set Interpolation with  $c_0 = -15\text{m}$  and  $c_0 = 15\text{m}$ .

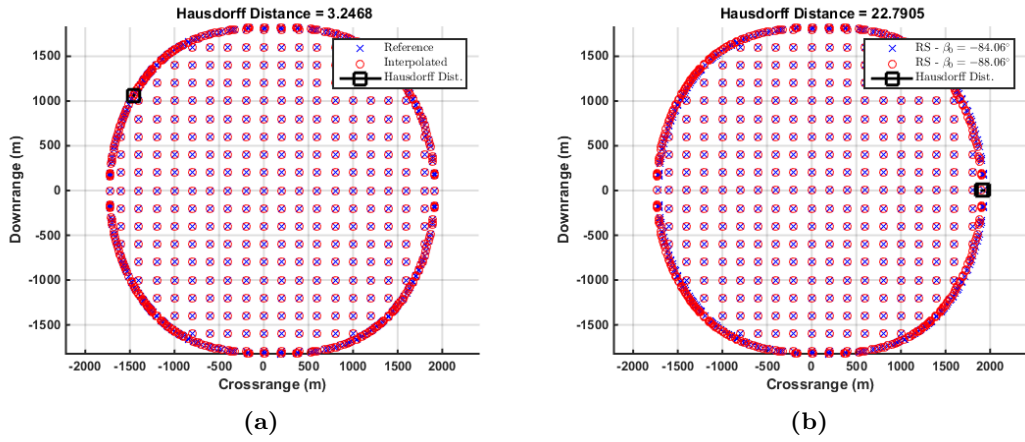


Figure 9.10: Set Interpolation with  $\beta_0 = -88.06^\circ$  and  $\beta_0 = -84.06^\circ$ .

## 9. SET INTERPOLATION

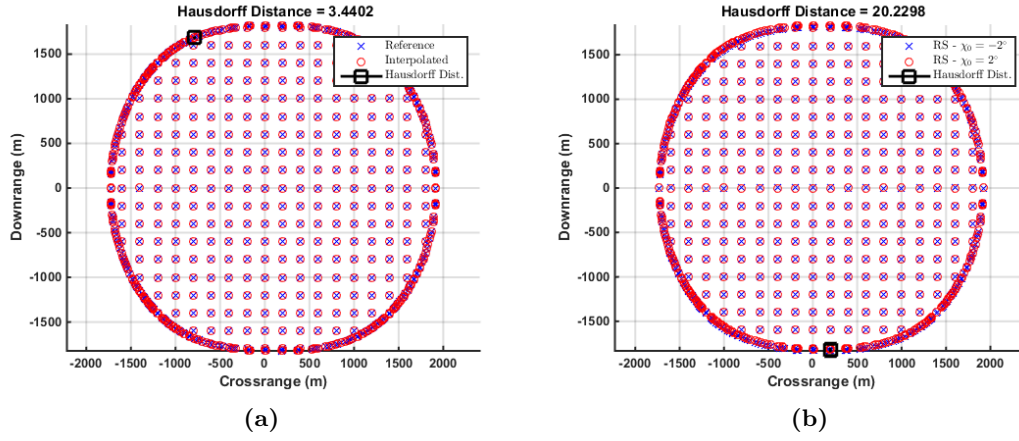


Figure 9.11: Set Interpolation with  $\chi_0 = -2^\circ$  and  $\chi_0 = 2^\circ$ .

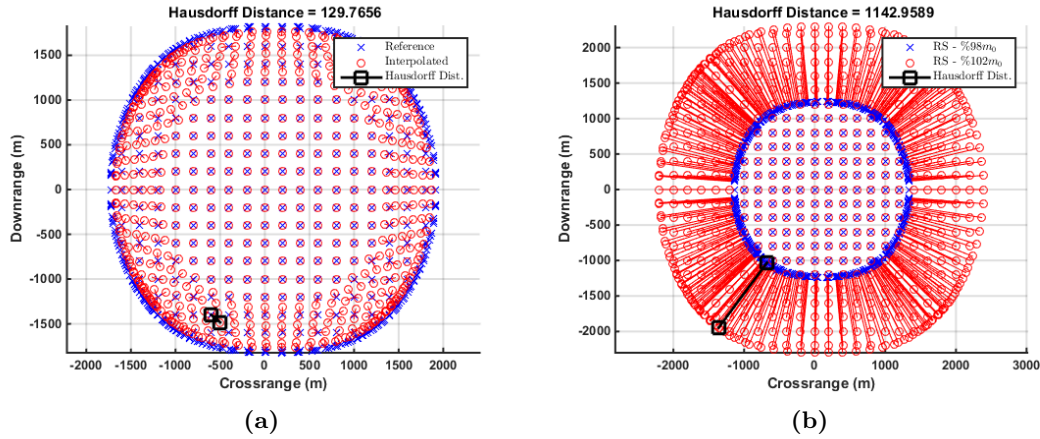


Figure 9.12: Set Interpolation with  $\%98 m_0$  and  $\%102 m_0$ .

It is seen that in for all states the HD between the interpolated and reference RS is smaller than the discretization error of the candidate landing surface of 200 m. The largest HD occurs for the initial deviation in initial mass. For the low perturbation HD is 129.76 and 294.94 m for high perturbation. As expected, when the deviation increases from the nominal solution, the interpolation error or in other words HD of the interpolated set w.r.t. reference set increases.

However, the interpolated states may not be realized with the actual dynamical system. In order to achieve the interpolated states, the dynamical system may violate the constraints for the control inputs. Therefore, the interpolated trajectories need to be verified by propagating the interpolated control inputs for large deviations.

In conclusion, the interpolation error is smaller than the discretization of the landing area for all scenarios. A RS library, which includes RS associated with different initial states, can be used to approximate the RS for different scenarios or initial conditions.

## Conclusion

In this thesis, an OCP based algorithm for approximating nonconvex reachable sets for continuous nonlinear dynamical systems is presented. The main principle is to find feasible state trajectories from the initial condition to the target state. Based on this idea, the algorithm discretizes the state space into a set of equidistant target points, and then for each grid point determines an optimal control law which minimizes the distance between the final state and the grid point.

The algorithm was applied to the terminal landing phase of a generic Moon landing scenario in order to approximate the attainable landing area with reachability analysis. Equality and inequality constraints on the final states were determined based on the safe landing conditions, final landing velocities and available propellant. Similarly, the maximum thrust provided by each of the thrusters was constrained, as well as the magnitude of the pitch and yaw rate. The algorithm uses the f-RPM to transcribe the OCP into NLP, which is then solved with standard optimization software yielding full state information at each of the predefined grid points.

Time-to-reach and propellant-to-reach maps were generated for the scenarios with different final time, bounds on the constraints and initial conditions. The RS computed with the developed algorithm constitutes the maximum attainable landing area of the lander and used as reference solution for comparison of other RS computations using a different guidance and control algorithm. It was shown that the area of the RS was increasing with the increasing final time. It was also demonstrated that the bounds on the thrust is an important factor on the area of the RS, and the bounds on the angular rates on the shape of the RS as shown in Figure 5.12 and Figure 5.11 respectively. With the increased bound on the thrust, the size of the RS also increased. In addition, the maximum allowed angular rates increased the agility of the lander, yielding a more circular safe landing region.

Moreover, in line with the spectral convergence behaviour of the PSM methods, the accuracy of the solution increased with the increase of nodes used for discretizing the OCP. However, the computation time increased exponentially with size of the problem. The RS was obtained in approximately 92 hours using 90 nodes on a desktop computer.

As a second example, a modified polynomial based Apollo guidance method was used

## 10. CONCLUSION

---

for comparing the performance of the developed algorithm. The nonlinear dynamics of the lander was inverted with feedback linearization and a Lyapunov based PD controller was designed to attenuate the inversion error. Trajectories were generated for the reference uniform gridded region on the Moon surface and solutions with active constraints were filtered. Open loop system is considered for the optimal control problem without any uncertainties or initial state error. The ratio of safe landing areas computed with two respective methods is 48.4% in favor of the developed main algorithm with a price of more computation time.

Monte Carlo simulations were carried out to show the robustness of the method and obtain sensitivity of the results based in dispersion on the initial states and parameters of the system. The nonlinear dynamics of the system were linearized around the reference trajectories. A Riccati-equation based controller was designed for the linearized system to track the reference trajectories. A normal distribution was assumed for the uncertainties, and the frequency of the successful landing was obtained for each specific point inside the reference RS. The largest ellipse was fitted inside the 100% success region and the predefined metrics were used to assess the results. It was shown that the area of the RS is most sensitive to the available propellant. The frequency of the successful landing within the reference RS decreased substantially with decrease in propellant mass.

In order to compare the MC results, RS with uncertainties were computed using unscented transform. A set of  $\sigma$ -points were chosen for the same sample mean and covariance used in MC simulations. These points were propagated with the nonlinear transformation and resulting multivariate probability density function was integrated with proper bounds. The probability of successful landing was evaluated for the points inside the reference RS and the results were compared with MC campaign. The results showed similar behaviour of the RS with uncertainty for both MC analysis and unscented transform based computations. The main advantage for using unscented transform is that the similar results with MC simulations were obtained with using substantially less number of simulations.

An interpolation method for reachable sets was proposed for the initial state errors. Two RS were computed with perturbed values if the initial states. The resulting trajectories were interpolated using cubic Hermite interpolation and the results were compared with the reference RS. It was shown that, in all cases the Hausdorff distance between the interpolated RS and reference RS was smaller than discretization error of the landing area.

In summary, the presented results showed the performance and limitations of the developed algorithm. The robustness and the accuracy of the algorithm was also demonstrated with different scenarios.

### 10.1 Suggested Future Work

The algorithm relies on computing optimal trajectories. The solution of associated NLP is time consuming and the computation speed for each single scenario is in the order of hours. In order to increase computation speed, the structure of the problem could be

exploited, better initial guess could be provided during the solution of NLP. Moreover smart mesh generation tools could be developed to reduce the number of points required to discretize the surface representing the landing area.

Real-time RS computation could provide essential information to HDA system. The major disadvantage of the trajectory based RS analysis is the demanding computational power. Interpolation of the off-line computed sets could be used to approximate RS with an acceptable error margin. Pre-computed RS library coupled with a suitable interpolation method might yield real-time capable solution. Moreover, necessary tools could be developed to synthesize the data from different sensors and use the results in decision making process during the mission phase.

Finally, the data obtained from RS computations could be also used for mission analysis. It was seen that, the shape and the area of the successful landing region was changing with different controller parameters. The gains could be optimized to obtain maximum landing region. Similarly, the area of the safe landing zone or rate of successful landing could be evaluated for different system parameters.

# References

- [1] F. ZHAO. *Automatic Analysis and Synthesis of Controllers for Dynamical Systems Based on Phase Space Knowledge*. PhD thesis, Massachusetts Institute of Technology, 1993. 2
- [2] M. ALTHOFF. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, Technical University of Munich, 2010. 2, 12, 13
- [3] P. A. CROLEY. *Reachable Sets for Multiple Asteroid Sample Return Missions*. Master's thesis, Naval Postgraduate School-Monterey California, 2005. 2
- [4] G. CHIN, S. BRYLOW, M. FOOTE, J. GARVIN, J. KASPER, J. KELLER, M. LITVAK, I. MITROFANOV, D. PAIGE, K. RANEY, M. ROBINSON, A. SANIN, D. SMITH, H. SPENCE, P. SPUDIS, S. A. STERN, AND M. ZUBER. **Lunar Reconnaissance Orbiter Overview: The Instrument Suite and Mission**. *Space Science Reviews*, **129**(4):391--419, 2007. 4
- [5] Y. E. ARSLANTAS, T. OEHLISCHLÄGEL, M. SAGLIANO, S. THEIL, AND C. BRAXMAIER. **Approximation of Attainable Landing Area of a Moon Lander by Reachability Analysis**. In *17th International Conference on Hybrid Systems: Computation and Control*, 2014. 7, 20
- [6] Y. E. ARSLANTAS, T. OEHLISCHLÄGEL, M. SAGLIANO, S. THEIL, AND C. BRAXMAIER. **Safe Landing Area Determination for a Moon Lander by Reachability Analysis**. In *International Astronautical Congress, Toronto*, 2014. 7, 20
- [7] Y. E. ARSLANTAS, T. OEHLISCHLÄGEL, AND M. SAGLIANO (IN PROGRESS). **Safe Landing Area Determination for a Moon Lander by Reachability Analysis**. *Acta Astronautica*. 7, 48
- [8] Y. E. ARSLANTAS AND T. OEHLISCHLÄGEL. **Real-time Capable Nonlinear Model Predictive Controller Design for The Upper Stage of a Launch Vehicle**. In *30th International Symposium on Space Technology and Science, Kobe/Japan*, 2015. 8
- [9] L. HUNEKER, M. SAGLIANO, AND Y. E. **SPARTAN: An Improved Global Pseudospectral Algorithm for High-fidelity Entry-Descent-Landing**



- 
- Guidance Analysis.** In *30th International Symposium on Space Technology and Science*, 2015. 8, 19
- [10] Y. E. ARSLANTAS AND T. OEHLISCHLÄGEL. **Real-time Capable Nonlinear Model Predictive Controller Design for The Upper Stage of a Launch Vehicle.** *Transactions of Japan Society for Aeronautical and Space Sciences (JSASS)*. 8
- [11] V. D'ONOFRIO, MARCO SAGLIANO, AND Y. E. ARSLANTAS. **Exact Hybrid Jacobian Computation for Optimal Trajectory Generation via Dual Number Theory.** In *AIAA SciTech Guidance, Navigation, and Control Conference*, San Diego, California, USA, January 2016. 8, 42
- [12] Y. E. ARSLANTAS AND S. THEIL (IN PROGRESS). **Attainable Landing Area Computation of a Lunar Lander with Uncertainty by Reachability Analysis.** In *Euro GNC 2017 - 4th CEAS Specialist Conference on Guidance, Navigation and Control*, Warsaw, Poland, April 2017. 8
- [13] R. BAIER, C. BÜSKENS, I. A. CHAHMA, AND M. GERDTS. **Aproximation of Reachable Sets by Direct Solution Methods of Optimal Control Problems.** *Optimization Methods and Software*, **22 (3)**:433--452, 2007. 12, 15
- [14] J. LYGEROS, C. TOMLIN, AND S. SASTRY. **Controllers for reachability specifications for hybrid systems.** *Automatica*, **35**:349--370, 1999. 12
- [15] I. KITSIOS AND J. LYGEROS. **Aerodynamic Envelope Computation for Safe Landing of the HL-20 Personnel Launch Vehicle using Hybrid Control.** In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, 2005. 12
- [16] J. BENITO AND K. D. MEASE. **Characterizing the Controllable Entry States and Reachable Sites for Planetary Landing.** In *6th International Planetary Probe Workshop, Atlanta, GA*, 2008. 12
- [17] J. BENITO AND K. D. MEASE. **Reachable and Controllable Sets for Planetary Entry and Landing.** *Journal of Guidance, Control and Dynamics*, **33**:2010, 2010. 12, 20
- [18] I. M. MITCHELL, A. M. BAYEN, AND C. J. TOMLIN. **A Time-Dependent HamiltonJacobi Formulation of Reachable Sets for Continuous Dynamic Games.** In *IEEE Transactions on Automatic Control*, 2005. 12
- [19] I. KAUSA, R. BAIER, M. GERDTS, M. GONTER, AND C. WEGWERTH. **Avoidance Trajectories for Driver Assistance Systems via Solvers for Optimal Control Problems.** In *Proceedings of the 20th International Symposium on Mathematical Theory of Networks and Systems (MTNS2012), Melbourne, Australia*, 2012. 12, 16

## REFERENCES

---

- [20] J. H. GILLULA, G. M. HOFFMANN, H. HUANG, M. P. VITUS, AND C. J. TOMLIN. **Applications of hybrid reachability analysis to robotic aerial vehicles.** *The International Journal of Robotics Research*, **30**:335--354, 2011. 12
- [21] T. DANG, C. L. GUERNIC, AND O. MALER. **Computing reachable states for nonlinear biological models.** *Theoretical Computer Science*, **412**:2095--2107, 2011. 12
- [22] M. J. HOLZINGER AND D. J. SCHEERES. **Reachability Set Subspace Computation for Nonlinear Systems using Sampling Methods.** In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) Orlando, FL, USA, December 12-15, 2011*. 12
- [23] O. BOTCHKAREV AND S. TRIPAKIS. **Verification of Hybrid Systems with Linear Differential Inclusions Using Ellipsoidal Approximations.** In *In proceedings of Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA,, 2000*. 12
- [24] H. YAZAREL AND G. J. PAPPAS. **Geometric Programming Relaxations for Linear System Reachability.** In *Proceeding of the 2004 American Control Conference Boston, Massachusetts June 30 - July 2, 2004*. 12
- [25] A. GIRARD. **Reachability of Uncertain Linear Systems Using Zonotopes.** *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, **3414**:291--305, 2005. 12, 13
- [26] I. HWANG, D. M. STIPANOVIC, AND C. J. TOMLIN. *Polytopic Approximations of Reachable Sets Applied to Linear Dynamic Games and a Class of Nonlinear Systems*. Springer, 2005. 12
- [27] C. L. GUERNIC. *Reachability Analysis of Hybrid Systems with Linear Continuous Dynamics*. PhD thesis, Universite Grenoble I Joseph Fourier, 2009. 12
- [28] A. CHUTINAN AND B. H. KROGH. **Computational Techniques for Hybrid System Verification.** *IEEE Transactions on Automatic Control*, **48**:64--75, 2003. 12, 13, 121
- [29] T. DANG AND O. MALER. **Reachability Analysis via Face Lifting.** *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, **1386**:96--109, 1998. 12
- [30] R. BAIER AND M. GERDTS. **A Computational Method for Non-Convex Reachable Sets using Optimal Control.** In *Proceedings of the European Control Conference (ECC) 2009, August 23-26, Budapest, Hungary, 2009*. 12, 13, 16, 50, 51

- 
- [31] A. B. KURZHANSKI AND P. VARAIYA. **On Ellipsoidal Techniques For Reachability Analysis. Part I: External Approximations.** *Optimization Methods and Software*, **17**:177--206, 2001. 13
- [32] I. M. MITCHELL. *Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Problems.* PhD thesis, Stanford University, 2002. 13
- [33] A.B. KURZHANSKI AND P. VARAIYA. **Ellipsoidal Techniques for Reachability Analysis: Internal Approximation.** *Systems and Control Letters*, **41**:201--211, 2000. 13
- [34] O. STURBERG AND B. H. KROGH. **Efficient Representation and Computation of Reachable Sets for Hybrid Systems.** *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, **2623**:482--497, 2003. 13
- [35] A. GIRARD, C. L. GUERNIC, AND O. MALER. **Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs.** *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, **3927**:257--271, 2006. 13
- [36] S. PRAJNA AND A. JADBABAIE. **Safety Verification of Hybrid Systems Using Barrier Certificates.** In *Hybrid Systems: Computation and Control*, **2993**:477--492, 2004. 14
- [37] L. GRUNE AND T. U. JAHN. **Computing Reachable Sets Via Barrier Methods on SIMD Architectures.** In *In Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria, September 10-14, 2012.* 15
- [38] R. BAIER, M. GERDTS, AND I. XAUSA. **Approximation of Reachable Sets using Optimal Control Algorithms.** 2012. 16
- [39] R. BAIER. **Set-Valued Euler's Method With Interpolated Distance Functions and Optimal Control.** In *Proceedings on the 20th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2012), July 9-13, 2012, Melbourne, Australia, 2012.* 16
- [40] G. LEITMANN. *The Calculus of Variations and Optimal Control.* New York: Springer, 1981. 17
- [41] D. G. HULL. *Optimal Control Theory for Applications.* New York: Springer-Verlag, 2003. 17
- [42] H. B. KELLER. *Numerical Solution of Two Point Boundary Value Problems.* SIAM, 1976. 17

## REFERENCES

---

- [43] J. STOER AND R. BULIRSCH. *Introduction to Numerical Analysis*. Springer-Verlag, 2002. 17
- [44] R. D. RUSSELL AND L. F. SHAMPINE. **A Collocation Method for BouBound Value Problems**. *Numerical Mathematics*, **19**(1):1--28, 1972. 17
- [45] J. T. BETTS. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, 2011. 18, 19
- [46] CHRISTOF BÜSKENS AND DENNIS WASSEL. *Modeling and Optimization in Space Engineering*, chapter The ESA NLP Solver WORHP, pages 85--110. Springer New York, New York, NY, 2013. 18
- [47] P. GILL, W. MURRAY, AND M.A. SAUNDERS. **User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming**. Technical report, University of California, 2007. 18
- [48] RICHARD H. BYRD, JORGE NOCEDAL, AND RICHARD A. WALTZ. *Large-Scale Nonlinear Optimization*, chapter Knitro: An Integrated Package for Nonlinear Optimization, pages 35--59. Springer US, Boston, MA, 2006. 18
- [49] L.T. BIEGLER AND V.M. ZAVALA. **Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization**. *Computers & Chemical Engineering*, **33**(3):575 -- 582, 2009. Selected Papers from the 17th European Symposium on Computer Aided Process Engineering held in Bucharest, Romania, May 2007. 18
- [50] JOHN T. BETTS. **Survey of Numerical Methods for Trajectory Optimization**. *Journal of Guidance, Control, and Dynamics*, **21**(2):193--207, March 1998. 18
- [51] G. L. BRAUER, D. E. CORNICK, AND R. STEVENSON. **Capabilities and Applications of the Program to Optimize and Simulate Trajectories**. Technical report, NASA-CR-2770, 1977. 19
- [52] F. FAHROO AND I. M. ROSS. **Costate Estimation by a Legendre Pseudospectral Method**. *Journal of Guidance, Control and Dynamics*, **24**:270--277, 2001. 19, 34
- [53] F. FAHROO AND I. M. ROSS. **Direct Trajectory Optimization by a Chebyshev Pseudospectral Method**. *Journal of Guidance, Control and Dynamics*, **25**:160--166, 2002. 19, 34
- [54] I. M. ROSS AND F. FAHROO. *New Trends in Nonlinear Dynamics and Control and their Applications*, chapter Legendre Pseudospectral Approximations of Optimal Control Problems, pages 327--342. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. 19

- 
- [55] J. T. BETTS. **SOCS User's Guide**. Technical Report M&CT-TECH-01-014, The Boeing Company, 2009. 19
- [56] I. M. ROSS AND F. FAHROO. **User's Manual for DIDO 2001  $\alpha$  : a MATLAB Application for Solving Optimal Control Problems**. Technical Report AAS-01-03, Department of Aeronautics and Astronautics, Naval Postgraduate School, 2001. 19
- [57] MICHAEL A. PATTERSON AND ANIL V. RAO. **GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Non-linear Programming**. *ACM Trans. Math. Softw.*, **41**(1):1:1--1:37, 2014. 19
- [58] M. FAHROO. F. AND ROSS. **On Discrete-Time Optimal Conditions for Pseudospectral Methods**. In *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, Colorado, August 2006. 19
- [59] R. K. CHENG. **Surveyor Terminal Guidance**. Technical report, NASA CR-57550, 1966. 20
- [60] S. J. CITRON, S. E. DUNIN, AND H. F. MEISSINGER. **A Terminal Guidance Technique for Lunar Landing**. *AIAA Journal*, **2**:503--309, 1964. 20
- [61] G. W. CHERRY. **A General, Explicit, Optimizing Guidance Law for Rocket-Propelled Spaceflight**. In *AIAA Astrodynamics Guidance and Control Conference*. AIAA-64-638, 1964. 20
- [62] G. W. CHERRY. **A Class of Unified Explicit Methods for Steering Throttleable and Fixed-Thrust Rockets**. Technical report, MIT, 1964. 20
- [63] A. R. KLUMPP. **Apollo Guidance, Navigation and Control**. Technical report, NASA, 1971. 20, 63
- [64] R. R. SOSTARIC AND J. R. REA. **Powered descent guidance methods for the Moon and Mars**. In *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, 2005. 20, 61
- [65] J. DELAUNE, D. D. ROSA, AND S. HOBBS. **Guidance and Control System Design for Lunar Descent and Landing**. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2010. 20
- [66] G. L. CARMAN, D. G. IVES, AND D. K. GELLER. **Apollo-derived Mars Precision Lander Guidance**. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 1998. 20
- [67] R. R. SOSTARIC AND J. R. REA. **Powered Descent Trajectory Guidance and Some Considerations for Human Lunar Landing**. In *Annual ASS Guidance and Control Conference*, 2007. 20

## REFERENCES

---

- [68] B. A. STEINFELDT, M. J. GRANT, D. M. MATZ, R. D. BRAUN, AND G. H. BARTON. **Guidance, Navigation, and CControl Technology System Trades for Mars Pinpoint Landing.** In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. AIAA-11-7217, 2008. 20
- [69] T. OEHLISCHLÄGEL, S. THEIL, H. KRÜGER, M. KNAUER, J. TIETJEN, AND C. BÜSKENS. **Optimal Guidance and Control of Lunar Landers with Non-throtttable Main Engine.** In *Advances in Aerospace Guidance, Navigation and Control*. Springer Berlin Heidelberg, 2011. 20, 24, 31
- [70] E. LOCKNER, T. OEHLISCHLÄGEL, S. THEIL, M. KNAUER, J. TIETJEN, AND C. BÜSKENS. **Real-time Capable Trajectory Synthesis via MultivaMulti Interpolation Methods for a Moon Landing Maneuver.** *CEAS Space Journal*, **6**:107--118, 2014. 20, 100, 101, 102
- [71] B. AÇIKMEŞE AND S.R. PLOEN. **A powered descent guidance algorithm for Mars pinpoint landing.** In ENZ, editor, *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, 2005. 20, 61
- [72] B. AÇIKMEŞE, J. M. CARSON, AND L. BLACKMORE. **Lossless Convexification of Non-Convex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem.** *IEEE Transactions on Control Systems Technology*, **21**:2104--2113, 2013. 20
- [73] ROBERT F. STENGEL. **Manual Attitude Control of the Lunar Module.** *Journal of Spacecraft and Rockets*, **7**(8):941--948, 1970. 32
- [74] D. GARG. *Advances in Global Pseudospectral Methods for Optimal Control.* PhD thesis, University of Florida, 2011. 34, 41
- [75] D. GARG, M. A. PATTERSON, C. FRANCOLIN, C. L. DARBY, G.T. HUNTINGTON, W.W. HAGER, AND A. V. RAO. **Direct trajectory optimization and costate estimation offinite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method.** *Computational Optimization and Applications*, **49**:335--358, 2011. 34
- [76] G. D. HUNTINGTON. *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems.* PhD thesis, Massachusetts Institute of Technology, 2007. 34
- [77] M. SAGLIANO, M. SAMAAAN, S. THEIL, AND E. MOOIJ. **SHEFEX-3 Optimal Feedback Entry Guidance.** In *AIAA SPACE 2014 Conference and Exposition*, August 2014. 35
- [78] M. SAGLIANO. **Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems.** *Operations Research Letters*, **42**:213--216, 2014. 35, 46

- 
- [79] MARCO SAGLIANO. *Development of a Novel Algorithm for High Performance Reentry Guidance*. PhD thesis, University of Bremen, 2016. 35
- [80] J. P. BERRUT AND L. N. TREFETHEN. **Barycentric Lagrange Interpolation**. *SIAM Review*, **46**:501--517, 2004. 39
- [81] M. SAGLIANO AND S. THEIL. **Hybrid Jacobian Computation for Fast Optimal Trajectories Generation**. In *AIAA Guidance, Navigation and Control(GNC) Conference*, August 2013. 41, 45
- [82] K.-L. LAI AND J.L. CRASSIDIS. **Extensions of the first and second complex-step derivative approximations**. *Journal of Computational and Applied Mathematics*, **219**:276--293, 2008. 42
- [83] C. ZHU, R. H. BYRD, P. LU, AND J. NOCEDAL. **L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization**. *ACM Transactions on Mathematical Software*, **23(4)**:550--560, 1997. 45
- [84] F. NAJSON AND K.D. MEASE. **A computationally non-expensive guidance algorithm for fuel efficient soft landing**. In *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, 2005. 61
- [85] CHRISTINA T. CHOMEL. *Development of an Analytical Guidance Algorithm for Lunar Descent*. PhD thesis, The University of Texas at Austin, 2007. 66, 69
- [86] PAOLO LUNGI, MICHELE LAVAGNA, AND ROBERTO ARMELLIN. **A semi-analytical guidance algorithm for autonomous landing**. *Advances in Space Research*, **55(11)**:2719 -- 2738, 2015. 80
- [87] ROBERTO FURFARO, DANIEL R. WIBBEN, BRIAN GAUDET, AND JULES SIMO. **Terminal Multiple Surface Sliding Guidance for Planetary Landing: Development, Tuning and Optimization via Reinforcement Learning**. *The Journal of the Astronautical Sciences*, **62(1)**:73--99, 2015. 80
- [88] R. FURFARO, D. CERSOSIMO, AND D. R. WIBBEN. **Asteroid Precision Landing via Multiple Sliding Surfaces Guidance Techniques**. *Journal of Guidance, Control and Dynamics*, **36(4)**:1075--1092, 2013. 80
- [89] D. R. WIBBEN AND R. FURFARO. **Terminal Guidance for Lunar Landing and Retargeting Using a Hybrid Control Strategy**. *Journal of Guidance, Control and Dynamics*, **5(5)**:1168--1172, 2016. 80
- [90] KENNETH J. DAVIDIAN, RONALD H. DIECK, AND ISAAC CHUANG. **A Detailed Description of the Uncertainty Analysis for High Area Ratio Rocket Nozzle Tests at the NASA Lewis Research Center**. Technical report, NASA, 1987. 80

## REFERENCES

---

- [91] MURRAY L. KERR, MIGUEL HAGENFELDT, JOSE A. OSPINA, JOSE MIGUEL RAMN, AND LUIS F. PENIN. **ESA Lunar Lander: Approach Phase Concept and G&C Performance.** In *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, MA, August 19-22 2013. 81
- [92] J. K. UHLMANN S. J. JULIER AND H. F. DURRANT-WHYTE. **A new approach for filtering nonlinear systems.** In *American Control Conference, Proceedings of the 1995*, **3**, pages 1628--1632, June 1995. 91
- [93] S. J. JULIER. **The scaled unscented transformation.** In *Proceedings of the 2002 American Control Conference*, **6**, pages 4555--4559, May 2002. 92
- [94] A. GENZ AND F. BRETZ. **Comparison of Methods for the Computation of Multivariate t Probabilities.** *Journal of Computational and Graphical Statistics*, **11**(4):950--971, 2002. 96
- [95] C. D. BOOR. *A Practical Guide to Splines*. Springer New York, 2001. 101
- [96] A. CHUTINAN AND B. H. KROGH. **Verification of Polyhedral-Invariant Hybrid Automata Using Polygonal Flow Pipe Approximations.** *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, **1569**:76--90, 1999. 121



## Appendix A

# Sample Set Computations

Use of polytopes to compute reachable is a simple, fast method for overapproximating the reachable sets. Following algorithm is applied for Van Der Pol system as described in<sup>9628</sup>. Idea is to obtain the boundary of the reachable set at discrete time steps and extend it in a way that it covers the whole trajectories. Given a finite set of points  $\Gamma$ , convex hull of  $\Gamma$  is the smallest convex set (a polytope) that contains  $\Gamma$ . Let  $\text{Ch}(\Gamma)$  denote the convex hull of  $\Gamma$ . Considering the system defined in (2.1) with given set of initial states  $X_0$  which is a polytope, polyhedral approximation  $\hat{\mathcal{R}}_{[0,t_f]}(X_0)$  of the Reachable set  $\mathcal{R}_{[0,t_f]}(X_0)$  computed by

$$\mathcal{R}_{[0,t_f]}(X_0) \subseteq \hat{\mathcal{R}}_{[0,t_f]}(X_0)$$

Let  $\text{Poly}(C,d)$  denote a convex polytope defined by the pair  $(C,d) \in R^{m \times n} \times R^m$  according to

$$\text{Poly}(C,d) = \{x \mid Cx \leq d\}$$

Each row of  $C$  is the normal vector to  $i^{\text{th}}$  face of a polytope. A polytope  $P$  has a finite number of vertices which are points in  $P$  that cannot be written as a strict convex combination of any other points in  $P$ . Let's denote  $V(P)$  the set of vertices of  $P$ .

Objective is to compute an outer approximation to the entire flow pipe, the polytope approximating the the  $k^{\text{th}}$  segment of the flow pipe should contain that particular segment. If the approximating polytope corresponds to a matrix pair  $(C,d)$  then we want;

$$\mathcal{R}_{[t_{k-1},t_k]}(X_0) \subseteq \text{Poly}(C,d)$$

Suppose that  $C$  is given. To obtain minimal approximation error for fixed  $C$ , we compute  $d$  to solve the following optimization problem:

$$\begin{aligned} \min_d \quad & \text{volume}[\text{Poly}(C,d)] \\ \text{s.t.} \quad & \mathcal{R}_{[t_{k-1},t_k]}(X_0) \subseteq \text{Poly}(C,d) \end{aligned} \tag{A.1}$$

We denote the minimal set  $\text{Poly}(C,d^*)$  which is the solution to (A.1) by  $S_C^{\text{min}}\mathcal{R}_{[t_{k-1},t_k]}(X_0)$ . The components of  $d^*$  solving (A.1) can be found by solving the following constrained

## A. SAMPLE SET COMPUTATIONS

---

optimization problems for  $i = 1, \dots, m$ .

$$\begin{aligned} \max_{x_0, t} \quad & c_i^T x(t, x_0) \\ \text{s.t.} \quad & x_0 \in (X_0) \\ & t \in [t_{k-1}, t_k] \end{aligned} \tag{A.2}$$

Let  $(x_0^*, t^*)$  be a solution to (A.2), then solution to (A.1) is given by  $d_i^* = c_i^T(t^*, x_0^*)$ , for  $i = 1, \dots, m$ . Let the time interval  $[0, t_f]$  be divided into  $N$  time steps  $[0, t_1], [t_1, t_2], \dots, [t_{N-1}, t_N]$ . The complete flow pipe approximation from  $t = 0$  to  $t = t_f$  is the union of all  $N$  flow pipe segments.

$$\hat{\mathcal{R}}_{[0, t_f]}(X_0) = \bigcup_{k=1, \dots, N} \hat{\mathcal{R}}_{[t_{k-1}, t_k]}(X_0) \tag{A.3}$$

As an example resulting approximation for the Van Der Pol equation

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -0.2(x_1^2 - 1)x_2 - x_1 \end{aligned} \tag{A.4}$$

In this example 2 simulations were run starting from extreme points of the initial set. Later on flow pipe is approximated by  $N$  time equal sets. Afterwards coordinates of the vertices obtained and  $Poly(C, d)$  was constructed. Direction vectors  $d$  which define the polytope were extracted. These direction vectors were bloated such that all the path lies within the new boundaries. This was done by solving the optimization problem in Eq. (A.1). Finally a new polytope  $Poly(C, d^*)$  was constructed and union of these segments overapproximates the true RS.

# Appendix B

## Additional Reachable Sets

This section includes additional reachable maps for the lunar lander. In Chapter 9, a trajectory interpolation method is used to measure the interpolated RS with different initial states. The Hausdorff distance values are tabulated in this Chapter together with interpolated sets. Appendix B.1 includes additional Figures that are not included in Chapter 9.

In Chapter 7, uncertainties are defined for initial states and system parameters. Attainable maps of the lander are computed for the closed loop system. Appendix B.2 includes additional reachable maps following the discussion in this chapter. The Figures make a comparison of RS and %100 success RS with uncertainty for initial states and system parameters. In addition, the nonconvex %100 success RS and maximum area ellipses were illustrated.

### B.1 Additional Interpolated RS

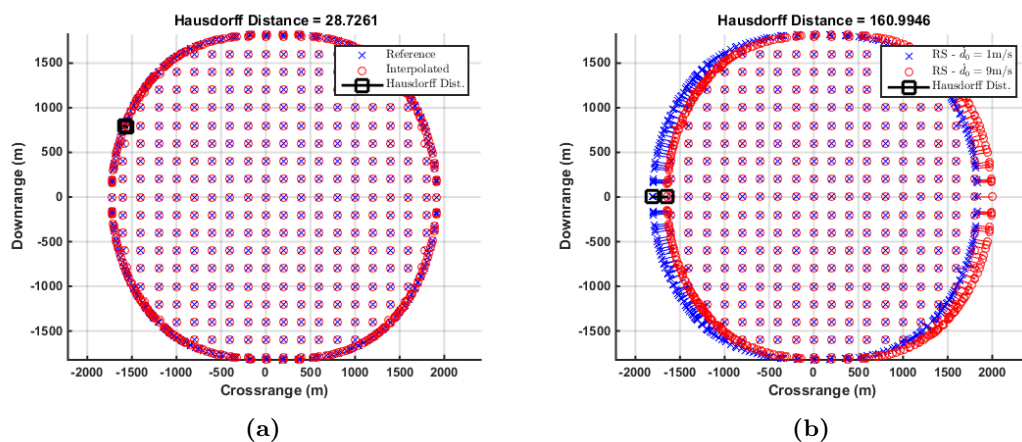


Figure B.1: Set Interpolation with  $\dot{d}_0 = 1\text{m/s}$  and  $\dot{d}_0 = 9\text{m/s}$ .

## B. ADDITIONAL REACHABLE SETS

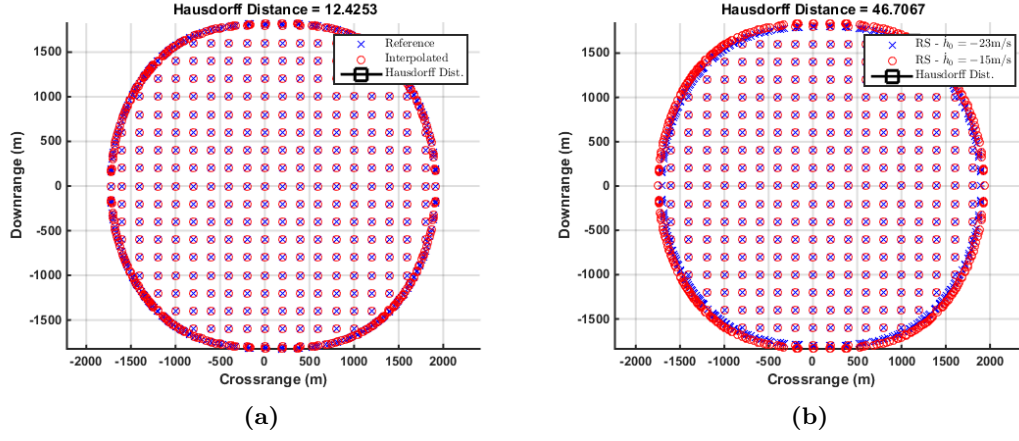


Figure B.2: Set Interpolation with  $\dot{h}_0 = -23\text{m/s}$  and  $\dot{h}_0 = -15\text{m/s}$ .

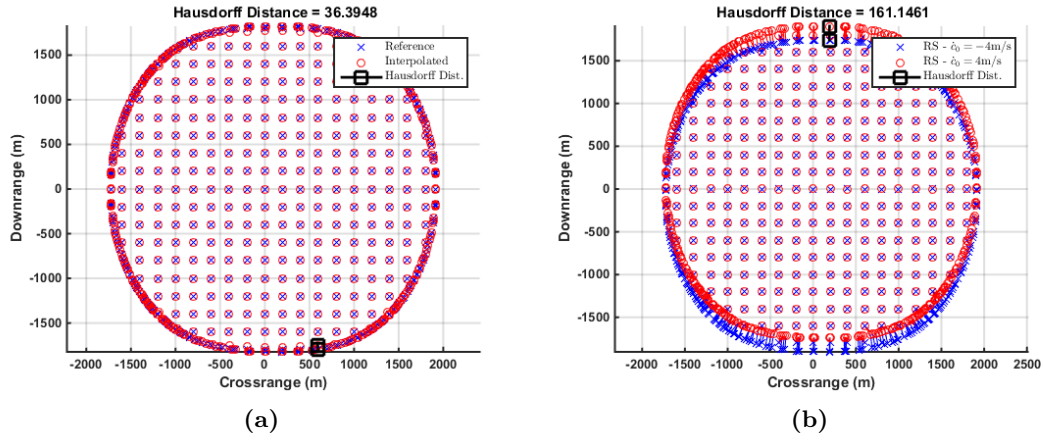


Figure B.3: Set Interpolation with  $\dot{c}_0 = -4\text{m/s}$  and  $\dot{c}_0 = 4\text{m/s}$ .

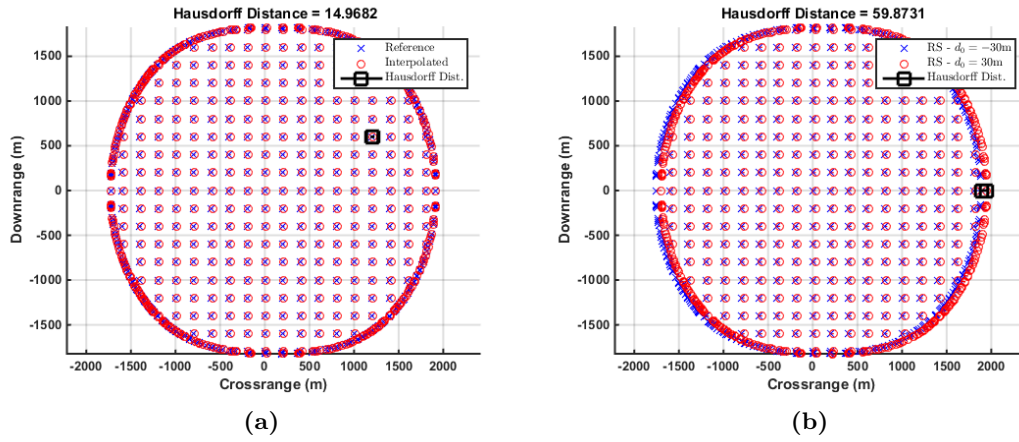


Figure B.4: Set Interpolation with  $d_0 = -30\text{m}$  and  $d_0 = 30\text{m}$ .

## B.1 Additional Interpolated RS

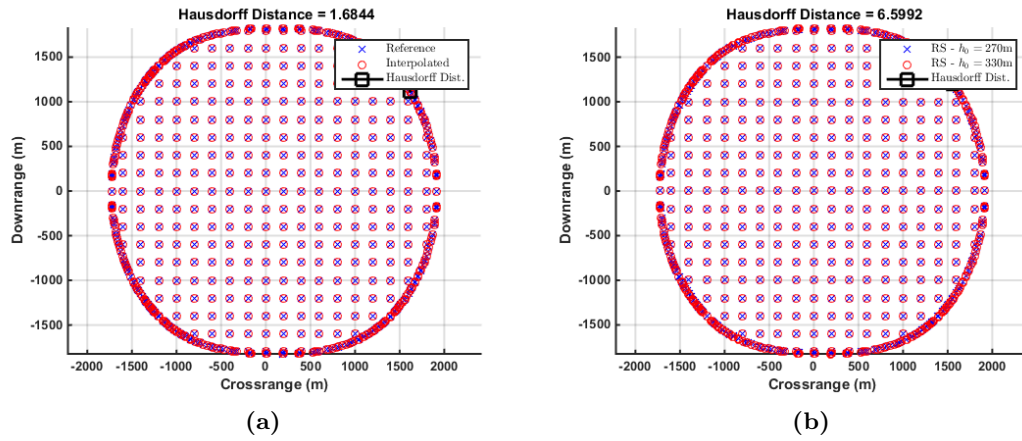


Figure B.5: Set Interpolation with  $h_0 = 270\text{m}$  and  $h_0 = 330\text{m}$ .

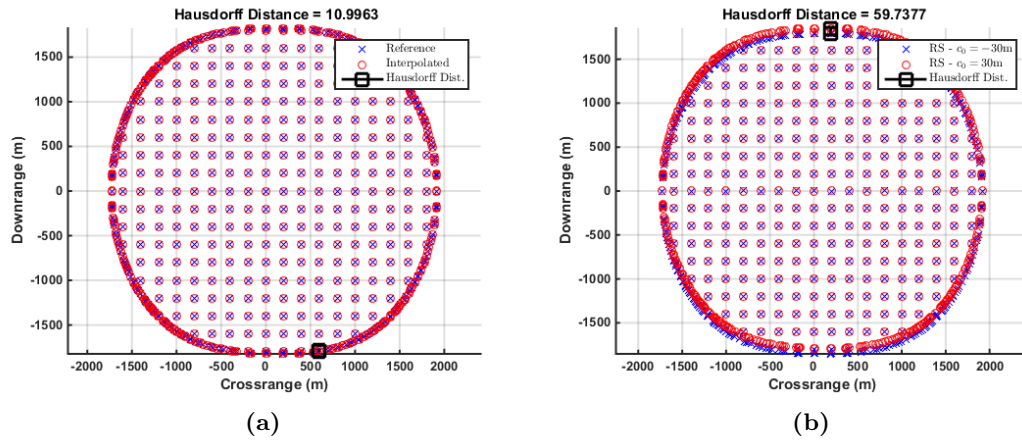


Figure B.6: Set Interpolation with  $c_0 = -30\text{m}$  and  $c_0 = 30\text{m}$ .

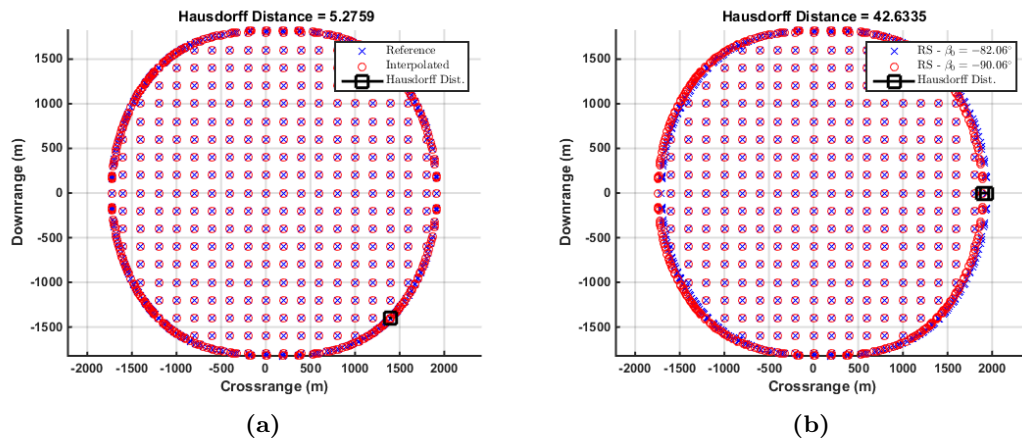


Figure B.7: Set Interpolation with  $\beta_0 = -90.06^\circ$  and  $\beta_0 = -82.06^\circ$ .

## B. ADDITIONAL REACHABLE SETS

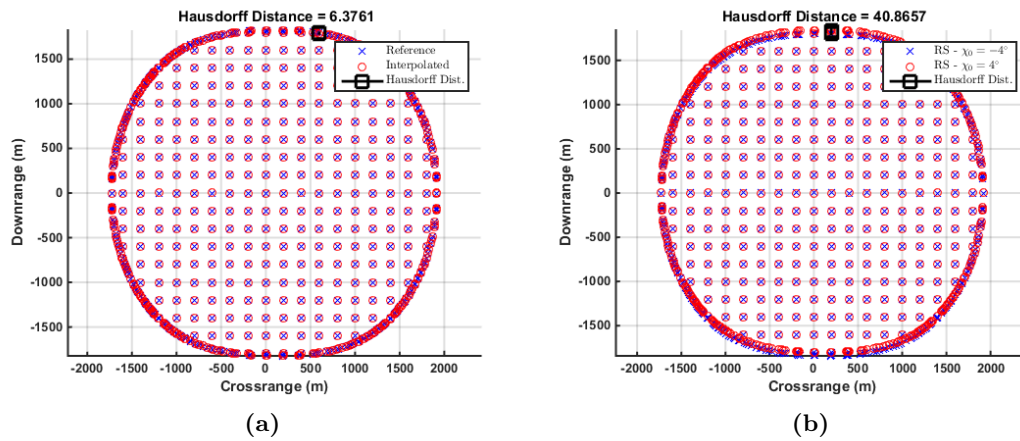


Figure B.8: Set Interpolation with  $\chi_0 = -4^\circ$  and  $\chi_0 = 4^\circ$ .

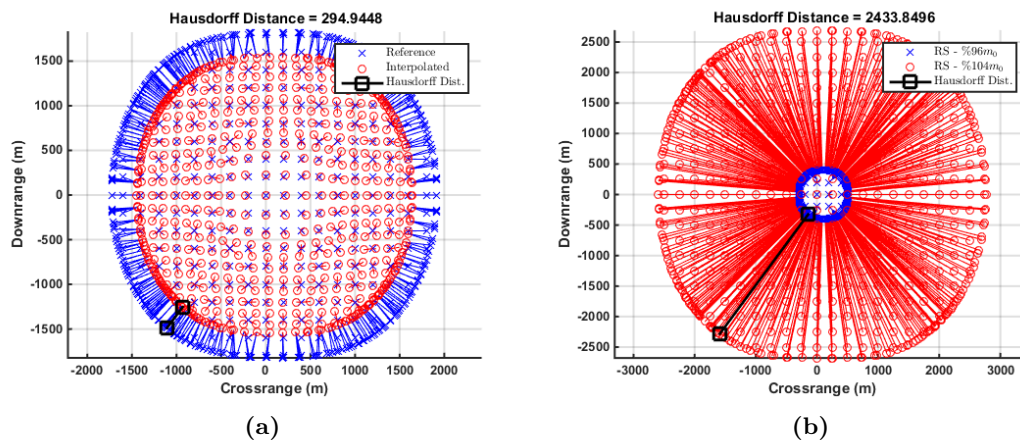
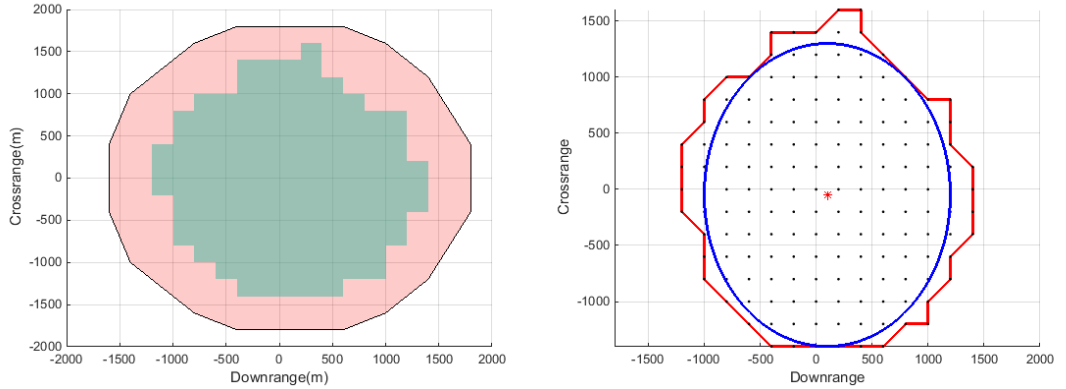


Figure B.9: Set Interpolation with  $\%96m_0$  and  $\%104m_0$ .

## B.2 Extended Monte Carlo Analysis

### Uncertainty in $\dot{d}_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $\dot{d}_0$ .

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $\dot{d}_0$ .

Figure B.10: MC Simulations with uncertainty in  $\dot{d}_0$ .

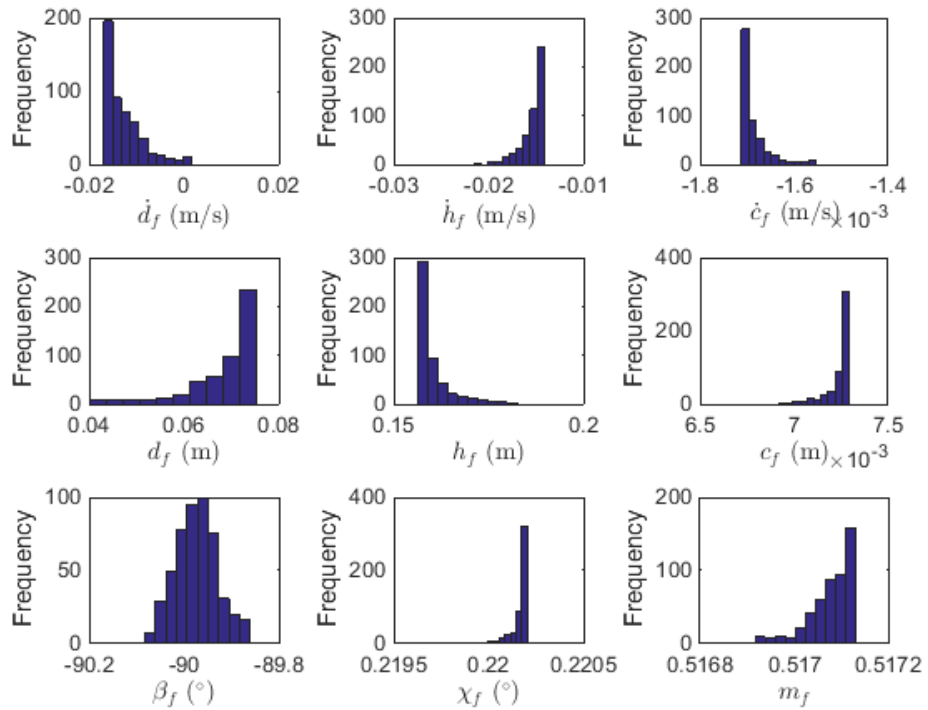
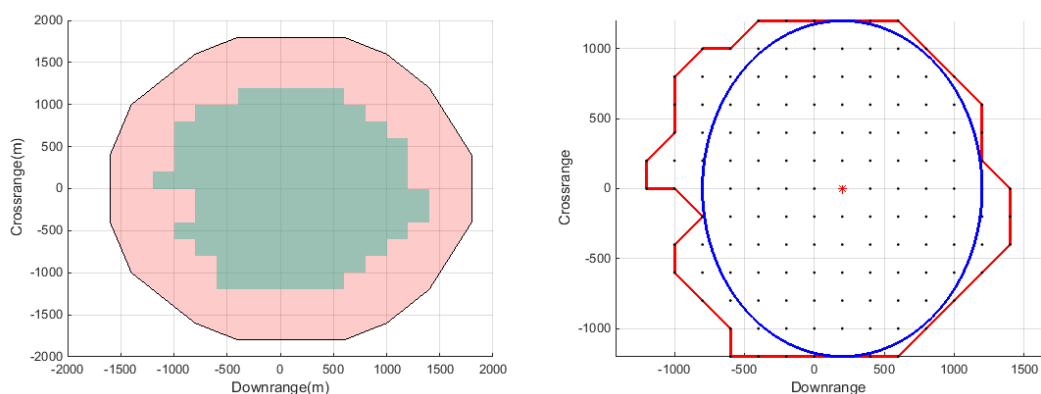


Figure B.11: Histogram of Final States with Uncertainty in  $\dot{d}_0$ .

## B. ADDITIONAL REACHABLE SETS

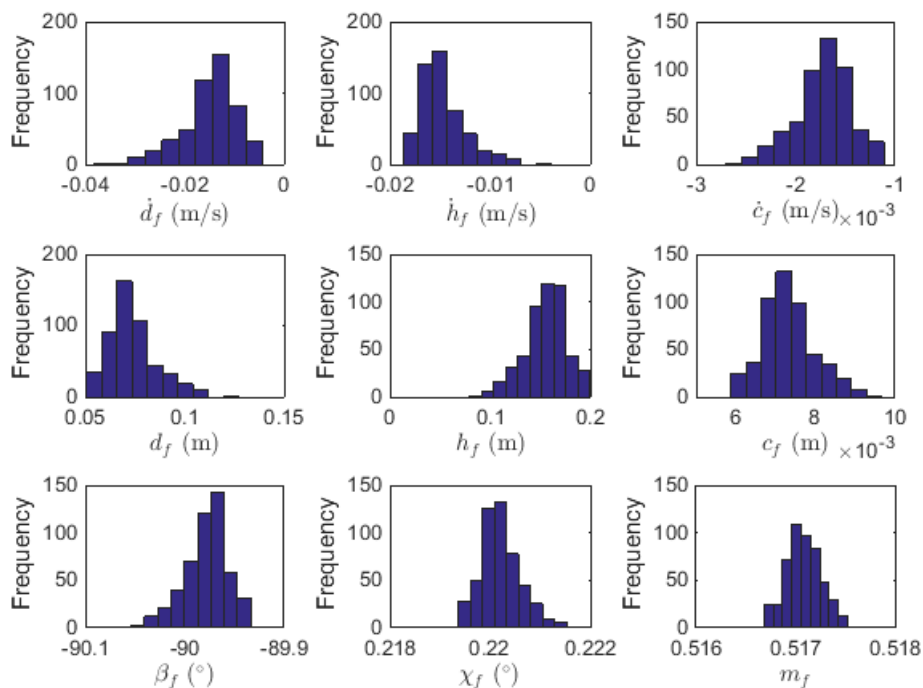
### Uncertainty in $\dot{h}_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $\dot{h}_0$ .

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $\dot{h}_0$ .

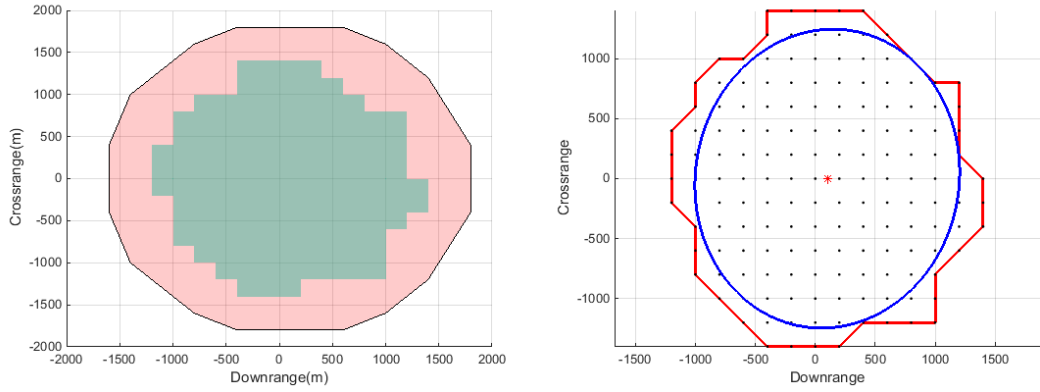
**Figure B.12:** MC Simulations with uncertainty in  $\dot{h}_0$ .



**Figure B.13:** Histogram of Final States with Uncertainty in  $\dot{h}_0$ .



Uncertainty in  $\dot{c}_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $\dot{c}_0$ .

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $\dot{c}_0$ .

Figure B.14: MC Simulations with uncertainty in  $\dot{c}_0$ .

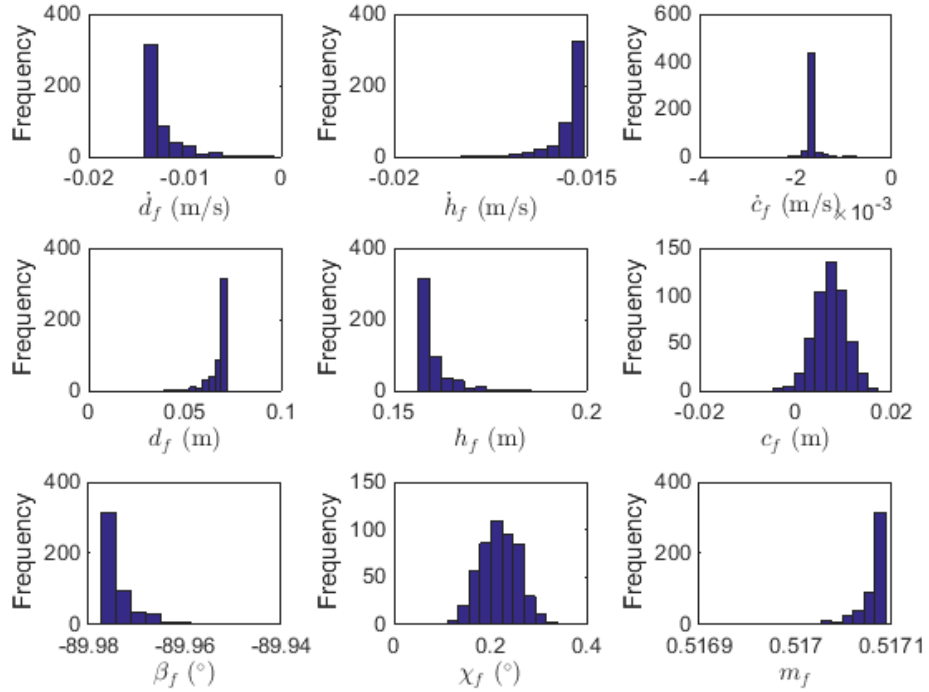
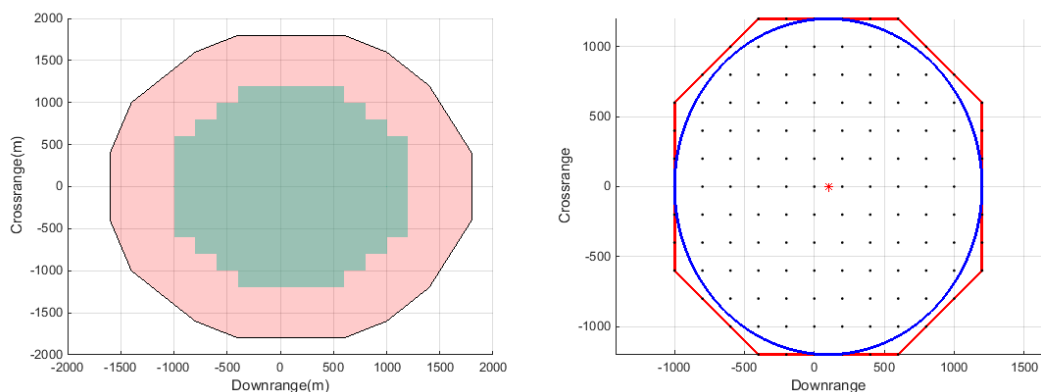


Figure B.15: Histogram of Final States with Uncertainty in  $\dot{c}_0$ .

## B. ADDITIONAL REACHABLE SETS

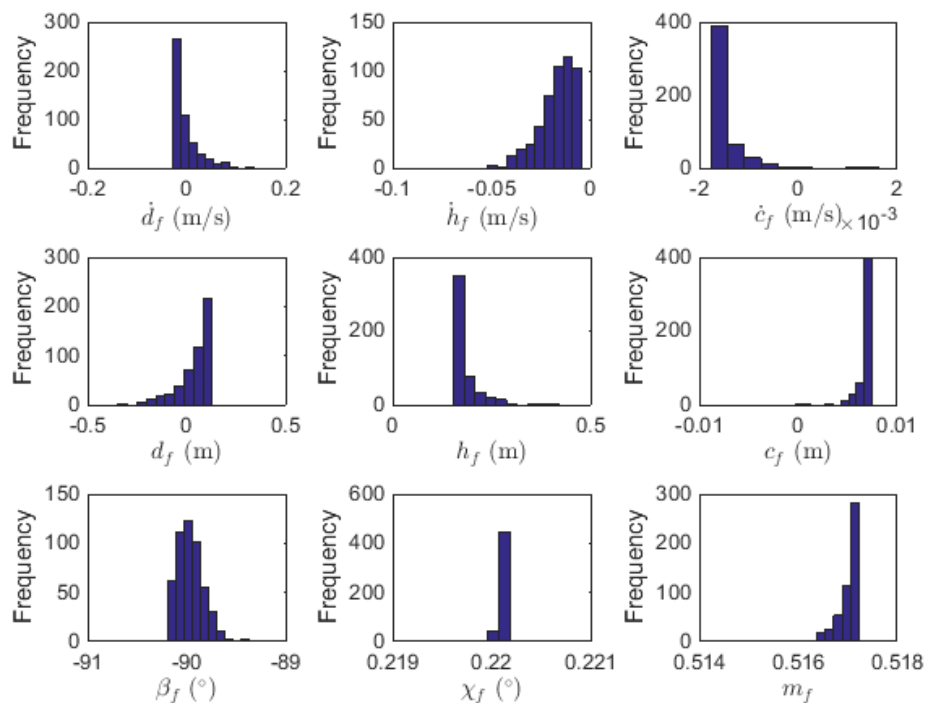
### Uncertainty in $d_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $d_0$ .

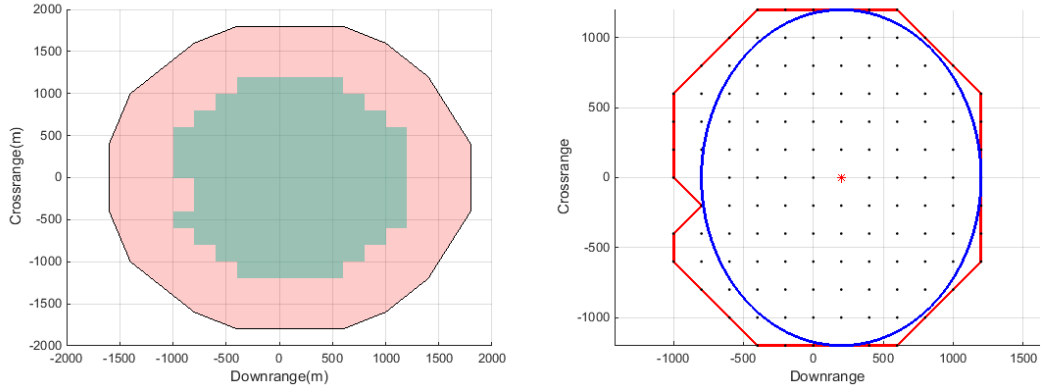
(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $d_0$ .

**Figure B.16:** MC Simulations with uncertainty in  $d_0$ .



**Figure B.17:** Histogram of Final States with Uncertainty in  $d_0$ .

Uncertainty in  $h_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $h_0$ . (b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $h_0$ .

Figure B.18: MC Simulations with uncertainty in  $h_0$ .

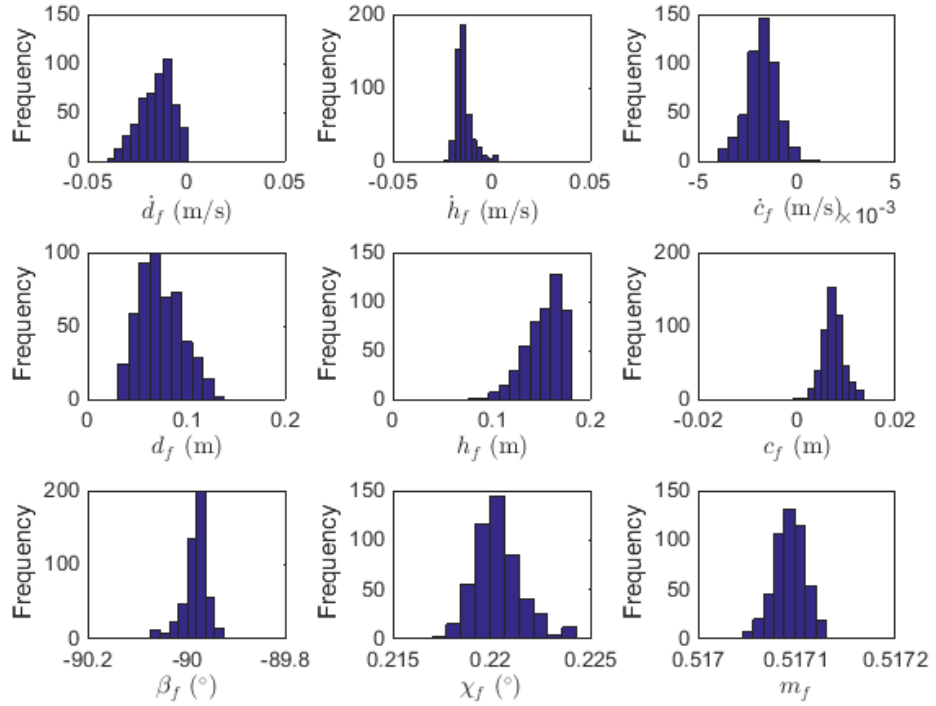
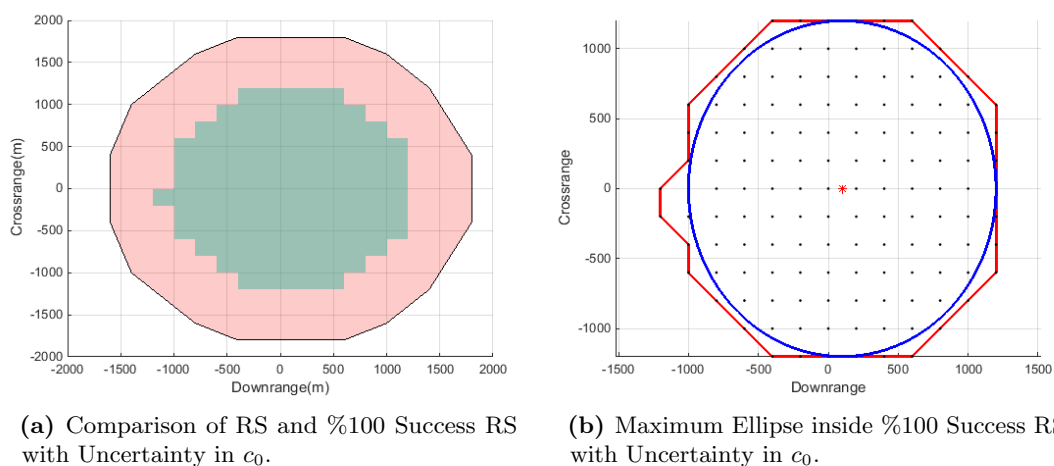


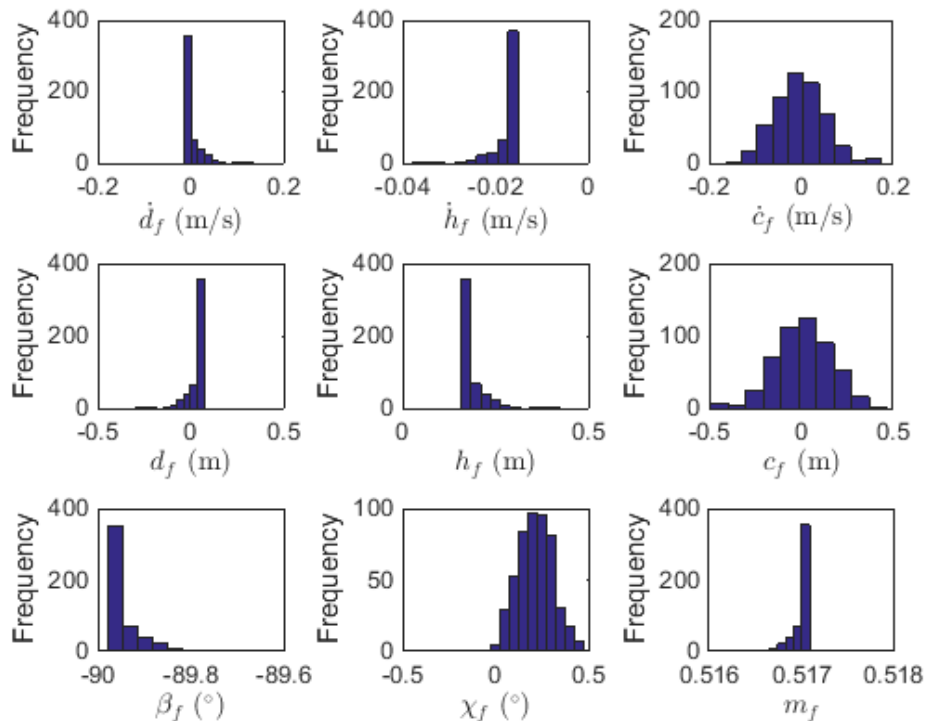
Figure B.19: Histogram of Final States with Uncertainty in  $h_0$ .

## B. ADDITIONAL REACHABLE SETS

### Uncertainty in $c_0$

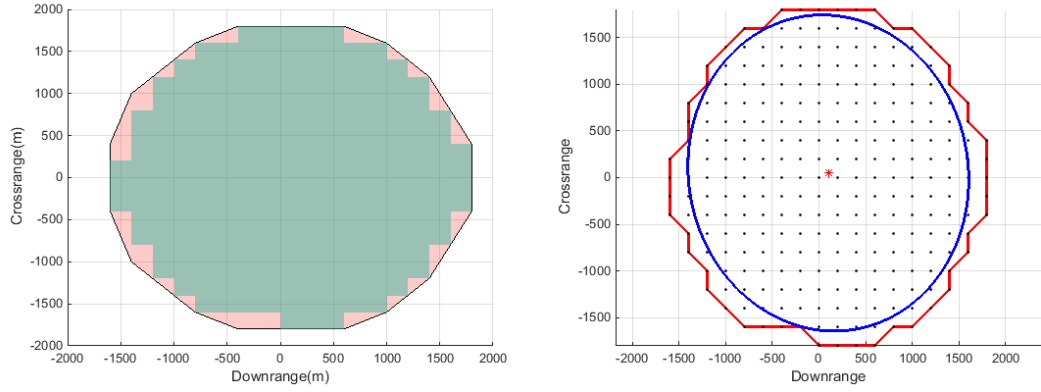


**Figure B.20:** MC Simulations with uncertainty in  $c_0$ .



**Figure B.21:** Histogram of Final States with Uncertainty in  $c_0$ .

Uncertainty in  $\beta_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $\beta_0$ .

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $\beta_0$ .

Figure B.22: MC Simulations with uncertainty in  $\beta_0$ .

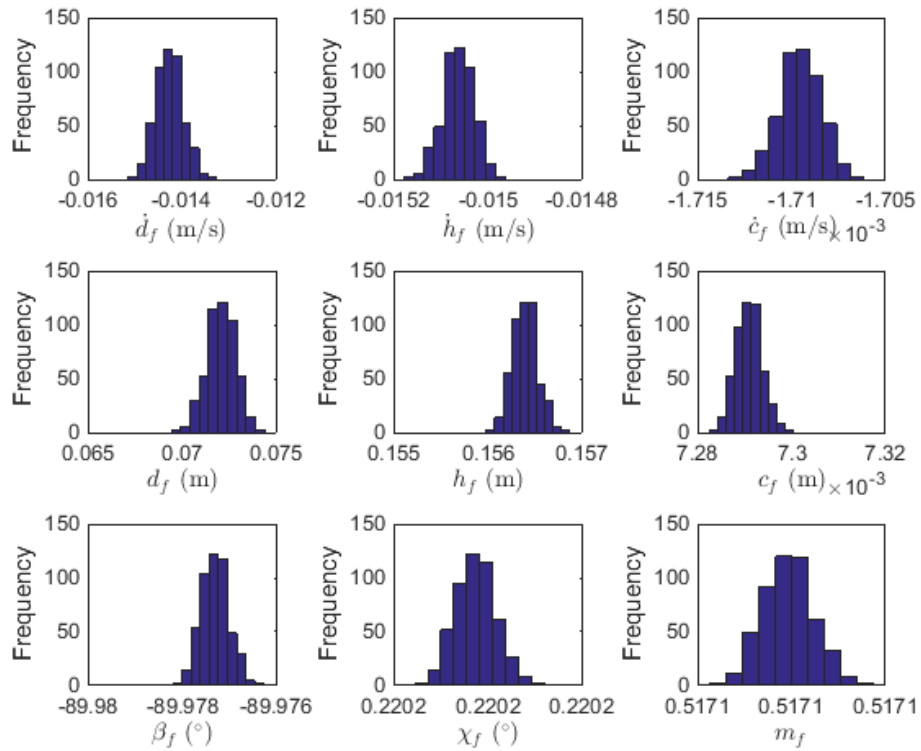
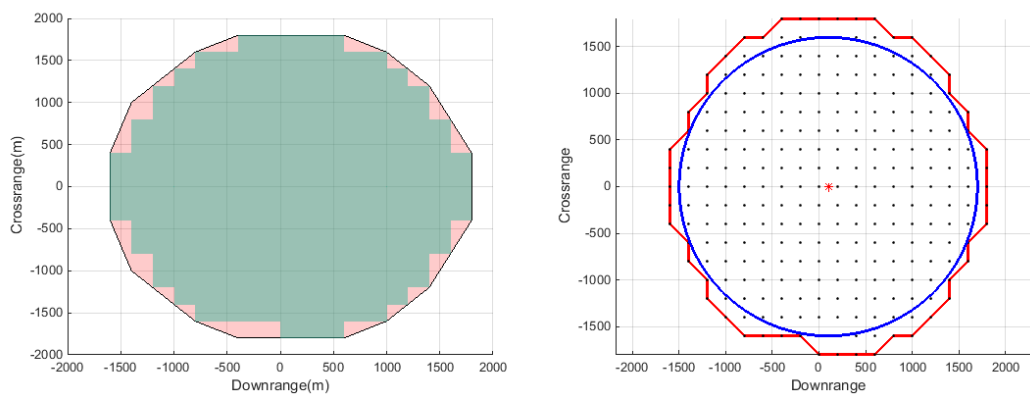


Figure B.23: Histogram of Final States with Uncertainty in  $\beta_0$ .

## B. ADDITIONAL REACHABLE SETS

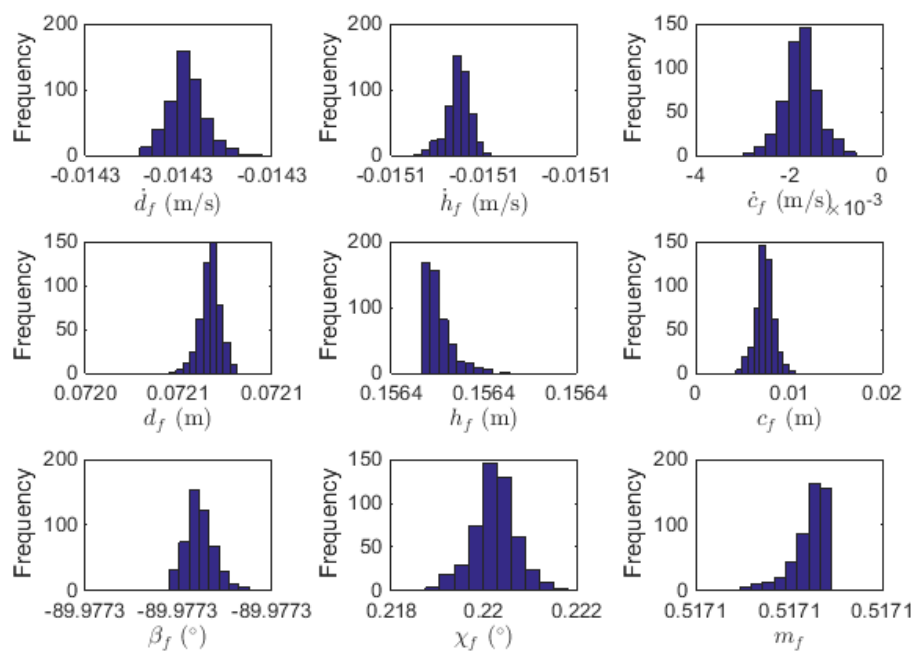
### Uncertainty in $\chi_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $\chi_0$ .

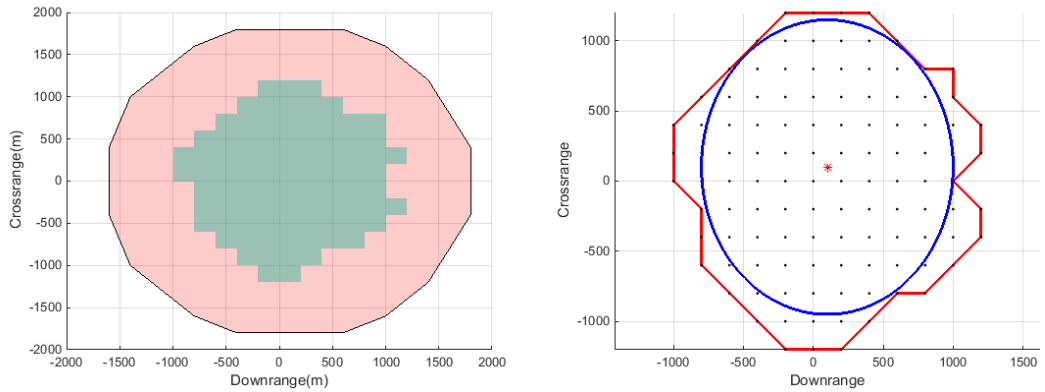
(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $\chi_0$ .

**Figure B.24:** MC Simulations with uncertainty in  $\chi_0$ .



**Figure B.25:** Histogram of Final States with Uncertainty in  $\chi_0$ .

Uncertainty in  $m_0$



(a) Comparison of RS and %100 Success RS with Uncertainty in  $m_0$ .

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in  $m_0$ .

Figure B.26: MC Simulations with uncertainty in  $m_0$ .

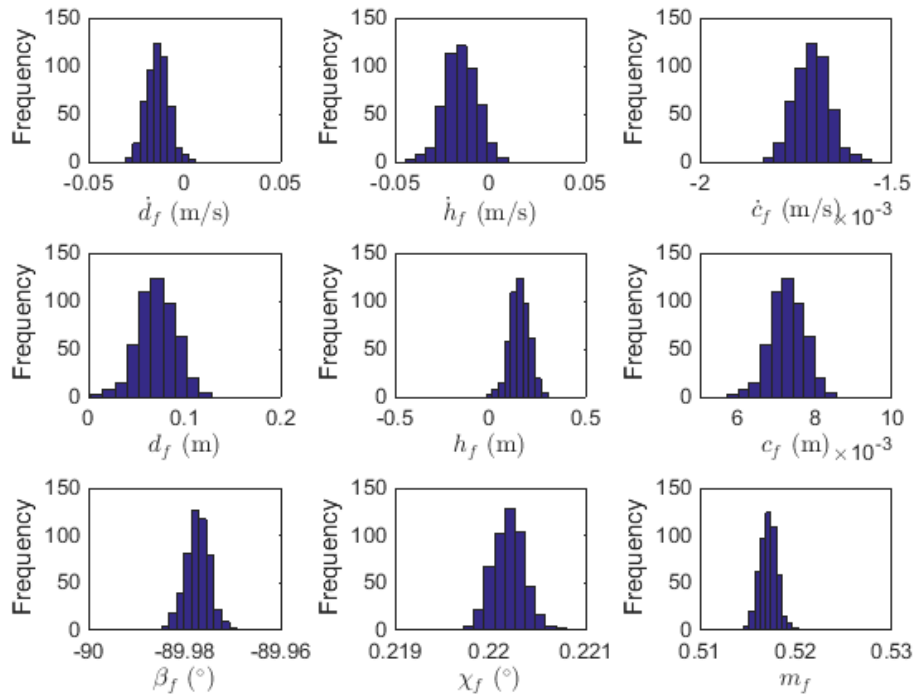


Figure B.27: Histogram of Final States with Uncertainty in  $m_0$ .

## B. ADDITIONAL REACHABLE SETS

### Uncertainty in $SFC$

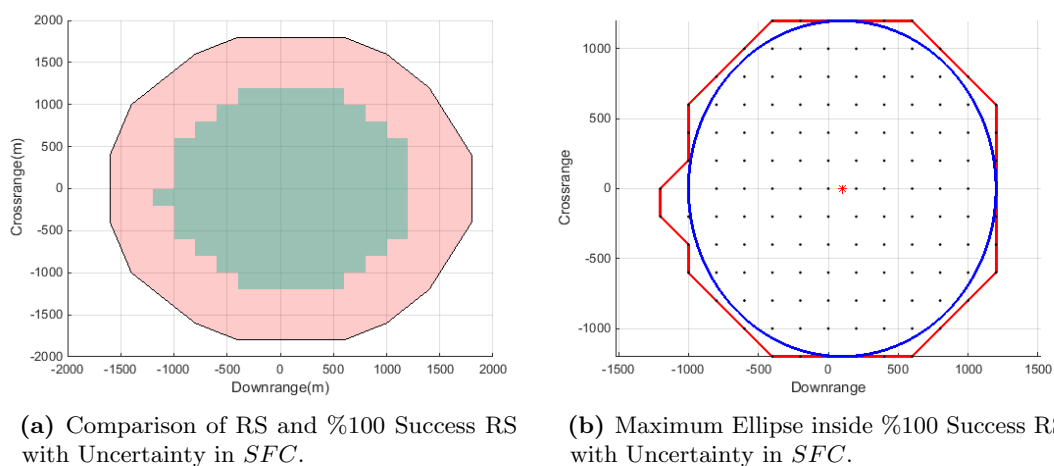


Figure B.28: MC Simulations with uncertainty in  $SFC$ .

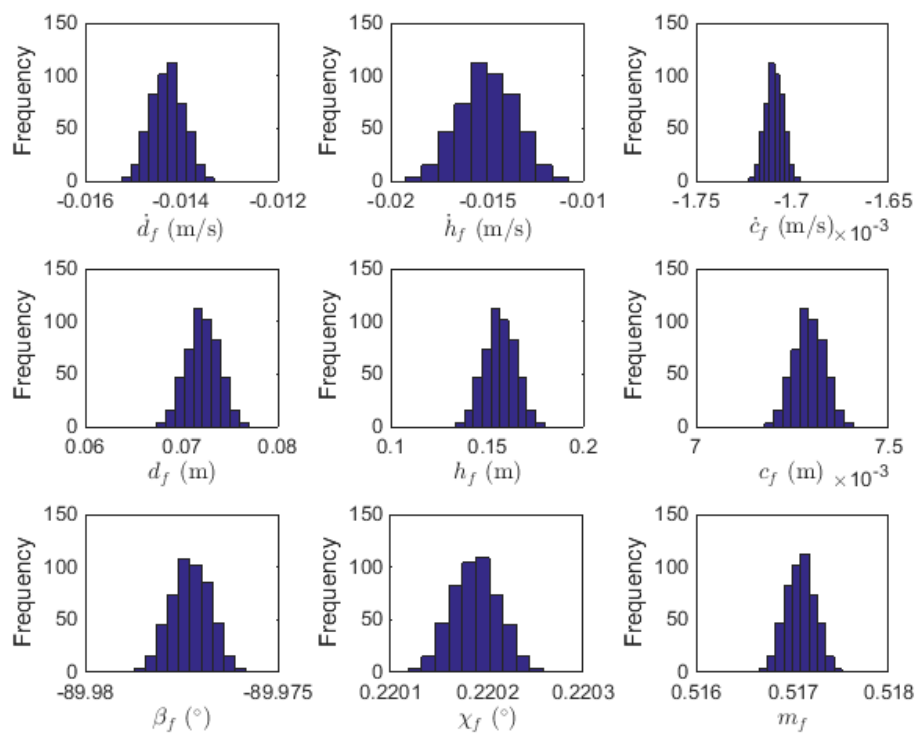
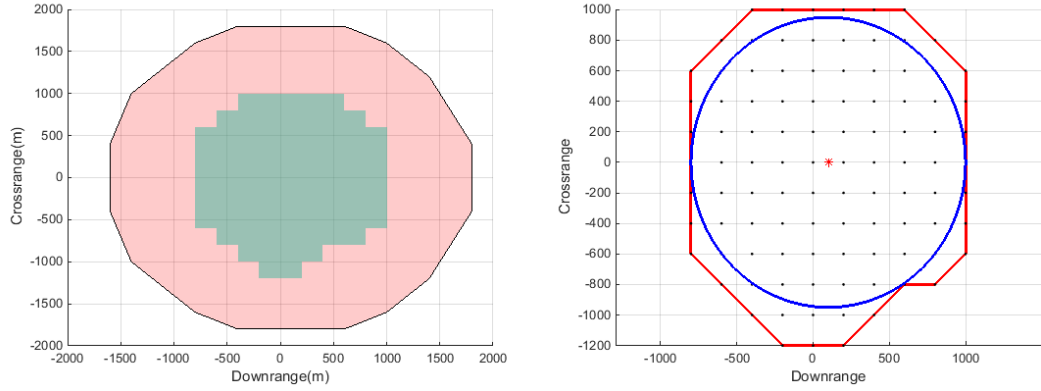


Figure B.29: Histogram of Final States with Uncertainty in  $SFC$ .



Uncertainty in all States



(a) Comparison of RS and %100 Success RS with Uncertainty in all States and Parameters.

(b) Maximum Ellipse inside %100 Success RS with Uncertainty in all States and Parameters.

Figure B.30: MC Simulations with uncertainty in all states and parameters.

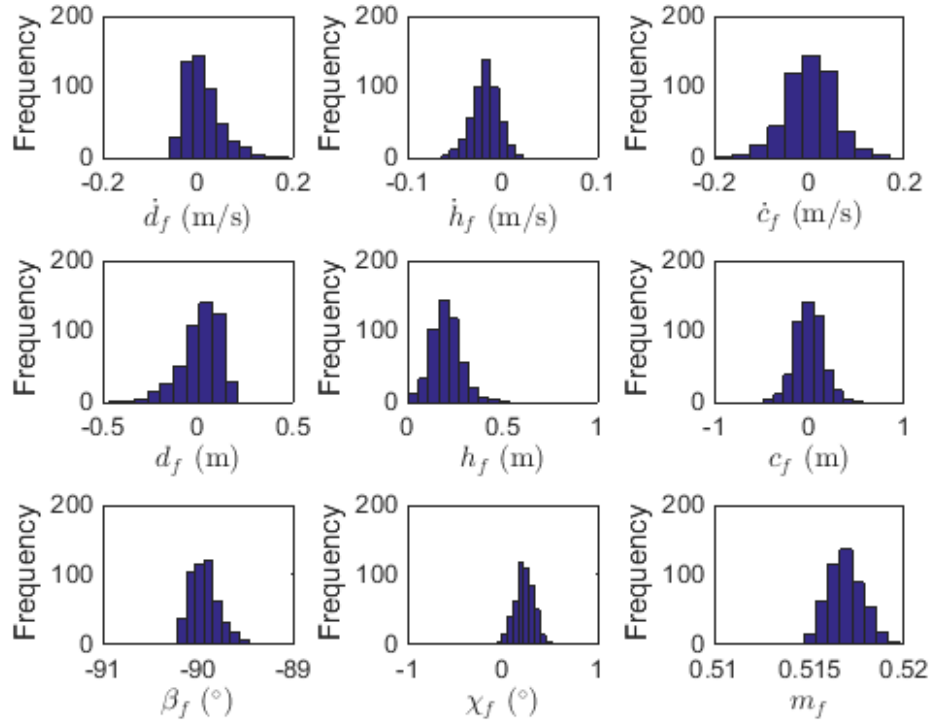


Figure B.31: Histogram of Final States with Uncertainty in all States and Parameters.