A NOVEL APPROACH TO
REACHABILITY ANALYSIS OF AERODYNAMIC INTERCEPTORS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TUĞBA BAYOĞLU AKALIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
AEROSPACE ENGINEERING

SEPTEMBER 2023

Approval of the thesis:

A NOVEL APPROACH TO REACHABILITY ANALYSIS OF AERODYNAMIC INTERCEPTORS

submitted by **Tuğba BAYOĞLU AKALIN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Serkan Özgen
Head of the Department, **Aerospace Engineering** ________________

Asst. Prof. Dr. Ali Türker Kutay
Supervisor, **Aerospace Engineering, METU** ________________

Examining Committee Members:

Assoc.Prof. Dr. Halil Ersin Söken
Aerospace Engineering, METU ________________

Asst. Prof. Dr. Ali Türker Kutay
Aerospace Engineering, METU ________________

Prof. Dr. Metin Uymaz Salamcı
Mechanical Engineering, Gazi University ________________

Prof. Dr. İlkay Yavrucuk
TUM School of Engineering and Design, Technical University of Munich ________________

Asst. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED University ________________

Date: 11.09.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Tuğba Bayoğlu Akalın

Signature:

ABSTRACT

A NOVEL APPROACH TO
REACHABILITY ANALYSIS OF AERODYNAMIC INTERCEPTORS

Bayoğlu Akalın, Tuğba
Doctor of Philosophy, Aerospace Engineering
Supervisor: Asst. Prof. Dr. Ali Türker Kutay

September 2023, 172 pages

An algorithm for reachability analysis has been developed to address the evolving challenges faced by air defense systems due to the increasing diversity of potential threats. The expansion of target sets that pose threats necessitates the development of enhanced defense mechanisms to effectively counter numerous targets, some of which are challenging to detect and track. To address these challenges, the coordinated use of interceptor kinematic capabilities to create a defended airspace shows promise.

To implement this strategy effectively, accurate forecasting of interceptor kinematic capabilities, referred to as reachable sets, is crucial. This study presents a novel reachability analysis algorithm with a specific focus on aerodynamic interceptors. The algorithm employs a directional search technique to determine reachable set boundaries along predefined search directions under various flight conditions and durations, while accounting for input constraints.

Key factors considered include energy dissipation during maneuvers, energy augmentation from thrust profiles, variable acceleration limits due to dynamic flight conditions, and autopilot dynamics. Leveraging Model Predictive Static Programming, the algorithm offers input-constrained suboptimal midcourse guidance law. This law assesses reachability, generates energy efficient trajectories, and addresses physical limits.

The resulting boundaries represent achievable minimum and maximum ranges under different conditions, aiding interceptor capability estimation. Furthermore, the research conducts a comparative analysis, highlighting the significance of input constraints in control system design. Comparing reachable sets with and without input constraints provides insights into system behavior, emphasizing the role of input constraints in guidance algorithm design.

Keywords: Reachable Set Computation, Reachability Analyses, Model Predictive Control, Optimal Control

ÖZ

AERODİNAMİK ÖNLEYİCİLERİN ULAŞILABİLİRLİK ANALİZİNE YENİ BİR YAKLAŞIM

Bayoğlu Akalın, Tuğba
Doktora, Havacılık ve Uzay Mühendisliği
Tez Yöneticisi: Dr. Öğr. Üy. Ali Türker Kutay

Eylül 2023, 172 sayfa

Potansiyel tehditlerin artan çeşitliliği nedeniyle hava savunma sistemlerinin karşı karşıya kaldığı gelişen zorlukları ele almak için erişilebilirlik analizi algoritması geliştirilmiştir. Tehdit oluşturan hedef setlerinin genişlemesi, bazılarının tespit edilmesi ve izlenmesi zor olan birçok hedefi etkili bir şekilde karşılamak için gelişmiş savunma mekanizmalarının geliştirilmesini gerektirmektedir. Bu zorlukların üstesinden gelmek için, savunulan bir hava sahası oluşturmak amacıyla önleyici kinematik yeteneklerinin koordineli kullanımı ümit vericidir.

Bu stratejiyi etkili bir şekilde uygulamak için önleyici kinematik yeteneklerinin, "ulaşılabilir küme" olarak adlandırılan, doğru bir şekilde tahmin edilmesi kritik önem taşımaktadır. Bu çalışma, özellikle aerodinamik önleyicilere odaklanan yeni bir erişebilirlik analizi algoritması sunmaktadır. Algoritma, girdi kısıtlamalarını hesaba katarak, çeşitli uçuş koşulları ve süreleri altında belirli arama yönleri boyunca erişilebilir küme sınırlarını belirlemek için yönlü arama tekniğini kullanmaktadır.

Dikkate alınan temel faktörler arasında manevralar sırasındaki enerji kaybı, itki profillerinden gelen enerji artışı, dinamik uçuş koşullarından kaynaklanan değişken ivme sınırları ve otopilot dinamikleri yer almaktadır. Model Tahminli Statik

Programlamadan yararlanan algoritma, girdi kısıtlamalı altoptimal ara safha güdüm yasası sunmaktadır. Bu yasa, erişilebilirliği değerlendirir, enerji verimli yörüngeler oluşturur ve fiziksel sınırları ele alır.

Elde edilen sınırlar, farklı koşullar altında ulaşılabilir minimum ve maksimum menzilleri temsil etmektedir ve önleyici kabiliyetinin tahmin edilmesine yardımcı olmaktadır. Ayrıca, araştırma, kontrol sistem tasarımında girdi kısıtlamalarının önemini vurgulayan karşılaştırmalı bir analiz gerçekleştirmektedir. Girdi kısıtlamalarını içeren ve içermeyen erişilebilir kümeleri karşılaştırarak, sistem davranışına dair içgörüler sunmaktadır ve güdüm algoritması tasarımındaki girdi kısıtlamalarının rolünü vurgulamaktadır.

Anahtar Kelimeler: Erişebilirlik Kümesi Hesaplaması, Erişebilirlik Analizleri,
Model Öngörülü Kontrol, Optimal Kontrol

To My Mother and My Significant Other

ACKNOWLEDGMENTS

Foremost, I am deeply indebted to my mother, Zühal Petekkaya. Her understanding, love, continuous support, and trust in my abilities were a constant source of motivation. Her sacrifices and patience have made all the difference, and I am forever grateful for her presence in my life.

I am also immensely grateful to my colleague and significant other, Gökcan Akalın. Your constant encouragement, patience, and understanding have been invaluable to me. Late night brainstorming sessions and your contribution have played a pivotal role in shaping this thesis. I cannot thank you enough for your guidance.

I want to express my gratitude to my father, Sezgin Bayoğlu, for his support and kind wishes.

I would also like to acknowledge my supervisors Ilkay Yavrucuk and Ali Türker Kutay, whose contributions and expertise have enriched this thesis. Your valuable input has played an important role in shaping the outcome of this research.

I am also grateful to my thesis committee member, Metin Salamcı, for his valuable feedback and suggestions during my study. Your recommendations have contributed to the development of this thesis.

Finally, I thank all friends, professors and family members who have supported me throughout this academic journey. Completing this thesis would not have been possible without the support, guidance, and encouragement of everyone mentioned above.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
MPSP	Model Predictive Static Programming
MPC	Model Predictive Control
PNG	Proportional Navigation Guidance
PN	Proportional Navigation
EIL	Earliest Intercept Line
MMSE	Minimum Mean Square Error
MAP	Maximum A Posteriori Probability
HPI	Highest Probability Interval
RS	Reachable Set
IC	Initial Condition

LIST OF SYMBOLS

SYMBOLS

$\vec{V}_M$	Interceptor velocity
$\mathcal{F}_I, \mathcal{F}_W$	Inertial reference frame and Wind frame
$\vec{w}_{W/I}$	Angular velocity of $\mathcal{F}_W$ with respect to $\mathcal{F}_I$
$a_{M_x}^W, a_{M_y}^W, a_{M_z}^W$	Interceptor acceleration components in $\mathcal{F}_W$
$a_{M_{y,c}}^W, a_{M_{z,c}}^W$	Interceptor acceleration command components in $\mathcal{F}_W$
$R_{M_x}, R_{M_y}, R_{M_z}$	Interceptor position components in $\mathcal{F}_I$
T	Thrust magnitude
τ	Autopilot time constant
$(\alpha_0, \beta_0), h_0, V_{M_0}$	Initial flight path angles, altitude and velocity of the interceptor
t_{go}, R	Time to go and range from initial interceptor position to desired terminal point
α, β	Angles to define orientation of $\vec{V}_M$ with respect to $\mathcal{F}_I$
α_s, β_s	Angles to define desired terminal position of the interceptor with respect to $\mathcal{F}_W$
$R_{min_{reach}}$	Minimum range reachability boundary set
$R_{max_{reach}}$	Maximum range reachability boundary set
S_{ref}	Reference area of the missile
m	Mass of the interceptor
$R(\vec{n}, \theta)$	The operation of rotating $\vec{n}$ vector around an angle θ

ϵ_y	Allowable reachability tolerance between desired terminal point and terminal point of the
i_{MPC}, i_{MPC}^{max}	Current iteration number and maximum allowable iteration number of MPP
r, r_{max}	Current iteration number and maximum allowable iteration number of Hildreth procedure
$D_K(\vec{a})$	The derivative of the vector $\vec{a}$ with respect to the frame represented by K.
$\mathcal{L}\{.\}$	Laplace transform function
$\hat{C}^{(K,L)}$	Transformation matrix from frame L to frame K.
$\bar{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$	Column matrix representation of vector $\vec{a}$
$\bar{U}^*$	Input vector history of model predictive programming utilized in first iteration
$\bar{U}$	Resultant input vector history of the model predictive programming
C_X	Total drag coefficient of the interceptor
C_{X_a}, C_{X_0}	Drag coefficient due to acceleration and base drag
h	Altitude (m)
$Temp$	Temperature (K)
ρ	Density of air (kg/m^3)
T_{SL}	Sea level temperature
ρ_{SL}	Density of air at sea level

R_g	Specific gas constant
a	Lapse rate
g	Gravitational constant

CHAPTER 1

INTRODUCTION

In recent years, the expansion of threats faced by air defense systems has presented a formidable challenge. The deployment of multiple threats with varying capabilities has become a prevalent tactic, aimed at overwhelming the capabilities of air defense systems. Additionally, the coordination among targets, uncertainties in target states, and unpredictable target maneuvers further compound the difficulties in defending specific airspace when interceptors are allocated in a one-to-one fashion. Consequently, there is an urgent need to develop new air defense capabilities that can effectively handle both single and multiple threats. To achieve this, it is essential to consider the physical limitations and kinematic capabilities of aerodynamic interceptors within the air defense strategy.

The significance of midcourse guidance for aerodynamic interceptors lies in its critical role in achieving successful target interception and ensuring an effective air defense strategy. Among the various guidance phases, midcourse guidance is particularly crucial for medium or long-range aerodynamic interceptors due to its extended duration. Throughout the midcourse phase, multiple sources of information are employed to guide the interceptor towards an optimal kinematic state that enables target acquisition by the seeker. This phase plays a crucial role in finalizing the allocation of interceptors to incoming targets, as well as ensuring their successful interception during the terminal phase. Consequently, the development of efficient midcourse strategies for each interceptor is of utmost importance.

A plethora of midcourse guidance strategies for aerodynamic interceptors can be found in the academic literature. Many of these strategies are derived from optimal control methodologies and Proportional Navigation (PN) based principals. While PN-based methods take into account target velocity and position information, they often overlook the evaluation of potential target maneuvers when designing the guidance law. On the other hand, optimal control-based guidance laws [1] primarily aim to achieve energy efficiency by minimizing total control effort during flight and maximum control command demand during the terminal phase. They are commonly employed to optimize interceptor flight parameters at the moment of interception.

However, existing applications in this field often neglect to consider if the interceptor can feasibly get to the desired destination with the available input set throughout the flight. To enhance the robustness of guidance laws against uncertainties in target information, a reachable set-based approach [5] has been proposed in recent literature. This approach takes into account not only the current trajectories of interceptors but also the potential trajectories resulting from changes in target kinematics. Consequently, this guidance law exhibits greater resilience in the face of uncertainties.

To facilitate the development of reachability-based midcourse guidance laws [5], it is crucial to conduct realistic and detailed computations of the reachable set. These computations aim to create a map of the attainable area, taking into account not only the kinematic capabilities of the aerodynamic interceptor but also the physical limitations imposed by the aircraft's design and the constraints of its actuators.

1.1 Reachability Concept

A reachable set can be defined as the set of states obtained by the application of different possible admissible control sequences from a given initial state. Reachability analysis is utilized to assess the reachable set over a specified time interval. The reachable set (admissible set of states) consists of state variables that include not only position but also other attributes such as velocity, orientation, or internal states. By focusing on reachable positions in the reachable set, refined analysis can be particularly relevant in applications where the position is the primary concern, such as path planning, obstacle avoidance, or spatial coverage. In this study, a specific focus is placed on the positions within the reachable set, which exclusively consists of positions that can be reached. The reachable zone is concerned with examining whether a designated point can ultimately be reached by a system originating from a specified point.

Figure 1 shows an example of a representative 2D reachable set (only spatial data) which is a set of positions that the interceptor can reach at t_{go} duration while following different trajectories.

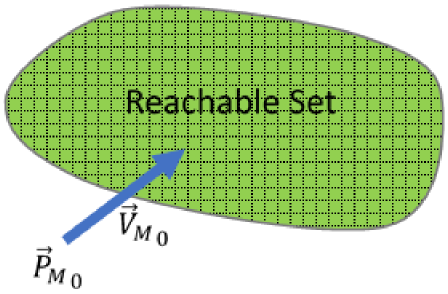


Figure 1: Reachable Set in 2D

Reachability analysis encompasses two complementary approaches: (1) Forward reachability, and (2) Backward reachability.

The aim of the forward reachability computation is to identify the final set of states at $t_f$or set of states in a time frame $0 < t < t_f$ which start from a given initial state $\vec{x}_0$ and initial time $t_0$and applying a sequence of control inputs. For example, the forward reachability analysis yields reachable sets which show the terminal target positions for a given initial condition, nonlinear system dynamics, and constraints as shown in Figure 2.

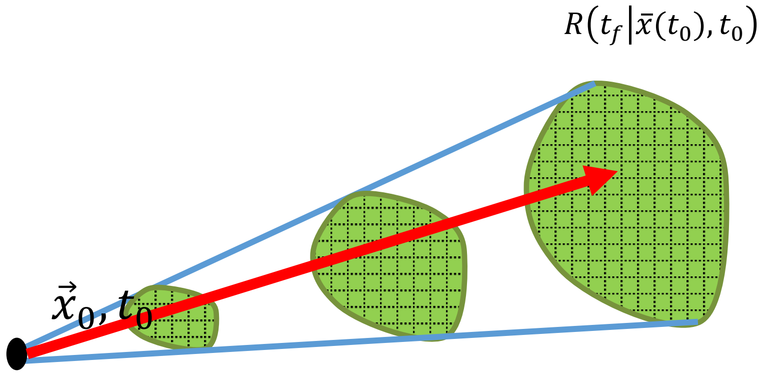


Figure 2: An illusrative forward reachable set

On the other hand, the aim of the backward reachability computation is to identify the initial set of states or previous set of states in a time frame $0 < t < t_f$ by using given target reachable set $R(t_f \mid \bar{x}(t_0), t_0)$. Similarly, for the same example, the set contains terminal target positions could be reachable with the given dynamical system if the initial state is inside the backward reachable set. Figure 3 shows the graphical representation of a backward reachable set as an example.

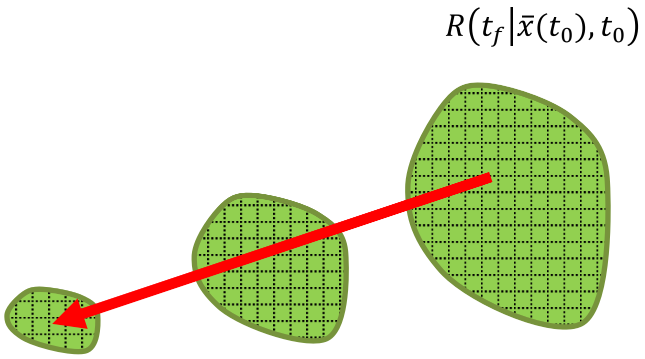


Figure 3:An illusrative backward reachable set

Numerous methodologies exist for computing reachable sets, each with its own strengths and limitations in terms of solution accuracy, computational requirements and class of the system.

One of the key factors in selecting an appropriate method is the accuracy of the solution. Different techniques offer varying levels of precision in representing the reachable sets, and the choice depends on the specific requirements of the problem at hand.

Another crucial factor is the computational power necessary for executing the method. Reachable set computations can be computationally intensive, particularly for nonlinear systems with complex constraints. Therefore, researchers must carefully evaluate the computational demands of each method and assess whether the available computational resources are sufficient for their application.

Additionally, the capability of the method to handle the specific class of systems is another key consideration. Given the focus on nonlinear dynamics in this study, it is essential to select a method that is tailored to these characteristics. Such methods should effectively capture the intricate behavior of nonlinear systems and properly account for the presence of constraints to ensure accurate and reliable reachable set computations.

This study concentrates on nonlinear system dynamics with bounded input limits, emphasizing the need for methods capable of accurately representing such nonlinear systems.

1.2 Problem and Objectives

The midcourse phase is the longest phase of the interceptor for long and medium range missions. Depending on the concept of operation, there can be several objectives for the design of the midcourse guidance algorithm. For one to one engagement case, five different objectives for the midcourse guidance can be listed as follows:

- Bring the interceptor to the best terminal handover point by using the target state supplied by external sources. The best terminal handover point is the point that ensures a successful interception at the terminal phase.
- Minimize the energy loss to maximize the warhead effectiveness and/or to gain high maneuverability for the terminal homing guidance.

- Minimize the flight time to intercept the target before attacking its objective.
- Maximize target coverage by the reachable set of pursuers at intercept.
- Protect and defend an area against the attacking target.

There have been different approaches proposed in order to develop a guidance law that satisfies the objectives listed above. The common approaches are; (1) Pursuit guidance such as pure pursuit, deviated pursuit, (2) PN based approaches such as true PNG, augmented PNG, biased PNG, (3) Optimal guidance, and (4) Artificial intelligence. The reachability-based guidance laws [5] that benefit from the reachable set analysis may not be categorized into any of aforementioned approaches. This approach has been recently studied in midcourse guidance algorithms to maximize target coverage at intercept (dynamic coverage theory) and/or defended area.

The guidance approaches listed above do not take the reachability of interceptors into account explicitly. Instead, the interceptor's speed advantage is assumed to be sufficient for ensuring reachability at the intercept point. However, there exists such a possibility that the interceptor cannot reach the target when uncertainty in target states is high or the target changes its strategy during flight; such as changing flight speed and heading. Moreover, coordination between numerous targets can decrease the possibility of an effective response by the air defense system. As a result, different from the traditional cooperative guidance approaches, application of dynamic coverage theory [18], [25], [26] for the cooperative guidance problem is being investigated by researchers. The objective of such an approach is to maximize the coverage of the region of possible target locations in the reachable set of the pursuer.

Dynamic coverage theory is proposed to maximize target coverage by the reachable set of pursuers at intercept. This approach is adapted if the target states' uncertainty is high and accurate target position information is not available.

Figure 4 illustrates an example of reachability-based air defense scenario. Each target can reach different positions at a certain duration depending on its maneuver, speed, and heading. Therefore, interceptors' reachable set should be able to cover possible target positions at the interception in the reachability-based air defense concept.

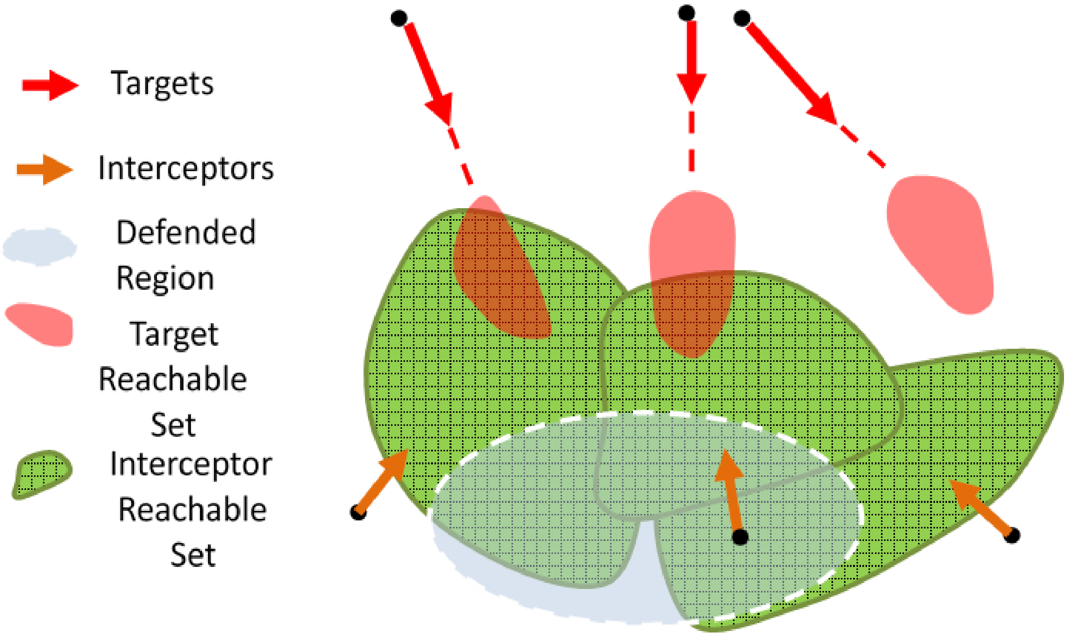


Figure 4: Example of Dynamic Coverage for Multiple Interceptors Against Multiple Targets

Apart from dynamic coverage theory, the area defense approach [11], [12], which is also different from the traditional guidance approaches, is proposed to guarantee interceptors to reach incoming targets before they can reach the defending area. This approach also utilizes reachable set of pursuers at intercept. The earliest intercept line (EIL) [20], which is computed by using the reachable set of interceptors and target is utilized to defend a given area. Figure 5 illustrates an example of intercept line computed from reachability set of interceptor and target and Figure 6 shows an example of intercept line control concept. Target and interceptor reachability set should be developed by considering different flight conditions such as different speed, maneuver, heading and by considering different time of flight durations for intercept line control concept.

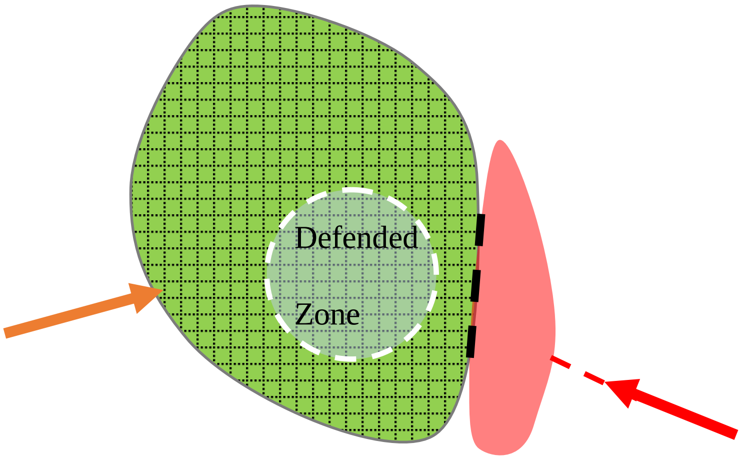


Figure 5: Intercept Line Developed From Reachability Sets

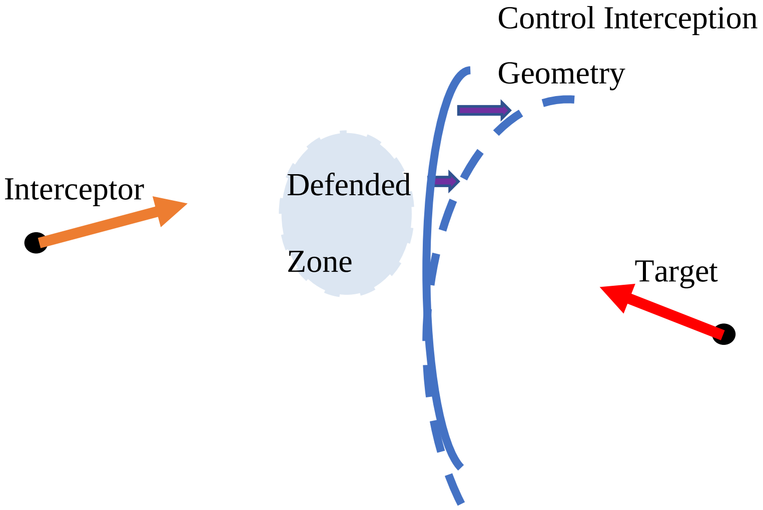


Figure 6: Example of Intercept Line Control

The traditional approach to the development of an interceptor guidance system is to design estimator, guidance algorithm and autopilot. It must be acknowledged that this kind of design approach has been successfully applied to various interceptor guidance systems for decades. However, the steady increase in performance requirements and the tendency to reduce unit costs for each new generation of interceptors leads to more demanding designs and continues to motivate researchers for more efficient guidance algorithms. The reachability-based guidance system in an interceptor can be described as shown in Figure 7: The traditional guidance system of an interceptor requires designing an estimator, guidance algorithm and autopilot. The reachability-based guidance system requires additional items: (1) Reachability Analysis Module and (2) Reachability Set Database.

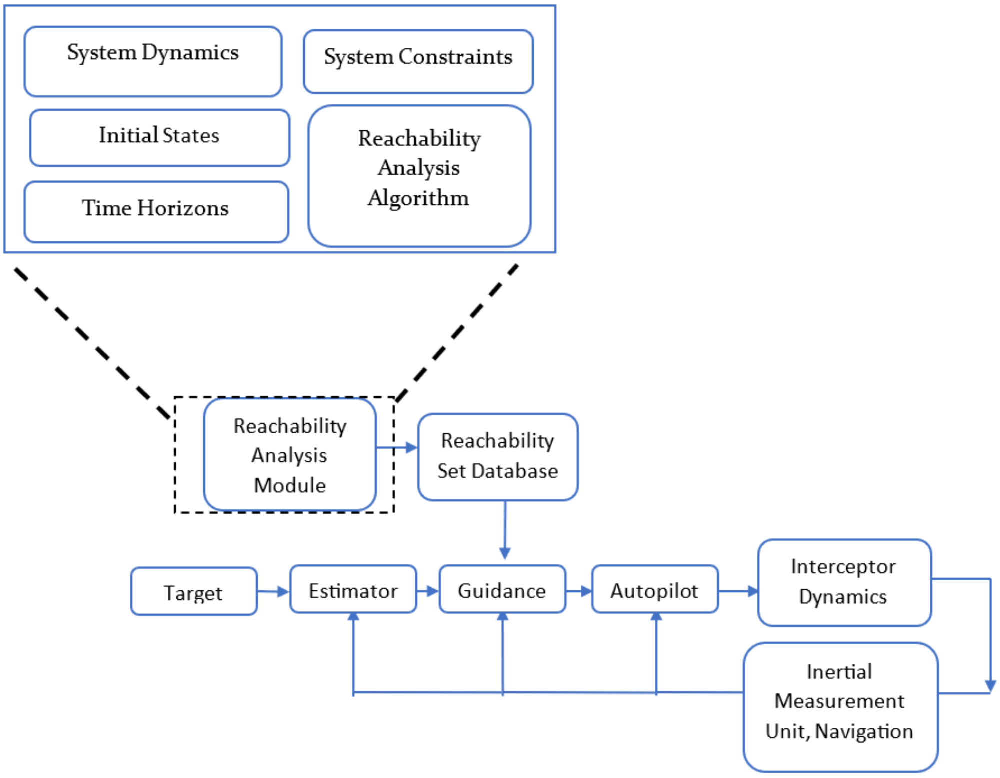

System Dynamics
System Constraints
Initial States
Reachability Analysis Algorithm
Time Horizons
Reachability Analysis Module
Reachability Set Database
Target
Estimator
Guidance
Autopilot
Interceptor Dynamics
Inertial Measurement Unit, Navigation

Figure 7: Reachability Based Guidance System

The Reachability Analysis Module is responsible for performing reachability analysis computations based on the system dynamics and constraints. It calculates the reachable sets and determines the set of achievable states for the interceptor. Reachability Analysis Module requires information regarding system dynamics, initial states, system constraints and time horizon. This data is utilized in reachability analysis algorithm to compute reachability information for the aerodynamic interceptor. The computed reachable sets for different initial conditions and constraints are stored in a database. This database serves as a reference for the guidance system to access the kinematic capabilities of the interceptor during the midcourse phase.

The reachability-based guidance approaches discussed in literature have been predominantly based on certain assumptions that neglect important factors such as speed variation of the interceptor and target due to aerodynamic drag and thrust, as well as system nonlinearities including acceleration limits. However, these assumptions can result in unacceptable interception lines and reachable sets, particularly in real-world applications where constant speed may not be satisfied or there are limitations in the system actuators. Hence, it becomes crucial to ensure realistic computations of reachable sets in the design of reachability-based midcourse guidance laws.

By addressing the aforementioned challenges, the computation of reachable sets offers valuable insights to the guidance and control algorithm, serving as a tool for various aspects such as mission design, target allocation, and performance assessment. To address the need for accurate and practical reachability analysis, this research focuses on the development of a reachability analysis algorithm. This algorithm aims to provide answers to key research questions that arise in the context of reachability-based midcourse guidance:

1. Theoretical Maximum and Minimum Reachable Set Boundaries: What are the maximum and minimum reachable set boundaries that can be achieved by the aerodynamic interceptor considering factors such as speed variation and system limitations?
2. Effect of Parameter Variations: How do variations in system parameters impact the shape and boundaries of reachable set?
3. Impact of Input Limits: How does the presence of limitations on system inputs, including acceleration limits, influence the characteristics of the reachable set?

The outcomes derived from the reachability analysis algorithm serve various purposes. Reachable set computations are conducted for all possible initial conditions, providing insights into the kinematic capabilities of the aerodynamic interceptor under different flight conditions. By satisfying defined performance criteria, calculation of minimum and maximum range reachability boundaries based on specified initial condition and search direction.

Importantly, the reachability analysis algorithm takes into consideration the limitations on system inputs. This aspect plays a crucial role in generating accurate and reliable reachable sets. By incorporating constraints on the system's inputs, such as acceleration limits, the algorithm ensures that the reachable set reflects the practical limitations of the interceptor's capabilities.

It is important to note that the reachability boundaries, representing the interceptor's kinematic capabilities, are assumed to be generated offline. These boundaries are then stored in a database embedded within the guidance computer for subsequent utilization in the guidance approach.

1.3 Scope and Principal Contributions

In this study, the reachable set computation approach for a nonlinear system is presented. The focus is on obtaining kinematic capabilities of an aerodynamic interceptor for reachability based guidance approaches.

This work has several contributions for both to the guidance and control fields. Basically, it derives a nonlinear dynamical model for the aerodynamic interceptor, where variations in velocity (due to aerodynamic drag and thrust) and acceleration limits are not disregarded. In addition, it develops a guidance algorithm for the given system dynamics of an aerodynamic interceptor to find feasible trajectories using optimal control. Model predictive static programming technique has been used to propose effective input constrained suboptimal midcourse guidance laws for engaging incoming targets. Apart from being energy efficient by minimizing the lateral acceleration demands throughout the trajectory, it enforces constraints on input parameters. The handling of inequality constraints pertaining to the input vector is achieved through the utilization of Hildreth's procedure [10]. These constraints are specifically defined based on the acceleration limits of the aerodynamic vehicle.

A novel method is introduced to create reachable sets for the nonlinear system. The problem of reachable set computation is solved by constrained model predictive static programming with different desired final conditions. By considering various flight conditions and durations, the boundaries of the reachable set in terms of minimum and maximum ranges are determined.

To illustrate the reachable set calculation procedure for the lateral plane, Figure 8. demonstrates an example for reachable set calculation procedure. The minimum and maximum boundary points of the reachable set for a specific search direction are calculated using a directional search approach, which checks the reachability at each calculation step. Notably, only one initial reachable position and its corresponding input history are required for each search direction. The reuse of previous terminal points as the initial estimate for subsequent points eliminates the need for predefining approximate geometries or defining grid point positions in the initial feasible set for reachability set computation.

Furthermore, the algorithm employed in this study leverages the reuse of optimal control information from previous terminal points, leading to improved computational efficiency by minimizing redundant calculations. This utilization of prior knowledge allows for a seamless integration of complex state constraints and boundary conditions, enabling more realistic and accurate modeling of the system.

Another noteworthy advantage of the proposed method is the elimination of cumbersome set operations, such as Minkowski sum [39] and convex hull computations, which are common in other reachability analysis approaches. By circumventing these complex set operations, the proposed method streamlines the computational process and mitigates the potential propagation of approximation errors.

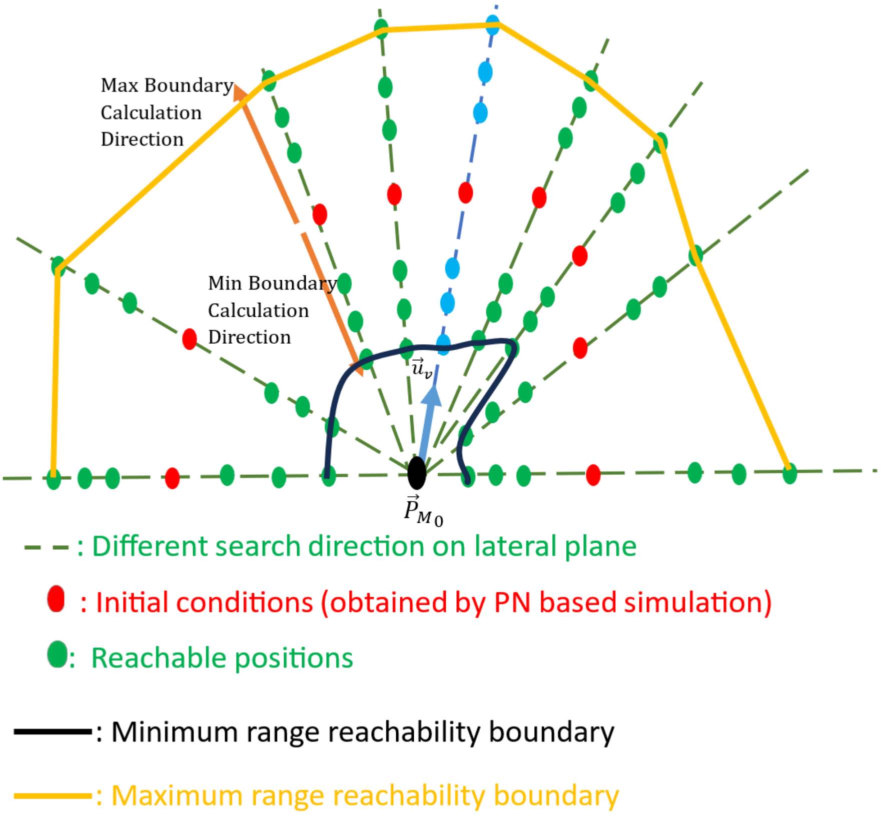

Max Boundary Calculation Direction
Min Boundary Calculation Direction
$\vec{u}_v$
$\vec{P}_{M0}$
: Different search direction on lateral plane
: Initial conditions (obtained by PN based simulation)
: Reachable positions
: Minimum range reachability boundary
: Maximum range reachability boundary

Figure 8: Reachable Set Boundary Calculation Procedure

A more realistic reachable set calculation is performed by considering the inclusion of autopilot dynamics with bounded acceleration responses. Additionally, a minimum total control command effort criterion is enforced to ensure the attainment of desired final conditions.

The effect of input constraint on the kinematic capabilities of the interceptor is explored by comparing the reachable sets obtained by input constrained and unconstrained cases. This analysis aims to assess the implications of input constraints on the interceptor's performance and capabilities. Additionally, the effect of initial state parameters on the reachability boundaries are investigated.

To ensure the robustness of the developed algorithm in the presence of parameter variations, a sensitivity analysis is conducted. Specifically, reachable sets for an aerodynamic interceptor are obtained while considering variations in system parameters.

During this study, the following tools were developed for creating and analyzing reachable sets:

- Guidance Algorithm: A guidance algorithm was developed specifically tailored to the system dynamics of an aerodynamic interceptor. This algorithm utilized optimal control techniques to find feasible trajectories, taking into account the constraints and objectives of the system.
- Reachable Set Computation Tools in Matlab: To perform the computation of reachable sets, a set of tools were developed using Matlab. These tools provided a platform for realistic reachable set computation.
- Comparative Analysis of Reachable Sets: In order to investigate the impact of input constraints on the kinematic capabilities of the interceptor, a comparative analysis of reachable sets was conducted. This analysis involved comparing the reachable sets derived from scenarios with input constraints to those without constraints. By quantifying the differences between these sets, the study was able to evaluate the influence of input limitations on the achievable trajectories and maneuverability of the interceptor.
- Sensitivity Analysis Tool for Reachable Set Computation: a sensitivity analysis tool was developed to investigate the impact of parameter variations on the computation of reachable sets. This tool enabled the incorporation of variations in the system parameters. By systematically varying the parameter values and computing the corresponding reachable sets, the sensitivity analysis tool provided insights into the sensitivity of the reachable sets to changes in the parameter values.

1.4 Outline of Thesis

The structure of this dissertation is organized as follows.

1) Introduction
2) Literature Survey
3) Example Problem (Nonlinear Dynamical Model for Aerodynamic Interceptor)
4) Guidance Algorithm Development
    a. Suboptimal Midcource Guidance with Terminal Position Constraint
    b. Suboptimal Midcource Guidance with Terminal Position Constraint and Bounded Acceleration
5) Reachable Set Analysis Tool Development
6) Set Computations
7) Sensitivity Analysis
8) Conclusion

The contents of each items listed above are elaborated as follows:

Chapter 1 provides background information for reachability based guidance systems and reachability concept. The importance of reachability analysis in guidance and control algorithm development is discussed. Challenges and limitations in traditional reachability analysis approaches and the need for an enhanced reachable set algorithm is highlighted.

Chapter 2 provides an theoretical foundations for reachability analysis principals and concepts. Key concepts such as reachable sets, system dynamics, constraints, and optimization techniques are explained.

Chapter 3 provides a comprehensive exposition of the nonlinear dynamic model employed for the representation of an aerodynamic interceptor.

In Chapter 4, the focus of the study shifts towards the design and implementation of the guidance algorithm utilized within the Reachability Analysis Algorithm. This algorithm serves as a pivotal component in the generation of reachable sets for an aerodynamic interceptor. To accomplish this objective, the concept of constrained model predictive static programming approach is employed. In the pursuit of enhancing the capabilities of the model predictive static programming technique, the incorporation of constraints on the control variable becomes paramount. This step recognizes the significance of accounting for input limitations imposed on the interceptor, ensuring a more realistic representation of its behavior. Consequently, a departure from the direct utilization of the equations associated with model predictive static programming becomes necessary. Instead, meticulous derivations are carried out to integrate the core principles of model predictive static programming with the consideration of input constraints. Moreover, comprehensive exposition of the Hildreth procedure [10], which is implemented as a vital component in solving the input-constrained model predictive static programming approach is provided. This chapter elucidates the underlying principles and algorithmic details of the Hildreth procedure [10], shedding light on its practical application within the context of the guidance algorithm.

Chapter 5 shows a comprehensive account of the reachable set computation tool developed as part of the Reachability Analysis Algorithm. This chapter serves as a detailed explanation and demonstration of the algorithm employed in generating reachable sets for the aerodynamic interceptor.

Chapter 6 provides a thorough analysis of the impact of input constraints on the kinematic capabilities of the interceptor. It explores how the inclusion of constraints on the control variables, such as maximum acceleration or control effort, influences the reachable set. It analyzes the changes in the reachable set boundaries, shapes, and coverage due to the presence of input constraints.

Chapter 7 of the study focuses on sensitivity analysis. The objective of this chapter is to assess the impact of parameter variations on the computed reachable sets. The sensitivity analysis involves varying the system parameters within specified ranges and analyzing the corresponding changes in the reachable sets. By systematically investigating the effects of these variations, the study aims to identify critical parameters and their influence on the reachable set.

Chapter 8 provides a brief summary of the design and implementation of the Reachability Analysis Algorithm to highlight the key points and identify the remaining challenges. A discussion of the algorithm’s contributions to the field of guidance systems is carried out. Then, future research directions and potential enhancements for the reachability set algorithm are discussed.

CHAPTER 2

LITERATURE REVIEW

Various methods have been developed for computing reachable sets in the existing literature. To comprehend the distinctions between these methods, it is important to establish a proper definition for reachability analysis. Reachability analysis involves the computation of the reach set, aiming to address requests such as:

1) Determining whether the reach set and a given target set intersect at a specified time.

2) Finding feasible initial conditions and controls that steer the system from a given initial condition to a reachable state within a specified time.

3) Analyzing the impact of input constraints, such as limitations on control inputs, on the size and shape of the reach set.

4) Visualizing the projection of the reach set onto designated 2 or 3 dimensional subspaces.

By addressing these objectives, reachability analysis contributes to the comprehensive understanding of controllers for dynamical systems.

2.1 Overview of Reachability Analysis Principles and Concepts

The principles of reachability analysis revolve around the understanding of system dynamics and constraints. Mathematical models, such as differential equations or discrete-time equations, are used to describe the evolution of the system over time. By analyzing these equations using numerical or analytical methods, researchers can determine the reachable states and trajectories of the system.

Modelling the dynamics of the system is crucial for reachability analysis. The system's behavior, including its nonlinearities and time-varying properties must be properly captured in the mathematical model used for reachability analysis.

Input constraints are another essential consideration in reachability analysis. These constraints limit the range of inputs that can be applied to the system, taking into account physical limitations, actuator capabilities, or design specifications. Incorporating input constraints in the analysis provides a more accurate representation of the system's behavior and helps identify the feasible states and trajectories.

Sensitivity is another crucial concept in reachability analysis. Real-world systems are often subject to parameter variations or disturbances. These variations can significantly impact the reachable set of the system.

Reachability analysis also considers performance specifications. Performance criteria measures the quality of the reachable set. Examples of performance metrics include minimizing the energy consumption, maximizing the covered area, or optimizing a specific objective function.

Various computational techniques are utilized in reachability analysis, including numerical methods and optimization algorithms. Careful consideration must be given to the choice of computational methods and algorithms to ensure efficient and accurate computation of reachable sets.

The results of reachability analysis provide insights into system behavior, performance limits, and the design of control strategies. They help to assess the feasibility of reaching desired states, and evaluating the impact of input constraints.

2.1.1 Reachability

Reachability refers to the property of a system to reach a certain state or set of states within a given time frame by applying admissable control inputs. It characterizes the system's ability to transition from an initial state to a desired or specified target state under certain constraints. The mathematical expression for reachability can be defined as:

$$\bar{x}\left(t_f\right)_{desired} \in R\left(t_f\middle|\bar{x}(t_0), t_0\right) \qquad \text{Eq. 1}$$

$\bar{x}(t_0)$ is the initial state of the system, $\bar{x}\left(t_f\right)_{desired}$is the desired state of the system, $t_0$and t_f are the initial and target time respectively. $R\left(t_f\middle|\bar{x}(t_0), t_0\right)$ represents the set of states that can be reached from the initial state $\bar{x}(t_0)$ within the time interval $[t_0, t_f]$ by applying admissable control inputs

2.1.2 Reachable Set

The reachable set is the set of all states that can be reached by a system from a given initial state under a set of admissible control inputs within a specified time frame. It represents the set of possible states that the system can occupy.

$$R(T|\bar{x}(t_0), t_0) = \{\bar{x}(t)|\bar{x}(t_0), \forall u(t) \in U, 0 \leq t \leq T\} \qquad \text{Eq. 2}$$

$R(T|\bar{x}(t_0), t_0)$ is the reachable set at time T, given the initial state $\bar{x}(t_0)$ and admissable control inputs U. In this study, the positions within the reachable set are specifically highlighted, with an exclusive composition of positions that can be reached.

2.1.3 Reachable Set Analysis

Reachable set analysis is a basic technique used to investigate and understand the capabilities of a given system. It involves the computation and analysis of the reachable sets of a system, which represent the set of all states computed by considering systems dynamics, constraints, and other relevant factors. The analysis may include creating algorithms for the computation of reachable sets, evaluating the shape of the reachable sets, and assessing their appropriateness for particular objectives such as traffic management or obstacle avoidance. In the scope of this research, reachable set analysis is used for examination of whether a system, originating from a specified point, can eventually reach a designated point.

2.2 Review of Relevant Techniques for Computation of Reachable Sets

Reachability analysis is a valuable tool applied to various real-world problems. It provides valuable insights into system behavior, control design, and decision-making in various real-world applications. Some of the applications in real world problems can be listed as follows:

- Aerospace and Aviation [45], [46]
- Guidance and control system design for interceptors [38]
- Safe landing problems for spacecrafts [37]
- Robotics [43]
- Traffic Flow [44]

The computation of reachable sets can pose several challenges and problems.Some of the common issues encountered in the computation of the reachable sets are:

Curse of Dimensionality: Reachable set computation becomes increasingly challenging as the dimension of the state space increases. The curse of dimensionality makes it difficult to accurately capture the reachable set in high-dimensional systems. As the number of dimensions grows, the computations become more complex and storage requirements are expected to increase.

Nonlinearity: Many real-world systems exhibit nonlinear dynamics and constraints, which can increase complexity in computing reachable sets. Nonlinearities can lead to non-convex reachable sets, making their computation more difficult.

Accuracy: System models need to be simplified in some of the cases to make the reachable set computation easier. However, model simplifications may lead to inaccurate or incomplete representations of system dynamics, potentially impacting the accuracy of reachable sets.

Computational Complexity: Reachable set computation can be computationally expensive, particularly for large-scale systems or when considering long time horizons. The numerical algorithms used for reachability analysis may require substantial computational resources.

Various methodologies exist for obtaining reachable sets (RS) for different classes of systems. These methodologies aim to strike a balance between **solution accuracy** and **computational requirements**. For linear systems, efficient algorithms have been recently developed, leading to well-understood methods for RS computation ([6], [7], [8], [9]). However, RS computation for nonlinear systems poses additional challenges as the resulting sets are typically nonconvex. As a result, only a limited number of methods are available for computing RS for nonlinear systems.

In general, three basic approaches can be classified for reachability set computation which are shown in Figure 9.

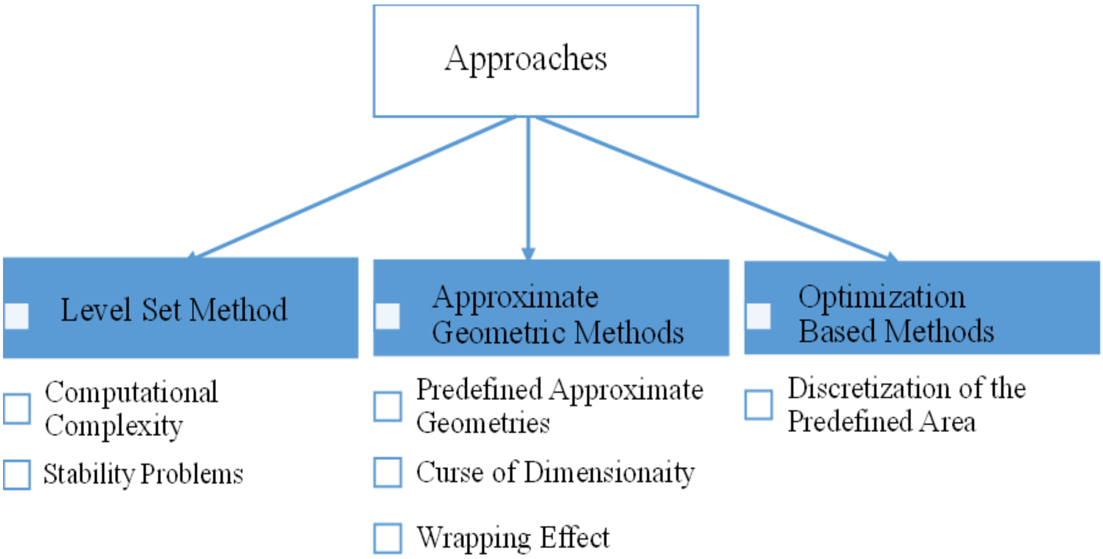

Approaches
Level Set Method
Approximate Geometric Methods
Optimization Based Methods
Computational Complexity
Stability Problems
Predefined Approximate Geometries
Curse of Dimensionaity
Wrapping Effect
Discretization of the Predefined Area

Figure 9: Fundamental Methods in Reachability Set Computation

2.2.1 Level Set Method

The prevalent numerical computation method, the Level Set Method [40] involves solving time-dependent partial differential equations and Hamiltonian equations. Level set function represents the distance to the boundary of the reachable set. The surface of the reachable set is defined by a partial differential equation in this approach. The PDE captures the dynamics of the system and the reachability properties. The level set function is updated iteratively using numerical techniques to solve the PDE. The method offers advantages such as capturing the full boundary of the reachable set. However, the solution procedure of the method introduces computational complexity, thereby restricting its applicability to smaller dimensional systems. Furthermore, the reachable set approximation may not converge to the actual reachable set, leading to potential deviations from the true solution.

2.2.2 Approximate Geometric Method

Numerical computation techniques are commonly employed in the literature to compute reachable sets, with a focus on approximating the reachable set using predefined geometries. These techniques aim to represent the reachable set by fitting it into predefined geometric shapes or structures.. Approximate Geometric Method represents the reachable set as using geometric shapes, such as a zonotope [6] or an ellipsoid [7]. This is based on the assumption that the reachable set for linear systems is convex in nature. Therefore, these methods employ geometric approximation techniques to construct convex representations of the reachable set. A consequence of this method is that the set grows at each iteration, leading to cumbersome computations of Minkowski sum [39]. Also, these geometric shapes provide a conservative approximation of the reachable set by bounding the possible states that the system can reach. Therefore, the method can overestimate the actual reachable set and this conservative approximation may lead to a larger approximation error which grows at each iteration and results in over-approximation of the complete set. Moreover, the computation of geometric approximations can be computationally intensive, especially for high-dimensional systems. Figure 10 depicts approximate geometric method employed for a dynamic system in a two-dimensional space.

Let P and B be two sets in 2 dimension. Assume the dynamic of the system is represented as follows:

$$x(k+1) = Ax(k) + Bv(k) \qquad \text{Eq. 3}$$

Q is the Minkowski sum of P and V which is used in the computation of the reachable set and defined as follows:

$$Q = P \oplus V, \{x + v \mid x \in P, v \in V\} \qquad \text{Eq. 4}$$

An extreme point on Q in a specific direction is the sum of extreme points in the specified direction on P and V. Affine transformations are applied to all combinations of vertices, resulting in an approximate convex shape. As the reachable set expands with each iteration, certain approximations have been proposed in the literature. Notably, one such approximation involves pushing the face by the element of Bv, aiming to provide a simplified representation of the reachable set. However, it is important to acknowledge that this approximation strategy introduces an inherent accumulation error (Wrapping Effect) as shown in Figure 10.

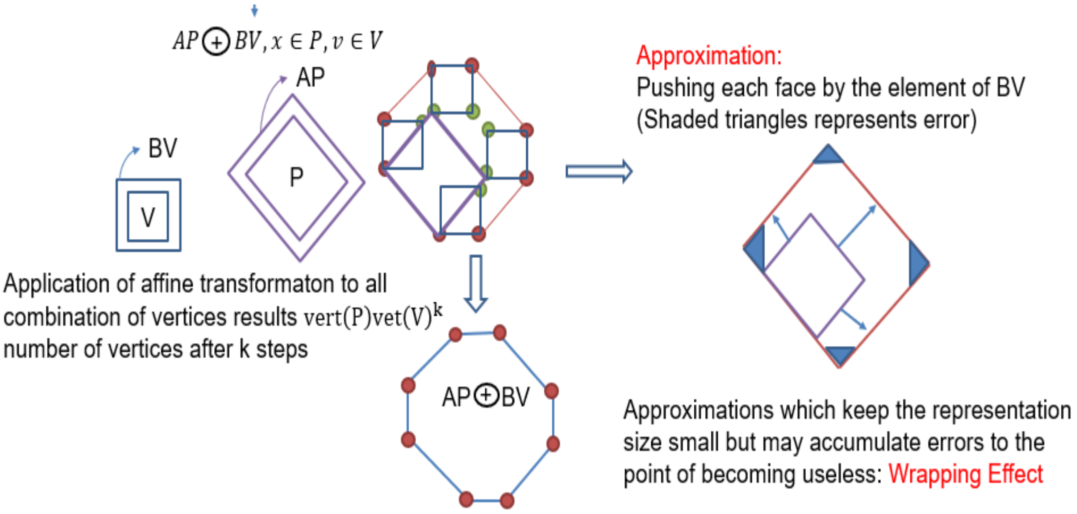

$AP \oplus BV, x \in P, v \in V$
AP
BV
V
P
Application of affine transformaton to all
combination of vertices results vert(P)vet(V)^k
number of vertices after k steps
AP⊕BV
Approximation:
Pushing each face by the element of BV
(Shaded triangles represents error)
Approximations which keep the representation
size small but may accumulate errors to the
point of becoming useless: Wrapping Effect

Figure 10: Application of Approximate Geometric Method

One approach that addresses the wrapping effect is presented in [6], where the algorithm utilizes zonotopes to approximate the reachable sets. Zonotopes are geometric objects that can represent the convex hull of a set of vectors and have been found to be effective in mitigating the accumulation of errors associated with the wrapping effect.

In [7], an alternative approach is introduced to address the computational complexity of polytopic geometric methods. This approach focuses on discrete-time systems and utilizes ellipsoidal approximations to compute the reachable sets. Ellipsoids are computationally efficient geometric shapes that can provide approximations of the reachable sets while reducing the computational complexity. By leveraging ellipsoidal approximations, [7] offers a practical solution to the computation of reachable sets for discrete-time systems.

These approaches demonstrate the diversity of techniques employed in the literature for computing reachable sets. By utilizing zonotopes and ellipsoids, these methods offer different trade-offs between computational complexity and accuracy. It should be also noted that these techniques are are well suited for linear systems.

2.2.3 Optimization Based Methods

Optimization Based Methods formulate the computation of reachable sets as an optimization problem. The objective is to find the set that contains all reachable states while satisfying system constraints. For general nonlinear systems, optimization-based methods ([41], [42]) are commonly applied. These approaches aim to find feasible solutions that capture the reachability properties of nonlinear systems.

However, it is important to note that the computation of RS for nonlinear systems remains an active area of research, and further advancements are needed to enhance accuracy and efficiency.

2.2.3.1 Optimization Based Methods for Convex Reachable Sets

Another noteworthy method is presented in [8], where the polytopic approximation of the reachable set is achieved through a set of support hyperplanes derived from an optimal control formulation. This approach combines the principles of optimal control theory with polytopic approximations to compute reachable sets. By formulating the problem as an optimal control problem, [8] provides a systematic framework for obtaining polytopic approximations that capture the reachable set boundaries.

The support function of a set $X \subseteq R^n$ is:

$$\rho(l, \mathrm{X}) = sup < l, x >, x \in X \qquad \text{Eq. 5}$$

The support function of a set is defined as the inner product between a given direction vector and any point on the boundary of the set. To approximate the reachable set using support functions, we define the support function $\rho(l, \mathrm{X})$ for a set $X(t)$ at time t in the direction of l.

The set of supporting points of X in l direction is defined as:

$$S(t, l) = (l^T x)_{x \in X} = \rho^*(l, X) \qquad \text{Eq. 6}$$

In the context of reachable set computation, the support function is used to compute the outer approximation of the reachable set. By considering a set of direction vectors, the support function can be calculated in each direction to obtain the bounding hyperplanes that define the reachable set.

To apply the support function formulation to the reachable set computation for a simple linear system without constraints, an optimization problem can be formulated to maximize the support function over a given time horizon.

The optimization problem can be formulated as follows:

$$\begin{gathered} maximize: S(t, l) = \max_{x \in X} l^T x\,(t) \\ subject\ to: \dot{x} = Ax + Bu \end{gathered} \qquad \text{Eq. 7}$$

$$x(0) \in X_0 (initial\ condition)$$

By solving the optimization problem for various direction vectors l, set of support hyperplanes that bound the reachable set can be determined. Each hyperplane is defined by its normal vector, which is proportional to the corresponding direction vector, and an offset.

It should be noted that this approach is only suitable for convex sets.

2.2.3.2 Optimization Based Methods for Nonconvex Reachable Sets

The methods discussed thus far have primarily focused on linear systems, as the reachable sets for nonlinear systems tend to be nonconvex. Limited research exists in the literature regarding the computation of reachable sets for nonlinear systems. Most of these studies are based on the approach outlined in [41]. In [41], the computation of reachable sets involves selecting a possible region for the reachable set and discretizing this region into grid points with a chosen step size as shown in Figure 11. Subsequently, an optimization problem is solved for each grid point to determine the control input that steers the system from the initial state to the final grid point. If a control input is found that guides the system to the desired grid point, then that grid point is considered reachable and included in the reachable set. Otherwise, the grid point is excluded from the set. The collection of all these final points forms the reachable set. The computation steps are outlined as follows:

Step1:Choose a region $G \subseteq R^n$ and discretize G into a grid $G_h \subset G$ with step size h, such that each element of G can be approximated by a grid point with error h.

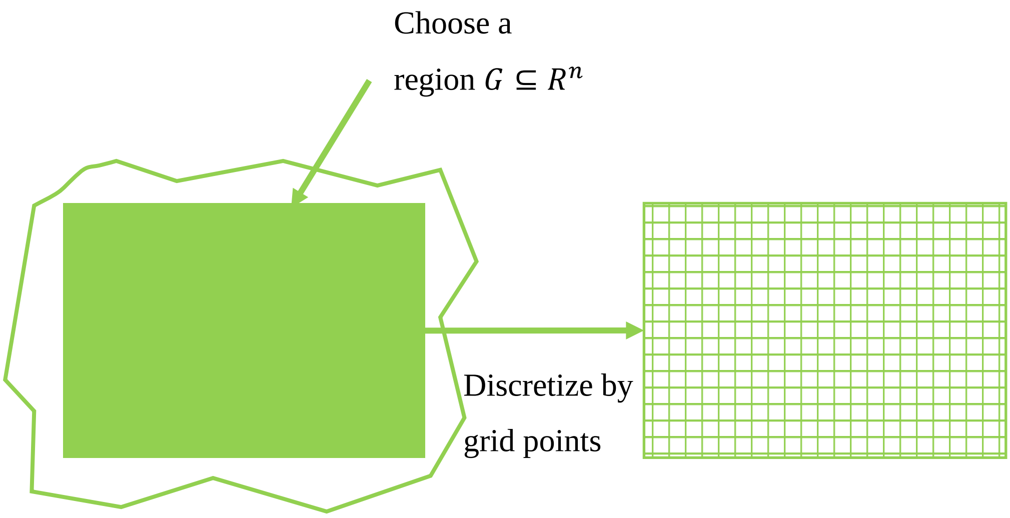


Figure 11: Discretization of Reachable Set by Grid Points

Step2: Solve the following optimal control problem for every $g_h \in G_h$:

$$Min\ \frac{1}{2}\|x(T) - g_h\|^2 \qquad \text{Eq. 8}$$

The condition shown below should also be satisfied:

$$\begin{gathered} x'(t) = f\big(x(t), u(t)\big), t_0 \leq t \leq T \\ x(0) = x_0 \\ u(t) = U \end{gathered} \qquad \text{Eq. 9}$$

Step3: Define the reachable set approximation by following relation:

$$R_h(T, t_0, x_0) = \bigcup_{g_h \in G_h} \{x^*(T; g_h)\} \qquad \text{Eq. 10}$$

The approach described earlier has been widely employed in numerous research studies focusing on nonlinear systems. One notable study, namely [42], presents a computational method specifically designed to approximate the reachable sets of nonlinear dynamic systems. This method utilizes grids to effectively cover the region of interest, and the distance function to the reachable set is evaluated at each grid point.

Similarly, in another study referenced as [37], an optimal control-based computation technique is applied to nonlinear systems. This involves projecting the equidistant grid points onto a reachable set to determine the maximum attainable landing area. To accomplish this, the state space is discretized using a set of uniform grid points, which are subsequently projected onto the plane defined by downrange and crossrange coordinates.

The primary advantage of the method is its ability to eliminate complex set operations such as Minkowski sum and convex hull calculations. By employing grid-based techniques, these operations can be avoided, simplifying the computational process.

However, it is important to note that this method is subject to the curse of dimensionality. As the state space dimension increases, the number of grid points required to accurately capture the system's behavior grows exponentially. This can result in computationally intensive calculations.

To address this challenge, one possible approach in literature is to compute a subset of the reachable set (RS) rather than the entire set. By focusing on a reduced set, the need for extensive computations can be alleviated while still providing valuable insights into the system's behavior.

2.3 Remarks

The computation of reachable sets (RS) is a crucial aspect in various fields, particularly in the analysis of nonlinear systems. Optimization-based methods have emerged as significant tools in RS computations, often combined with approximate geometric and level set methods for linear systems. However, a novel approach is proposed in this thesis, which solely relies on optimal control techniques to compute reachable sets for nonlinear systems. Moreover, an extension to the Model Predictive Static Programming (MPSP) technique is introduced by incorporating constraints on the control variable.

The devised numerical method enables the approximation of reachable sets for nonlinear problems through the utilization of optimal control techniques. The key idea is to formulate an optimal control problem that effectively captures the behavior of the system without explicitly requiring the determination of the feasible set coinciding with the reachable set. To compute the reachability boundaries, the MPSP method is employed for different search angles, flight durations, and initial flight conditions. The approach incorporates terminal constraints in the calculation of reachable sets. Alongside terminal constraints at the end of the flight time, the optimization problem encompasses the minimization of acceleration demands subject to specified acceleration limits at each time step. This objective aims to reduce the kinematic energy loss of the interceptor, ensuring more efficient and effective control. By meeting the specified performance criteria, the maximum and minimum range reachability boundaries are determined. The minimum and maximum boundary points of the reachable set for a specific search direction are calculated using a simple directional search approach, which involves moving the initial guess forward and backward in the search directions while assessing reachability at each step.

The proposed approach offers several advantages over existing methods discussed in Section 2.2.3.2:. Firstly, it eliminates the need for defining predefined geometries or grid point positions for reachability set computation, allowing for greater flexibility and ease of implementation. Additionally, the algorithm leverages the reuse of optimal control from previous terminal points as the starting guess for subsequent points. Notably, this approach readily integrates complex state constraints and boundary conditions, enabling more realistic and accurate modeling. The elimination of cumbersome set operations and the propagation of approximation errors is another notable advantage of the proposed method.

2.4 Exploration of Guidance Methods Incorporating Reachable Set Analysis

This section delves into the utilization of reachable set analysis within the context of exploring guidance methods. The study focuses on the midcourse phase of an aerodynamic interceptor as an illustrative problem. The objective is to develop an algorithm for reachable set analysis that can be incorporated into reachability-based midcourse guidance approaches.

One-to-One Engagement Case

The existing literature offers various guidance methods for aerodynamic interceptors, including Pursuit guidance, PN-based guidance, and Optimal guidance, which have been proposed to satisfy objectives in midcourse guidance. Recently, reachability-based guidance approaches have gained attention in midcourse guidance algorithms with the goal of maximizing target coverage at intercept, as governed by dynamic coverage theory, and/or defending a specific area.

Dynamic coverage theory deviates from traditional guidance law objectives by maximizing target coverage through the reachable set of pursuers at intercept. This approach is particularly useful when uncertainties in target states are high and precise target position information is unavailable. The target's location is characterized by probability density functions, and existing literature describes three primary approaches for guiding interceptors based on the target's probability density function: (1) the minimum mean square error criterion (MMSE), (2) the maximum a posteriori probability criterion (MAP), and (3) the highest probability interval criterion (HPI) [17]. The MMSE estimator aims to find target position that minimizes mean square error estimate of the target probability density function at interception. The MAP estimator, on the other hand, seeks the target position that maximizes the probability density function of the target at interception. Meanwhile, the HPI estimator strives to find the target position that maximizes the probability of the target remaining reachable by the interceptor. The following 2 papers propose a predictive guidance law that uses HPI as a terminal constraint for the interceptor. For instance, [17] presents a predictive guidance law that maximizes the probability of the target being present in the reachable set of the pursuer. The approach is employed for the scenario where there is uncertain information about the target state. The suggested method uses the probability density of the predicted target position to find the terminal constraint for the predictive control. The terminal constraint is chosen to position the reachable set of the pursuer to maximize the probability of interception. [18] is an extended study of [17] which is analyzed in the presence of multiple decoys. When the pursuer could not discriminate between the decoy and the target, guidance law maintains the undiscriminated target and decoys in the reachable set of the interceptor for as long as possible. The suggested approach is suitable when there is partial information on the target state. The benefits of the proposed approach are that it accounts for the probability density of the system's state and unknown future measurements, and the control input of the interceptor is bounded.

In another line of research, [19] proposes EIL concept for midcourse guidance algorithm to defend the area using Dubins curves. The earliest intercept line (EIL) which is computed by using the reachable set of interceptor and target is utilized to defend a given area. The method controls the movement of EIL to enable the area defended to be shaped during the engagement [19]. While [19] demonstrates how modifying the EIL can enhance the defended zone, it does not explicitly explain the real-time utilization, analytical or numerical control manipulation of the EIL within the midcourse guidance algorithm. To address this limitation, [20] presents an analytic approach to control and modify the EIL, thereby overcoming the deficiencies observed in the earlier study [19]. Similar to [19], this paper demonstrates a midcourse guidance strategy for area air defense based on EIL guidance. Intercept geometry and defended area are controlled by an analytical solution. The benefits of the proposed approach are: lateral acceleration saturation in the terminal phase is decreased and, the approach compensates for unpredictable target maneuvers. However, it is important to note that the aforementioned approaches are predicated upon certain assumptions. Specifically, they assume constant velocities for both the interceptor, neglecting factors such as energy loss due to maneuvers, aerodynamic drag, and energy increase resulting from thrust forces. These assumptions may potentially lead to inaccurate determination of reachable points, consequently resulting in unrealistic interception lines and reachable sets. Furthermore, it is worth mentioning that these approaches are designed for two-dimensional space, specifically in the lateral plane, thereby limiting their applicability in three-dimensional scenarios.

Many-on-Many Engagement Case

In the context of many-on-many engagements, considerable attention has been devoted to the utilization of traditional cooperative guidance approaches that impose constraints on impact time and impact angle at intercept. However, the existing literature on cooperative guidance incorporating reachability analysis remains relatively scarce. Many traditional cooperative guidance algorithms are devised based on time-to-go consensus, which implies that missiles are planned to intercept the target at the same time. While acceleration commands which are perpendicular to LOS vector are adjusted to ensure that miss distance converges to zero, the acceleration command along LOS is designed to guarantee that all interceptors reach the target simultaneously. The cooperation between interceptors is achieved via a communication network to reach a consensus on the arrival time. Central and distributed control architectures have been developed on the basis of consensus policy. [21] proposes a method to control impact time so that multiple interceptors can reach the target simultaneously. There are also several studies related to the impact time control of interceptors. For the aforementioned objective, several studies adopt different methods such as navigation gain scheduling, PN-based approaches, and receding horizon control to achieve interception at the specific time. References between [29]-[35] are notable studies related to cooperative guidance law for impact time control.

In some studies, the cooperation between interceptors is utilized to restrict the target’s possible set of evasive maneuvers by controlling the relative geometry. For instance, [22] proposes an approach for impact angle control, designing a guidance law that enables simultaneous interception of the target at a specific impact angle. [23] proposes an optimal control-based cooperative guidance law that enforces the desired relative terminal geometry between a team of interceptors. Furthermore, [24] pursues a different objective such that the cooperative guidance law aims to reduce variability of the relative look angle between two interceptors who are in collaboration against the maneuvering target. It is assumed that interceptors are equipped with a directional antenna, in order to take power consumption,robust communication and resistance to jamming into consideration. However, variability of look angle during flight may disturb the communication between interceptors; therefore, the objective of designed cooperative guidance law is chosen to reduce the variability of the look angle.

The cooperative guidance approaches mentioned up to this point do not take the reachability of interceptors into account explicitly. Instead, the interceptor's speed advantage is assumed to be sufficient for ensuring reachability at the intercept point. However, there exists such a possibility that the interceptor can not reach the target when uncertainty in target states is high or the target changes its strategy during flight; such as changing flight speed and heading. Moreover, coordination between numerous targets can decrease the possibility of an effective response by the air defense system. As a result, different from the traditional cooperative guidance approaches, application of dynamic coverage theory for the cooperative guidance problemis being investigated by researchers. The objective of such an approach is to maximize the coverage of the region of possible target locations in the reachable set of the pursuer. [25] suggests a coverage-based approach for the cooperative interception problem. The approach addresses multiple interceptors case against a maneuvering target with decoys. The joint interception probability of multiple interceptors is estimated, in order to maximize the probability of destroying a target vehicle escorted with decoys. Probability density function of the actual target position for a given interception time is estimated with the Kalman filter. The joint interception probability obtained with the consideration of the interceptor's kill radius. Interceptor positions for the next time step are calculated by maximizing joint interception probability. These positions are checked whether they are within the interceptor's reachable set. If they are not within their reachable set, then the nearest point in the reachable set is chosen as the desired interceptor position. The process is repeated until total flight time is achieved. This approach s developed with the assumption of linear and planar engagement kinematics. Furthermore, [26] introduces a cooperative guidance law for multiple interceptors against multiple high speed maneuvering targets with a centralized control strategy. Possible target locations due to target state uncertainties and noises are described by a region of space. Augmented reachable set of interceptors is positioned such that the space formed by possible positions of all targets is covered as much as possible. Consequently, the probability of interception is expected to be maximized. The analysis is performed in the lateral plane. [27] presents a cooperative guidance algorithm for pursuit and evasion problems. Pursuer's guidance algorithm employs the forward reachable set analysis. The union of multiple pursuers' forward reachable set is utilized, in order to obtain greater coverage for interception against a maneuvering evader. A two-dimensional kinematic model in the horizontal plane and Ackerman steering ground vehicle are used in the analysis. [28] introduces an asynchronous cooperative guidance law for multiple interceptors against a high speed maneuvering target. Reachable sets of interceptors are blended in order to cover possibility space for the target position in an asynchronous way and guarantee collision between interceptors.

The aforementioned reachability-based guidance approaches have been developed under the assumption of negligible speed variation for both the interceptor and the target resulting from factors such as aerodynamic drag and thrust. This assumption facilitates the analytical representation of interception lines, enabling the explicit calculation of strategies for the interceptor to control these lines. However, it is important to note that these assumptions may lead to interception lines and reachable sets that are deemed unacceptable in real-world applications, primarily due to significant errors in calculations, unless the constant speed assumption is reasonably valid and satisfied.

CHAPTER 3

NONLINEAR DYNAMICAL MODEL FOR AERODYNAMIC INTERCEPTOR

In order to analyze the reachable set of an interceptor, it is necessary to establish a mathematical model. This mathematical model is designed to accommodate non-ideal responses of the interceptor. In this study, a point mass interceptor model is employed as the basis for the analysis.

The calculation of the interceptor's velocity, position, and flight path angles involves integrating the derivatives of these state variables. By integrating the derivatives, the current states of the interceptor can be determined.

3.1 Coordinate Systems and Transformation Matrices

Two coordinate systems are utilized in this study to describe the motion of the interceptor. These are (1) the inertial coordinate system ($\mathcal{F}_I$) and (2) the wind coordinate system ($\mathcal{F}_W$). Each coordinate system provides a distinct reference frame for analyzing the interceptor's dynamics.

In the inertial coordinate system, axes are represented as $\vec{u}_1^I$,$\vec{u}_2^I$ and $\vec{u}_3^I$. The $\vec{u}_1^I$ axis points towards the north, the $\vec{u}_3^I$ axis points the downwards to Earth's center, and the $\vec{u}_2^I$ axis is the complementing orthogonal axis found by the right hand rule.

On the other hand, the wind coordinate system ($\mathcal{F}_W$) is attached to the interceptor and moves along with it. The $\vec{u}_1^W$ axis points toward the interceptor velocity direction. The $\vec{u}_2^W$ axis points to the right of the $\vec{u}_1^W$ axis in the horizontal plane. The $\vec{u}_3^W$ axis points down.

The related frame definitions and transformation matrices are shown below.

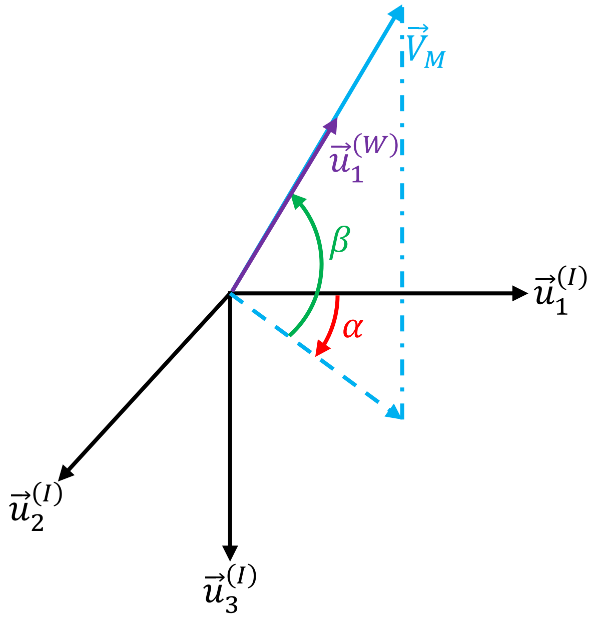


Figure 12: Representation of Interceptor Velocity Vector

To transition from the inertial frame to the wind frame, a series of rotations is performed. This transformation involves rotating $\mathcal{F}_I$ about the $\vec{u}_3^I$ axis by an angle α, yielding the intermediate frame $\mathcal{F}_O$. Subsequently, $\mathcal{F}_O$is rotated about the $\vec{u}_2^O$ axis by an angle β to obtain the wind frame $\mathcal{F}_W$. The coordinate axes are depicted in Figure 12, with the interceptor's velocity vector $\vec{V}_M$ aligned with the $\vec{u}_1^W$ axis.

$$\mathcal{F}_I \rightarrow R(\vec{u}_3^I, \alpha) \rightarrow \mathcal{F}_O \rightarrow R(\vec{u}_2^O, \beta) \rightarrow \mathcal{F}_W \qquad \text{Eq. 11}$$

Transformation matrix from O ($\mathcal{F}_O$) frame to inertial frame is determined by rotating $\mathcal{F}_I$ about $\vec{u}_3^I$ by an angle α. This matrix, denoted as $\hat{C}^{(I,O)}$, is given by:

$$\hat{C}^{(I,O)} = e^{\tilde{u}_3 \alpha} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq. 12}$$

Similarly, the transformation matrix from the wind frame to O frame is determined by rotating $\mathcal{F}_O$ about $\vec{u}_2^O$ by an angle β. This matrix, denoted as $\hat{C}^{(O,W)}$, is given by:

$$\hat{C}^{(O,W)} = e^{\tilde{u}_2 \beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \qquad \text{Eq. 13}$$

3.2 Ideal Interceptor Kinematics

In the ideal kinematics calculations, the states of a system can be determined by integrating the derivatives of those states. For example, in the case of the velocity vector $\vec{V}_M$ in the inertial frame, its derivative with respect to frame $\mathcal{F}_W$ is expressed as:

$$\left\{D_I\vec{V}_M\right\}^{(W)} = \left\{D_W\vec{V}_M + \vec{w}_{W/I} \times \vec{V}_M\right\}^{(W)} \qquad \text{Eq. 14}$$

Here, D_I and D_W represents derivative operations with respect to frame $\mathcal{F}_W$. $\vec{w}_{W/I}$ is the angular velocity of $\mathcal{F}_W$ with respect to $\mathcal{F}_I$. The angular velocity can be represented using the flight path angles in the following manner:

$$\left\{\vec{w}_{W/I}\right\}^{(W)} = \left\{\dot{\alpha}\vec{u}_3^{(O)} + \dot{\beta}\vec{u}_2^{(W)}\right\}^{(W)} = \begin{bmatrix} -\dot{\alpha}\sin\beta \\ \dot{\beta} \\ \dot{\alpha}\cos\beta \end{bmatrix} \qquad \text{Eq. 15}$$

The acceleration of the interceptor can be expressed as follows:

$$\left\{D_I\vec{V}_M\right\}^{(W)} = \begin{bmatrix} a_{M_x}^{\ W} \\ a_{M_y}^{\ W} \\ a_{M_z}^{\ W} \end{bmatrix} = \begin{bmatrix} \dot{V}_M \\ V_M\dot{\alpha}\cos\beta \\ -V_M\dot{\beta} \end{bmatrix} \qquad \text{Eq. 16}$$

The dynamics of the flight path angles α (azimuth angle) and β (climb angle) with respect to interceptor accelerations can be described as follows:

$$\begin{gathered} \dot{\alpha} = \frac{a_{M_y}^{\ W}}{V_M\cos(\beta)} \\ \dot{\beta} = -\frac{a_{M_z}^{\ W}}{V_M} \end{gathered} \qquad \text{Eq. 17}$$

When it is assumed that the rate of change of velocity magnitude depends on the thrust, drag force , and the component of gravity in the direction of the velocity vector, the rate of change of velocity magnitude can be expressed as follows:

$$\dot{V}_M = \frac{T - 0.5\rho V_M^2 S_{ref} C_X(\alpha,\beta,V_M)}{m} - g sin(\beta) \qquad \text{Eq. 18}$$

Furthermore, the position derivative of the interceptor in the $\mathcal{F}_I$ frame can be computed as:

$$\left\{D_I\vec{R}_M\right\}^{(I)} = \left\{\dot{\vec{R}}_M\right\} = \begin{bmatrix} \dot{R}_{M_x} \\ \dot{R}_{M_y} \\ \dot{R}_{M_z} \end{bmatrix} = \begin{bmatrix} V_M \cos(\beta) \cos(\alpha) \\ V_M \cos(\beta) \sin(\alpha) \\ -V_M \sin(\beta) \end{bmatrix} \qquad \text{Eq. 19}$$

The state space representation of the ideal interceptor can be described as follows:

$$\dot{x} = \begin{bmatrix} \dot{R}_{M_x} \\ \dot{R}_{M_y} \\ \dot{R}_{M_z} \\ \dot{V}_M \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} V_M \cos(\beta) \cos(\alpha) \\ V_M \cos(\beta) \sin(\alpha) \\ -V_M \sin(\beta) \\ {a_M}_x^W \\ \dfrac{{a_M}_y^W}{V_M \cos(\beta)} \\ -\dfrac{{a_M}_z^W}{V_M} - \dfrac{g cos(\beta)}{V_M} \end{bmatrix}$$

$$= \begin{bmatrix} V_M \cos(\beta) \cos(\alpha) \\ V_M \cos(\beta) \sin(\alpha) \\ -V_M \sin(\beta) \\ \dfrac{T - 0.5\rho V_M^2 S_{ref} C_X\left({a_M}_y^W, {a_M}_z^W, V_M\right)}{m} - g sin(\beta) \\ \dfrac{{a_M}_y^W}{V_M \cos(\beta)} \\ -\dfrac{{a_M}_z^W}{V_M} \end{bmatrix} \qquad \text{Eq. 20}$$

The forward difference approximation is used to calculate the time step Δt, discretized ideal interceptor model is as shown below:

$$\bar{x}(k+1) = F(\bar{x}(k), \bar{u}(k))$$

$$= \begin{bmatrix} R_{M_x}(k) + V_M(k)\cos\big(\beta(k)\big)\cos\big(\alpha(k)\big)\,\Delta t \\ R_{M_y}(k) + V_M(k)\cos\big(\beta(k)\big)\sin\big(\alpha(k)\big)\,\Delta t \\ R_{M_z}(k) - V_M(k)\sin\big(\beta(k)\big)\,\Delta t \\ V_M(k) + \dfrac{T - 0.5\rho V_M^2 S_{ref} C_X\left(a_{M_y}^{W}{}_{c}(k), a_{M_z}^{W}{}_{c}(k), V_M(k)\right)}{m}\Delta t - gsin\big(\beta(k)\big)\Delta t \\ \alpha(k) + \dfrac{a_{M_y}^{W}(k)}{V_M(k)\cos\big(\beta(k)\big)}\Delta t \\ \beta(k) - \dfrac{a_{M_z}^{W}(k)}{V_M(k)}\Delta t \end{bmatrix} \qquad \text{Eq. 21}$$

State variables at time t_k are represented by k for such that $R_{M_x}(k) = R_{M_x}(t_k)$.

The input vector is defined for ideal autopilot dynamics as follows:

$$\bar{u} = \begin{bmatrix} a_{M_{y,c}}^{W} \\ a_{M_{z,c}}^{W} \end{bmatrix} = \begin{bmatrix} a_{M_y}^{W} \\ a_{M_z}^{W} \end{bmatrix} \qquad \text{Eq. 22}$$

3.3 Non Ideal Interceptor Kinematics

To incorporate the response delay of the interceptor, a first-order transfer function is assumed to model the interceptor's response to a given command.

The transfer function is described by the equation:

$$\frac{a_{M_y}^{W}(s)}{a_{M_{y,c}}^{W}(s)} = \frac{a_{M_z}^{W}(s)}{a_{M_{z,c}}^{W}(s)} = \frac{1}{\tau s + 1} \qquad \text{Eq. 23}$$

Where $a_{M_y}^{W}$ and $a_{M_z}^{W}$ represent the yaw and pitch plane acceleration responses of the interceptor, respectively, $a_{M_{y,c}}^{W}$ and$a_{M_{z,c}}^{W}$ represent corresponding commanded accelerations, s is the Laplace variable, and τ is the time constant of the autopilot time response.

The dynamic response of the interceptor can be represented as follows:

$$\tau \dot{a}_{M_y}^{W}(t) + a_{M_y}^{W}(t) = a_{M_{y,c}}^{W}(t) \qquad \text{Eq. 24}$$

To solve this differential equation, we can express the solution as a sum of the homogeneous and particular solutions:

$$a_{M_y}^{W}(t) = a_h(t) + a_p(t) \qquad \text{Eq. 25}$$

$$\begin{aligned} a_h(t) &\rightarrow \mathit{Homogeneous\ solution} \\ a_p(t) &\rightarrow \mathit{Particular\ solution} \end{aligned} \qquad \text{Eq. 26}$$

The homogeneous solution can be expressed as:

$$a_h(t) = K_1 e^{xt} \qquad \text{Eq. 27}$$

Where K_1 is an arbitrary constant and x is the root of the characteristic equation given by:

$$\dot{a_h}(t) + \frac{a_h(t)}{\tau} = 0 \qquad \text{Eq. 28}$$

$$K_1 e^{xt} x + \frac{K_1 e^{xt}}{\tau} = K_1 e^{xt}\left(x + \frac{1}{\tau}\right) = 0 \rightarrow x = -\frac{1}{\tau} \qquad \text{Eq. 29}$$

Thus, the homogeneous solution becomes:

$$a_h(t) = K_1 e^{-\frac{t}{\tau}} \qquad \text{Eq. 30}$$

For the particular solution, a suggested form is:

$$a_p(t) = \ K_2(t) e^{-\frac{t}{\tau}} \qquad \text{Eq. 31}$$

Differentiating $a_p(t)$ with respect to time yields:

$$\dot{a}_p(t) = \left(\dot{K}_2 - \frac{K_2}{\tau}\right) e^{-\frac{t}{\tau}} \qquad \text{Eq. 32}$$

Substituting $a_p(t)$ into differential equation yields:

$$\dot{a_p}(t) + \frac{a_p(t)}{\tau} = \left(\dot{K}_2 - \frac{K_2}{\tau} + \frac{K_2}{\tau}\right) e^{-\frac{t}{\tau}} = \dot{K}_2 e^{-\frac{t}{\tau}} = \frac{a_{M_{y,c}}^{W}(t)}{\tau} \qquad \text{Eq. 33}$$

Integrating both sides of the equation yields:

$$K_2 = \int \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi + K_3 \qquad \text{Eq. 34}$$

$$\rightarrow discard\ K_3, since\ it\ will\ mix\ into\ the\ homogeneous\ solution$$

The particular solution becomes:

$$a_p(t) = \left(\int_0^t \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) e^{-\frac{t}{\tau}} \qquad \text{Eq. 35}$$

After finding particular and homogenous solution, the general solution ($a_{M_y}^{W}(t)$) in Eq. 25 can be expressed as follows:

$$a_{M_y}^{W}(t) = K_1 e^{-\frac{t}{\tau}} + \left(\int_0^t \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) e^{-\frac{t}{\tau}} \qquad \text{Eq. 36}$$

$$a_{M_y}^{W}(t_k) = K_1 e^{-\frac{t_k}{\tau}} + \left(\int_0^{t_k} \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) e^{-\frac{t_k}{\tau}} = a_{M_y}^{W}(k) \qquad \text{Eq. 37}$$

$$K_1 = a_{M_y}^{W}(k) e^{\frac{t_k}{\tau}} - \left(\int_0^{t_k} \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) \qquad \text{Eq. 38}$$

Then, the general solution becomes:

$$a_{M_y}^{W}(t) = \left[a_{M_y}^{W}(k) e^{\frac{t_k}{\tau}} - \left(\int_0^{t_k} \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) \right] e^{-\frac{t}{\tau}} + \left(\int_0^t \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{\frac{\xi}{\tau}} d\xi \right) e^{-\frac{t}{\tau}} \qquad \text{Eq. 39}$$

$$a_{M_y}^{W}(t) = a_{M_y}^{W}(k) e^{-\frac{(t-t_k)}{\tau}} + \left(\int_{t_k}^{t} \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{-\frac{(t-\xi)}{\tau}} d\xi \right) \qquad \text{Eq. 40}$$

For $t = t_{k+1}$ and $t_{k+1} - t_k = \Delta t$:

$$a_{M_y}^{W}(t_{k+1}) = a_{M_y}^{W}(k) e^{-\frac{\Delta t}{\tau}} + \left(\int_{t_k}^{t_{k+1}} \frac{a_{M_{y,c}}^{W}(\xi)}{\tau} e^{-\frac{(t_{k+1}-\xi)}{\tau}} d\xi \right) \qquad \text{Eq. 41}$$

If we assume that $a_{M_{y,c}}^{W}(t)$ is constant between sampling updates, such that $a_{M_{y,c}}^{W}(t) = a_{M_{y,c}}^{W}(k), when\, t_k \leq t < t_{k+1}$:

$$a_{M_y}^{W}(t_{k+1}) = a_{M_y}^{W}(k)e^{-\frac{\Delta t}{\tau}} + \frac{a_{M_{y,c}}^{W}(k)}{\tau}\int_{t_k}^{t_{k+1}} e^{-\frac{(t_{k+1}-\xi)}{\tau}} d\xi \qquad \text{Eq. 42}$$

$$a_{M_y}^{W}(t_{k+1}) = a_{M_y}^{W}(k)e^{-\frac{\Delta t}{\tau}} + \frac{a_{M_{y,c}}^{W}(k)}{\tau}e^{-\frac{t_{k+1}}{\tau}}\left[\int_{t_k}^{t_{k+1}} e^{\frac{\xi}{\tau}} d\xi\right] \qquad \text{Eq. 43}$$

$$a_{M_y}^{W}(t_{k+1}) = a_{M_y}^{W}(k)e^{-\frac{\Delta t}{\tau}} + \frac{a_{M_{y,c}}^{W}(k)e^{-\frac{t_{k+1}}{\tau}}}{\tau}\left[\tau\left(e^{\frac{t_{k+1}}{\tau}} - e^{\frac{t_k}{\tau}}\right)\right] \qquad \text{Eq. 44}$$

$$a_{M_y}^{W}(t_{k+1}) = a_{M_y}^{W}(k)e^{-\frac{\Delta t}{\tau}} + a_{M_{y,c}}^{W}(k) - a_{M_{y,c}}^{W}(k)e^{-\frac{\Delta t}{\tau}} \qquad \text{Eq. 45}$$

τ represent a time constant of the autopilot time response. The discretization of the acceleration time response from time t_k to t_{k+1} given by the following expression:

$$a_{M_y}^{W}(k+1) = a_{M_y}^{W}(k)e^{-\frac{\Delta t}{\tau}} + a_{M_{y,c}}^{W}(k)\left(1 - e^{-\frac{\Delta t}{\tau}}\right) \qquad \text{Eq. 46}$$

Similarly, the acceleration response in the pitch plane can be represented as:

$$a_{M_z}^{W}(k+1) = a_{M_z}^{W}(k)e^{-\frac{\Delta t}{\tau}} + a_{M_{z,c}}^{W}(k)\left(1 - e^{-\frac{\Delta t}{\tau}}\right) \qquad \text{Eq. 47}$$

In this representation, the acceleration profile in yaw and pitch channels become new state variables. Therefore, the discretized non-ideal interceptor model can be demonstrated as:

$$\bar{\mathrm{x}}(k+1) = F\big(\bar{\mathrm{x}}(k), \bar{u}(k)\big) = \begin{bmatrix} R_{M_x}(k+1) \\ R_{M_y}(k+1) \\ R_{M_z}(k+1) \\ V_M(k+1) \\ \alpha(k+1) \\ \beta(k+1) \\ a_{M_y}^W(k+1) \\ a_{M_z}^W(k+1) \end{bmatrix} = \begin{bmatrix} F_1\big(\bar{x}(k), \bar{u}(k)\big) \\ F_2\big(\bar{x}(k), \bar{u}(k)\big) \\ F_3\big(\bar{x}(k), \bar{u}(k)\big) \\ F_4\big(\bar{x}(k), \bar{u}(k)\big) \\ F_5\big(\bar{x}(k), \bar{u}(k)\big) \\ F_6\big(\bar{x}(k), \bar{u}(k)\big) \end{bmatrix}$$

$$= \begin{bmatrix} R_{M_x}(k) + V_M(k)\cos\big(\beta(k)\big)\cos\big(\alpha(k)\big)\,\Delta t \\ R_{M_y}(k) + V_M(k)\cos\big(\beta(k)\big)\sin\big(\alpha(k)\big)\,\Delta t \\ R_{M_z}(k) - V_M(k)\sin\big(\beta(k)\big)\,\Delta t \\ V_M(k) + \dfrac{T - 0.5\rho V_M^2 S_{ref} C_X\left(a_{M_y}^W(k), a_{M_z}^W(k), V_M(k)\right)}{m}\Delta t - g sin\big(\beta(k)\big)\Delta t \\ \alpha(k) + \dfrac{a_{M_y}^W(k)}{V_M(k)\cos\big(\beta(k)\big)}\Delta t \\ \beta(k) - \dfrac{a_{M_z}^W(k)}{V_M(k)}\Delta t \\ a_{M_y}^W(k)e^{-\frac{\Delta t}{\tau}} + a_{M_{y,c}}^W(k)\left(1 - e^{-\frac{\Delta t}{\tau}}\right) \\ a_{M_z}^W(k)e^{-\frac{\Delta t}{\tau}} + a_{M_{z,c}}^W(k)\left(1 - e^{-\frac{\Delta t}{\tau}}\right) \end{bmatrix} \qquad \text{Eq. 48}$$

The control input vector is specified as:

$$\bar{\mathrm{u}}(k) = \begin{bmatrix} a_{M_{y,c}}^W(k) \\ a_{M_{z,c}}^W(k) \end{bmatrix} \qquad \text{Eq. 49}$$

3.4 Environment

In this study, the computation of air density and speed of sound is based on the standard atmosphere model. These quantities are updated in accordance with the interceptor's altitude. The temperature can be expressed by the equation:

$$\mathrm{Temp} = T_{SL} + ah \qquad \text{Eq. 50}$$

In this equation, h is the altitude (m) and T_{SL} denotes the sea level temperature which is 288.16 K and a is the lapse rate. The lapse rate is taken as -0.0065 K/m for altitudes below 11 km..

The density can be calculated using the following equation:

$$\rho = \rho_{SL} + \left(\frac{\text{Temp}}{T_{SL}}\right)^{-g/(1+aR_g)} \qquad \text{Eq. 51}$$

Here, ρ_{SL} is see level density which is $1.225\ \text{kg/m}^3$, R_g is the specific gas constant which is $287\frac{\text{J}}{kg}.K$ and$g$ is $9.80665\ \text{m/s}^2$

3.5 Modelling of Aerodynamic Drag

In the model of the interceptor, the consideration of aerodynamic drag is crucial, as it relies on both the acceleration response and the Mach number. Consequently, the drag model incorporates the kinematic energy loss of the interceptor resulting from acceleration and base drag. The aerodynamic drag model can be expressed as follows:

$$C_X\left(a_{M_y}^{W}, a_{M_z}^{W}, V_M\right) = C_{X_0}(V_M) + C_{X_a}(V_M)\sqrt{{a_{M_y}^{W}}^2 + {a_{M_z}^{W}}^2} \qquad \text{Eq. 52}$$

In this equation, $C_{X_0}(V_M)$ represents base drag coefficient at a given velocity and $C_{X_a}(V_M)$ represent drag experienced by the interceptor due to acceleration. The term $\sqrt{{a_{M_y}^{W}}^2 + {a_{M_z}^{W}}^2}$ accounts for the magnitude of the acceleration response in the lateral and vertical directions.

Since drag coefficients are typically available only for discrete flight variables, the drag coefficients for intermediate values of the flight variables are obtained through linear interpolation. This allows for the estimation of drag coefficients at specific points within the range of flight variables.

By incorporating both the acceleration response and the Mach number into the drag model and utilizing linear interpolation for estimating drag coefficients, the interceptor model takes into account the effects of aerodynamic drag on the interceptor's performance and the resulting kinematic energy loss.

CHAPTER 4

GUIDANCE ALGORITHM DEVELOPMENT

This chapter presents the guidance algorithm design that employs the MPSP approach. Specifically, the implementation of the Suboptimal Midcourse Guidance with Terminal Position Constraint method (4.1), which has shown promising performance in the literature [4], is investigated for an aerodynamic interceptor. By incorporating this method into the research, the aim is to enhance the guidance capabilities of aerodynamic interceptors and further investigate their operational effectiveness.

To improve upon the existing guidance algorithm, the equations are extended to incorporate input constraints, particularly bounded acceleration (4.2). These constraints reflect the physical limitations of the interceptor's propulsion system and impose restrictions on the maximum achievable acceleration during its trajectory. By accounting for these constraints, the modified guidance algorithm aims to provide more realistic and practical guidance commands, leading to improved interceptor performance.

In order to assess the performance of the modified suboptimal midcourse guidance algorithm with input constraint, a comparative analysis is conducted against the previously implemented Suboptimal Midcourse Guidance with Terminal Position Constraint method. By evaluating the performance of these two algorithms, the improvements achieved by incorporating the input constraint can be quantitatively measured, and the algorithm's ability to meet equalitty constraint on terminal position can be ascertained.

4.1 Suboptimal Midcource Guidance with Terminal Position Constraint

The output for the nonlinear system discussed in Chapter 3 is expressed as a linear combination of various state and control variables.

$$\bar{y}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{M_x}(k) \\ R_{M_y}(k) \\ R_{M_z}(k) \\ V_M(k) \\ \alpha(k) \\ \beta(k) \\ a_{M_y}^W(k) \\ a_{M_z}^W(k) \end{bmatrix} \qquad \text{Eq. 53}$$

The primary goal is to compute the control input vector $\bar{u}(k)$ at each time step, ensuring that the final output reaches to desired value $\bar{y}_N^d$ with minimal control action. This optimization problem can be formulated with a cost function that quantifies the control effort and constraints that ensure the output reaches the desired value.

Hence, the optimization objective can be succinctly described as follows:

$$\min_{\bar{u}_k} J(\bar{u}_k) = \frac{1}{2}\sum_{k=1}^{N-1} \bar{u}_k^T R_k \bar{u}_k, \qquad \text{Eq. 54}$$

$$subject\ to\ the\ following\ constraints: \bar{y}_N - \bar{y}_N^d = 0$$

To facilitate the optimization process, initial control input vector history is used to linearize the nonlinear system and its output. This linearization is performed to approximate the system dynamics and output behavior.

$$d\bar{y}_N \cong \bar{y}_N^* - \bar{y}_N \qquad \text{Eq. 55}$$

$$d\bar{x}_k \cong \bar{x}_k^* - \bar{x}_k \qquad \text{Eq. 56}$$

$$d\bar{u}_k \cong \bar{u}_k^* - \bar{u}_k \qquad \text{Eq. 57}$$

$$d\bar{y}_N = Ad\bar{x}_1 + B_1 d\bar{u}_1 + B_2 d\bar{u}_2 + \cdots + B_{N-1} d\bar{u}_{N-1} \qquad \text{Eq. 58}$$

Since $\bar{x}_1^*$ is set equal to the initial condition $\bar{x}_1$, $d\bar{x}_1$ becomes zero. Therefore, $d\bar{y}_N$ in Eq. 58 simplifies to:

$$d\bar{y}_N = \sum_{k=1}^{N-1} B_k d\bar{u}_k \qquad \text{Eq. 59}$$

$$A = \left[\frac{\partial \bar{y}_N}{\partial \bar{x}_N}\right]\left[\frac{\partial \bar{F}_{N-1}}{\partial \bar{x}_{N-1}}\right]\left[\frac{\partial \bar{F}_{N-2}}{\partial \bar{x}_{N-2}}\right]\cdots\left[\frac{\partial \bar{F}_1}{\partial \bar{x}_1}\right] \qquad \text{Eq. 60}$$

$$B_k = \left[\frac{\partial \bar{y}_N}{\partial \bar{x}_N}\right]\left[\frac{\partial \bar{F}_{N-1}}{\partial \bar{x}_{N-1}}\right]\left[\frac{\partial \bar{F}_{N-2}}{\partial \bar{x}_{N-2}}\right]\cdots\left[\frac{\partial \bar{F}_{k+1}}{\partial \bar{x}_{k+1}}\right]\left[\frac{\partial \bar{F}_k}{\partial \bar{u}_k}\right] = B_k^0\left[\frac{\partial \bar{F}_k}{\partial \bar{u}_k}\right] \qquad \text{Eq. 61}$$

The linearized equations can be represented in matrix form, where the matrices A and B_k capture the dependencies between the variables. The matrix A represents the partial derivatives of the output with respect to the state variables. The matrix B_k represents the partial derivatives of the output with respect to the control variables. These matrices are calculated iteratively starting from $B_{N-1}^0 = \left[\frac{\partial \bar{y}_N}{\partial \bar{x}_N}\right]$.

Note that $\frac{\partial \bar{F}(\bar{x}(k),\bar{u}(k))}{\partial \bar{x}(k)}$ and $\frac{\partial \bar{F}(\bar{x}(k),\bar{u}(k))}{\partial \bar{u}(k)}$ can be expressed as follows:

$$\frac{\partial \bar{F}(\bar{x}(k),\bar{u}(k))}{\partial \bar{x}(k)} = \begin{bmatrix} 1 & 0 & 0 & dt\cos(\beta)\cos(\alpha) & -dtV_M\cos(\beta)\sin(\alpha) & -dtV_M\sin(\beta)\cos(\alpha) & 0 & 0 \\ 0 & 1 & 0 & dt\cos(\beta)\sin(\alpha) & dtV_M\cos(\beta)\cos(\alpha) & -dtV_M\sin(\beta)\sin(\alpha) & 0 & 0 \\ 0 & 0 & 1 & -dt\sin(\beta) & 0 & -dtV_M\cos(\beta) & 0 & 0 \\ 0 & 0 & 0 & 1-dt\frac{\rho V_M C_X}{m} & 0 & -dtg\cos(\beta) & -dt\frac{0.5\rho V_M^2 C_{X_a} a_{M_y}^W}{m\sqrt{{a_{M_y}^W}^2 + {a_{M_z}^W}^2}} & -dt\frac{0.5\rho V_M^2 C_{X_a} a_{M_z}^W}{m\sqrt{{a_{M_y}^W}^2 + {a_{M_z}^W}^2}} \\ 0 & 0 & 0 & -dt\frac{{a_{M_y}}^W}{V_M^2\cos(\beta)} & 1 & dt\frac{{a_{M_y}}^W}{V_M\cos(\beta)^2}\sin(\beta) & \frac{dt}{V_m\cos\beta} & 0 \\ 0 & 0 & 0 & dt\frac{{a_{M_z}}^W + g\cos(\beta)}{V_M^2} & 0 & 1+dt\frac{g}{V_M}\sin(\beta) & 0 & -\frac{dt}{V_m} \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\tau \mathrm{dt}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\tau \mathrm{dt}} \end{bmatrix} \qquad \text{Eq. 62}$$

$$\frac{\partial \bar{F}(\bar{x}(k),\bar{u}(k))}{\partial \bar{u}(k)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1-e^{-\tau \mathrm{d}t} & 0 \\ 0 & 1-e^{-\tau \mathrm{d}t} \end{bmatrix} \qquad \text{Eq. 63}$$

By utilizing the definitions provided above, the cost function can be formulated using $d\bar{u}_k$:

$$J = \frac{1}{2}\sum_{k=1}^{N-1} (\bar{u}_k^* - d\bar{u}_k)^T R_k (\bar{u}_k^* - d\bar{u}_k) \qquad \text{Eq. 64}$$

To simplify the optimization process, the equality constraint is formulated in terms of $d\bar{u}_k$. In order to achieve the desired output value $\bar{y}_N^d$, the following equality must be satisfied:

$$d\bar{y}_N = \bar{y}_N^d - \bar{y}_N^* \qquad \text{Eq. 65}$$

The difference in input vector history has to meet $d\bar{y}_N$ condition specified in Eq. 65. By leveraging the connection between the input vector history and $d\bar{y}_N$, equality constraint can be formulated with respect to $d\bar{u}_k$ as depicted below:

$$d\bar{y}_N - \sum_{k=1}^{N-1} B_k d\bar{u}_k = 0 \qquad \text{Eq. 66}$$

To solve the optimization problem, a Hamiltonian function J^* is defined using the cost function (Eq. 64) and the equality constraint (Eq. 66):

$$\begin{aligned} J^*\left(d\bar{u}_k, \bar{\lambda}\right) = \frac{1}{2}\sum_{k=1}^{N-1} (\bar{u}_k^* - d\bar{u}_k)^T R_k (\bar{u}_k^* - d\bar{u}_k) \\ + \bar{\lambda}^T \left(d\bar{y}_N - \sum_{k=1}^{k=N-1} B_k d\bar{u}_k \right) \end{aligned} \qquad \text{Eq. 67}$$

where $\bar{\lambda}$, is the Lagrange multiplier. The necessary conditions for optimality are applied, which involve setting the partial derivatives of the Hamiltonian function with respect to the control variables and Lagrange multipliers to zero. By solving these equations, the optimal control input for each time step is obtained:

$$\frac{\partial J^*}{\partial \bar{\lambda}} = 0 \rightarrow \boxed{d\bar{y}_N = \sum_{k=1}^{k=N-1} B_k d\bar{u}_k} \qquad \text{Eq. 68}$$

$$\frac{\partial J^*}{\partial d\bar{u}_k} = 0 \rightarrow \boxed{d\bar{u}_k = \bar{u}_k^* + R_k^{-1}\left(B_k^T \bar{\lambda}\right)}, \qquad \text{Eq. 69}$$
$$k = 1,2,\cdots,(N-1)$$

The determination of the Lagrange multipliers involves expressing the equality constraint in terms of the control variables and solving for the multipliers. The process for determining the related Lagrange multipliers is illustrated below:

$$d\bar{y}_N = \sum_{k=1}^{k=N-1} B_k\left(\bar{u}_k^* + R_k^{-1}\left(B_k^T \bar{\lambda}\right)\right) \rightarrow \boxed{\bar{\lambda} = -A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda\right)}$$
$$\text{where } A_\lambda = -\left(\sum_{k=1}^{k=N-1} B_k R_k^{-1} B_k^T\right)^{-1}, \bar{b}_\lambda = \sum_{k=1}^{k=N-1} B_k \bar{u}_k^* \qquad \text{Eq. 70}$$

The updated expression for the control input (Eq. 71) is obtained by substituting the Lagrange multipliers (Eq. 70) into the equation in Eq. 69:

$$d\bar{u}_k = \bar{u}_k^* - R_k^{-1} B_k^T A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda\right) \qquad \text{Eq. 71}$$

It should be noted that the derivation of the closed-form control update relies on small error approximations, which may not always hold true. Therefore, an iterative process is often required to refine the control input and converge to the optimal solution. The convergence is defined when the desired output value is closely approximated by the actual output value.

Overall, the process involves linearizing the nonlinear system, formulating an optimization problem with a cost function and constraints, deriving the necessary conditions for optimality, updating the control input based on the Lagrange multipliers, and iteratively refining the solution until convergence is achieved.

4.2 Suboptimal Midcource Guidance with Terminal Position Constraint and Bounded Acceleration

This part of the study provides a detailed derivation of the equations governing the modified guidance algorithm, considering both the terminal position constraint and the input constraint of bounded acceleration. The resulting equations capture the intricate relationships between the interceptor's position, velocity, and acceleration, as well as the desired terminal position. Through this derivation, a solid foundation is aimed to be established for the subsequent analysis and evaluation.

4.2.1 Derivations of Equations for the Design of Midcource Guidance Algorithm with Terminal Position and Input Constraints

The optimization problem, which is solved using the model predictive static programming, involves determining the inputs at time steps $k = 1,2,\cdots,(N-1)$ that satisfy the conditions specified below:

- The output at final time step,($\bar{y}_N$), should be equal to the desired output:

$$\bar{y}_N = \bar{y}_N^d \qquad \text{Eq. 72}$$

- The inputs should satisfy the inequality constraints defined for the each time step , where G_k and $\bar{W}_k$ represent the constraint matrices:

$$G_k\bar{u}_k \leq \bar{W}_k, where\ G_k = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}\ and\ \bar{W}_k = \begin{bmatrix} a(k) \\ a(k) \\ b(k) \\ b(k) \end{bmatrix} \qquad \text{Eq. 73}$$

- The calculation of inputs needs to minimize the cost function defined as follows:

$$J(\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_{N-1}) = \frac{1}{2}\sum_{k=1}^{N-1} \bar{u}_k^T R_k \bar{u}_k \qquad \text{Eq. 74}$$

Thus, the goal of the optimization problem can be summarized in the following manner:

$$\min_{\bar{u}_k} J(\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_{N-1}) \\ = subject\ to\ the\ following\ constraints: \qquad \text{Eq. 75}$$

$$\bar{y}_N - \bar{y}_N^d = 0 \qquad \text{Eq. 76}$$

$$G_k \bar{u}_k - \bar{W}_k \leq 0, \qquad k = 1,2,\cdots,(N-1) \qquad \text{Eq. 77}$$

To calculate the inputs that solve the optimization problem, Hilderth's procedure [10] for solving inequality constraints on inputs is employed in conjunction with the model predictive static programming method [4]. Prior to constructing the Hamiltonian function, which is a prerequisite for the application of the approach, the nonlinear interceptor model needs to be linearized with respect to the initial input values ($\bar{u}_k^0,$, $k = 1,2,\cdots,(N-1)$) as shown below:

$$\bar{y}_N = \bar{y}_N^* + \left[\frac{\partial \bar{y}_N}{\partial \bar{x}_N}\right](\bar{x}_N - \bar{x}_N^*) + HigherOrderTerms \qquad \text{Eq. 78}$$

$$\bar{x}_{k+1} = \bar{F}_k(\bar{x}_k^*, \bar{u}_k^*) + \left[\frac{\partial \bar{F}_k}{\partial \bar{x}_k}\right]\Bigg|_{\substack{\bar{x}_k = \bar{x}_k^* \\ \bar{u}_k = \bar{u}_k^*}} (\bar{x}_k - \bar{x}_k^*) \\ + \left[\frac{\partial \bar{F}_k}{\partial \bar{u}_k}\right]\Bigg|_{\substack{\bar{x}_k = \bar{x}_k^* \\ \bar{u}_k = \bar{u}_k^*}} (\bar{u}_k - \bar{u}_k^*) + HigherOrderTerms \qquad \text{Eq. 79}$$

$$d\bar{y}_N \cong \bar{y}_N^* - \bar{y}_N \qquad \text{Eq. 80}$$

$$d\bar{x}_k \cong \bar{x}_k^* - \bar{x}_k \qquad \text{Eq. 81}$$

$$d\bar{u}_k \cong \bar{u}_k^* - \bar{u}_k \qquad \text{Eq. 82}$$

By utilizing the definitions $d\bar{y}_N$ (Eq. 80) , $d\bar{x}_k$ (Eq. 81) and $d\bar{u}_k$ (Eq. 82), and under the assumption that the higher-order terms in Eq. 78 and Eq. 79 can be neglected due to the small error assumption, the final time step can be expressed as shown below:

$$d\bar{y}_N = \left[\frac{\partial \bar{y}_N}{\partial \bar{x}_N}\right]\Bigg|_{\bar{x}_N = \bar{x}_N^*} d\bar{x}_N \qquad \text{Eq. 83}$$

$$\bar{x}_N = \bar{F}(\bar{x}^*_{N-1}, \bar{u}^*_{N-1}) + \left[\frac{\partial \bar{F}_{N-1}}{\partial \bar{x}_{N-1}}\right]\Bigg|_{\substack{\bar{x}_{N-1}=\bar{x}^*_{N-1} \\ \bar{u}_{N-1}=\bar{u}^*_{N-1}}} (\bar{x}_{N-1} - \bar{x}^*_{N-1}) + \left[\frac{\partial \bar{F}_{N-1}}{\partial \bar{u}_{N-1}}\right]\Bigg|_{\substack{\bar{x}_{N-1}=\bar{x}^*_{N-1} \\ \bar{u}_{N-1}=\bar{u}^*_{N-1}}} (\bar{u}_{N-1} - \bar{u}^*_{N-1}) \qquad \text{Eq. 84}$$

$$\frac{\partial \bar{F}(\bar{x}(k), \bar{u}(k))}{\partial \bar{x}(k)} = \begin{bmatrix} 1 & 0 & 0 & dt\cos(\beta)\cos(\alpha) & -dtV_M\cos(\beta)\sin(\alpha) & -dtV_M\sin(\beta)\cos(\alpha) & 0 & 0 \\ 0 & 1 & 0 & dt\cos(\beta)\sin(\alpha) & dtV_M\cos(\beta)\cos(\alpha) & -dtV_M\sin(\beta)\sin(\alpha) & 0 & 0 \\ 0 & 0 & 1 & -dt\sin(\beta) & 0 & -dtV_M\cos(\beta) & 0 & 0 \\ 0 & 0 & 0 & 1 - dt\frac{\rho V_M C_X}{m} & 0 & -dtgcos(\beta) & -dt\frac{0.5\rho V_M^2 C_{X_a} a^W_{M_y}}{m\sqrt{{a^W_{M_y}}^2 + {a^W_{M_z}}^2}} & -dt\frac{0.5\rho V_M^2 C_{X_a} a^W_{M_z}}{m\sqrt{{a^W_{M_y}}^2 + {a^W_{M_z}}^2}} \\ 0 & 0 & 0 & -dt\frac{{a_{M_y}}^W}{V_M^2\cos(\beta)} & 1 & dt\frac{{a_{M_y}}^W}{V_M\cos(\beta)^2}\sin(\beta) & \frac{dt}{V_m\cos\beta} & 0 \\ 0 & 0 & 0 & dt\frac{{a_{M_z}}^W + gcos(\beta)}{V_M^2} & 0 & 1 + dt\frac{g}{V_M}\sin(\beta) & 0 & -\frac{dt}{V_m} \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-\tau \mathrm{dt}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\tau \mathrm{dt}} \end{bmatrix} \qquad \text{Eq. 85}$$

$$\frac{\partial \bar{F}(\bar{x}(k), \bar{u}(k))}{\partial \bar{u}(k)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 - e^{-\tau \mathrm{d}t} & 0 \\ 0 & 1 - e^{-\tau \mathrm{d}t} \end{bmatrix} \qquad \text{Eq. 86}$$

For the sake of simplicity in notation, the following abbreviations are employed as shown below:

$$\left[\frac{\partial\bar{F}_k}{\partial\bar{x}_k}\right] \triangleq \left[\frac{\partial\bar{F}_k}{\partial\bar{x}_k}\right]\Bigg|_{\substack{\bar{x}_k=\bar{x}_k^* \\ \bar{u}_k=\bar{u}_k^*}}, \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right] \triangleq \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\Bigg|_{\bar{x}_N=\bar{x}_N^*}, \left[\frac{\partial\bar{F}_k}{\partial\bar{u}_k}\right]$$
$$\triangleq \left[\frac{\partial\bar{F}_k}{\partial\bar{u}_k}\right]\Bigg|_{\substack{\bar{x}_k=\bar{x}_k^* \\ \bar{u}_k=\bar{u}_k^*}} \qquad \text{Eq. 87}$$

By substituting the conjugates of the state variables corresponding to the other time steps, as expressed in Eq. 79, into Eq. 84 represented at the final time step, equation for $d\bar{y}_N$ (Eq. 83) can be expressed as follows:

$$d\bar{y}_N = \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\left(\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{x}_{N-1}}\right]d\bar{x}_{N-1} + \left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{u}_{N-1}}\right]d\bar{u}_{N-1}\right)$$
$$d\bar{y}_N = \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{x}_{N-1}}\right]\left(\left[\frac{\partial\bar{F}_{N-2}}{\partial\bar{x}_{N-2}}\right]d\bar{x}_{N-2} + \left[\frac{\partial\bar{F}_{N-2}}{\partial\bar{u}_{N-2}}\right]d\bar{u}_{N-2}\right)$$
$$+ \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{u}_{N-1}}\right]d\bar{u}_{N-1} \qquad \text{Eq. 88}$$
$$\vdots$$
$$\vdots$$
$$d\bar{y}_N = Ad\bar{x}_1 + B_1 d\bar{u}_1 + B_2 d\bar{u}_2 + \cdots + B_{N-1}d\bar{u}_{N-1}$$

The A and B_k can be calculated as shown below:

$$A = \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{x}_{N-1}}\right]\left[\frac{\partial\bar{F}_{N-2}}{\partial\bar{x}_{N-2}}\right]\cdots\left[\frac{\partial\bar{F}_1}{\partial\bar{x}_1}\right]$$
$$B_k = \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{x}_{N-1}}\right]\left[\frac{\partial\bar{F}_{N-2}}{\partial\bar{x}_{N-2}}\right]\cdots\left[\frac{\partial\bar{F}_{k+1}}{\partial\bar{x}_{k+1}}\right]\left[\frac{\partial\bar{F}_k}{\partial\bar{u}_k}\right] = B_k^0\left[\frac{\partial\bar{F}_k}{\partial\bar{u}_k}\right] \qquad \text{Eq. 89}$$

The matrix B_k^0 can be recursively calculated as shown below:

$$B_{N-1}^0 = \left[\frac{\partial\bar{y}_N}{\partial\bar{x}_N}\right]$$
$$B_{N-2}^0 = B_{N-1}^0\left[\frac{\partial\bar{F}_{N-1}}{\partial\bar{x}_{N-1}}\right] \qquad \text{Eq. 90}$$
$$\vdots$$
$$B_k^0 = B_{k+1}^0\left[\frac{\partial\bar{F}_{k+1}}{\partial\bar{x}_{k+1}}\right]$$

As the initial condition $\bar{x}_1^*$ is always set equal to $\bar{x}_1$, $d\bar{x}_1$ becomes always zero. Therefore, the equation for $d\bar{y}_N$ (Eq. 88) can be expressed as the sum of the products of B_k and $d\bar{u}_k$ for $k = 1,2,\ldots,(N-1)$.

$$d\bar{x}_1 = 0 \qquad \text{Eq. 91}$$

$$d\bar{y}_N = B_1 d\bar{u}_1 + B_2 d\bar{u}_2 + \cdots + B_{N-1} d\bar{u}_{N-1} \qquad \text{Eq. 92}$$

$$d\bar{y}_N = \sum_{k=1}^{N-1} B_k d\bar{u}_k \qquad \text{Eq. 93}$$

In the given context, we can express the cost function, equality constraints, inequality constraints, and Hamiltonian function using the defined terms. The cost function, denoted by J expressed in Eq. 74, can be described with respect to $d\bar{u}_k$ (Eq. 82):

$$J = \frac{1}{2} \sum_{k=1}^{N-1} (\bar{u}_k^* - d\bar{u}_k)^T R_k (\bar{u}_k^* - d\bar{u}_k) \qquad \text{Eq. 94}$$

Likewise, the equality and inequality constraints can be formulated in relation to $d\bar{u}_k$. The equality constraint, which ensures that the desired output value $\bar{y}_N^d$ is reached, can be expressed as:

$$d\bar{y}_N = \bar{y}_N^d - \bar{y}_N^* \qquad \text{Eq. 95}$$

Here, $d\bar{y}_N$ represents the difference between the desired output and the actual output.

To satisfy the equality constraint, the difference in input vector history ($d\bar{y}_N$) can be related to the input increments ($d\bar{u}_k$) using the following expression:

$$d\bar{y}_N - \sum_{k=1}^{N-1} B_k d\bar{u}_k = 0 \qquad \text{Eq. 96}$$

Inequality constraints in Eq. 77 can be formulated with respect to $d\bar{u}_k$ (Eq. 82) as demonstrated below:

$$G_k(\bar{u}_k^* - d\bar{u}_k) - \bar{W}_k \leq 0, \qquad k = 1,2,\cdots,(N-1) \qquad \text{Eq. 97}$$

These inequality constraints define limits or boundaries within which the optimization variables ($d\bar{u}_k$) must operate.

Furthermore, we can express the Hamiltonian function, denoted by J^*, by using definitions in Eq. 95, Eq. 96, Eq. 97. The Hamiltonian function combines the cost function, equality constraint, and inequality constraints as demonstrated below:

$$\begin{aligned} J^*\left(d\bar{u}_k, \bar{\lambda}, \bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1}\right) \\ &= \frac{1}{2}\sum_{k=1}^{N-1}(\bar{u}_k^* - d\bar{u}_k)^T R_k (\bar{u}_k^* - d\bar{u}_k) \\ &+ \bar{\lambda}^T\left(d\bar{y}_N - \sum_{k=1}^{k=N-1} B_k d\bar{u}_k\right) \\ &+ \left(\sum_{k=1}^{N-1}\bar{\rho}_k^T(G_k(\bar{u}_k^* - d\bar{u}_k) - \bar{W}_k)\right) \end{aligned} \qquad \text{Eq. 98}$$

where $\bar{\lambda}, \bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1}$ are Lagrange multipliers associated with the equality constraint and inequality constraints, respectively. The Hamiltonian function represents the objective that needs to be minimized in the optimization process.

To facilitate the application of Hildreth's procedure, it is essential to represent the Hamiltonian function J^* solely in terms of Lagrange multipliers of inequality constraints $(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1})$. This reduction of variables enables the application of necessary conditions for optimality. Necessary conditions for optimality can be described as follows:

$$\frac{\partial J^*}{\partial \bar{\lambda}} = 0 \rightarrow \boxed{d\bar{y}_N = \sum_{k=1}^{k=N-1} B_k d\bar{u}_k} \qquad \text{Eq. 99}$$

$$\frac{\partial J^*}{\partial \bar{\rho}_k} = 0 \rightarrow \boxed{G_k(\bar{u}_k^* - d\bar{u}_k) - \bar{W}_k = 0}, \quad k = 1,2,\cdots,(N-1) \qquad \text{Eq. 100}$$

$$\frac{\partial J^*}{\partial d\bar{u}_k} = 0 \rightarrow \boxed{d\bar{u}_k = \bar{u}_k^* + R_k^{-1}\left(B_k^T\bar{\lambda} + G_k^T\bar{\rho}_k\right)}, \quad k = 1,2,\cdots,(N-1) \qquad \text{Eq. 101}$$

The related Lagrange multipliers for inequality constraints can be computed by using the equality condition for $d\bar{u}_k$:

$$\bar{\rho}_k = -(G_k R_k^{-1} G_k^T)^{-1}\left(\bar{W}_k + G_k R_k^{-1} B_k^T \bar{\lambda}\right) \qquad \text{Eq. 102}$$

By employing the equality condition for $d\bar{u}_k$ (Eq. 101) in the optimality condition concerning the equality constraints (Eq. 99), the related Lagrange multipliers can be determined as illustrated below:

$$d\bar{y}_N = \sum_{k=1}^{k=N-1} B_k \left(\bar{u}_k^* + R_k^{-1}\left(B_k^T \bar{\lambda} + G_k^T \bar{\rho}_k\right)\right)$$
$$\rightarrow \boxed{\bar{\lambda} = -A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)} \qquad \text{Eq. 103}$$

$$A_\lambda = -\left(\sum_{k=1}^{k=N-1} B_k\, R_k^{-1} B_k^T\right)^{-1}, \bar{b}_\lambda = \sum_{k=1}^{k=N-1} B_k \bar{u}_k^*\,, \bar{c}_\rho$$
$$= \sum_{k=1}^{k=N-1} B_k\, R_k^{-1} G_k^T \bar{\rho}_k \qquad \text{Eq. 104}$$

Furthermore, the expression for $d\bar{u}_k$ (Eq. 101) can be updated as follows:

$$d\bar{u}_k = \bar{E}_k + R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + R_k^{-1} G_k^T \bar{\rho}_k,$$
$$\bar{E}_k = \bar{u}_k^* - R_k^{-1} B_k^T A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda\right) \qquad \text{Eq. 105}$$

By applying the optimality condition as depicted in the above equations, the variables $d\bar{u}_k$ and $\bar{\lambda}$ are computed in terms of the Lagrange multipliers $(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1})$ associated with the input inequality constraints. Consequently, the Hamiltonian function J^* in Eq. 98 can be expressed in the same form $(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1})$ as shown below:

$$J^*(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1})$$
$$= \bar{C} - \frac{1}{2}\bar{c}_\rho^T A_\lambda \bar{c}_\rho - \left(-d\bar{y}_N^T + \bar{b}_\lambda^T\right) A_\lambda \bar{c}_\rho$$
$$-\frac{1}{2}\sum_{k=1}^{N-1} \bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k - \sum_{k=1}^{N-1} \bar{\rho}_k^T\, \bar{W}_k \qquad \text{Eq. 106}$$

Inserting $\bar{c}_\rho$ expressed in Eq. 104 into above cost function equation and assuming R_k is symmetric matrix yields:

$$\begin{aligned} J^* = \bar{C} + \frac{1}{2}\sum_{k=1}^{N-1}\left(d\bar{u}_k^T R_k d\bar{u}_k - 2\bar{u}_k^{*T} R_k d\bar{u}_k\right) \\ + \bar{\lambda}^T\left(d\bar{y}_N - \sum_{k=1}^{k=N-1} B_k d\bar{u}_k\right) \\ + \left(\sum_{k=1}^{N-1}\bar{\rho}_k^T(G_k(\bar{u}_k^* - d\bar{u}_k) - \bar{W}_k)\right) \end{aligned} \tag{Eq. 107}$$

$$\begin{aligned} J^* = \bar{C} + \frac{1}{2}\sum_{k=1}^{N-1} d\bar{u}_k^T R_k d\bar{u}_k - \frac{1}{2}\sum_{k=1}^{N-1} 2\bar{u}_k^{*T} R_k d\bar{u}_k \\ + \bar{\lambda}^T\left(d\bar{y}_N - \sum_{k=1}^{N-1} B_k d\bar{u}_k\right) \\ + \left(\sum_{k=1}^{N-1}\bar{\rho}_k^T(G_k\bar{u}_k^* - \bar{W}_k)\right) - \sum_{k=1}^{N-1}\bar{\rho}_k^T G_k d\bar{u}_k \end{aligned} \tag{Eq. 108}$$

Inserting $\bar{\lambda} = -A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)$ into Eq. 108 and evaluating $-\left(d\bar{y}_N - \bar{b}_\lambda\right)A_\lambda^T d\bar{y}_N$ term in the constant $\bar{C}$ term gives:

$$\begin{aligned} J^* = \bar{C} + \frac{1}{2}\sum_{k=1}^{N-1} d\bar{u}_k^T R_k d\bar{u}_k - \frac{1}{2}\sum_{k=1}^{N-1} 2\bar{u}_k^{*T} R_k d\bar{u}_k \\ + \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T\left(A_\lambda^T\sum_{k=1}^{N-1} B_k d\bar{u}_k\right) \\ + \bar{c}_\rho A_\lambda^T d\bar{y}_N + \left(\sum_{k=1}^{N-1}\bar{\rho}_k^T(G_k\bar{u}_k^* - \bar{W}_k)\right) \\ - \sum_{k=1}^{N-1}\bar{\rho}_k^T G_k d\bar{u}_k \end{aligned} \tag{Eq. 109}$$

Note that:

$$d\bar{u}_k = \bar{E}_k + R_k^{-1}B_k^T A_\lambda \bar{c}_\rho + R_k^{-1}G_k^T\bar{\rho}_k \tag{Eq. 110}$$

$$\frac{1}{2}\sum_{k=1}^{N-1} d\bar{u}_k^T R_k d\bar{u}_k$$
$$= \frac{1}{2}\sum_{k=1}^{N-1} \bar{E}_k^T R_k \bar{E}_k + \sum_{k=1}^{N-1} \bar{E}_k^T (B_k^T A_\lambda \bar{c}_\rho + G_k^T \bar{\rho}_k) \qquad \text{Eq. 111}$$
$$+ \frac{1}{2}\sum_{k=1}^{N-1} \big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho$$
$$+ R_k^{-1} G_k^T \bar{\rho}_k\big)^T R_k \big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + R_k^{-1} G_k^T \bar{\rho}_k\big)$$

$$\frac{1}{2}\sum_{k=1}^{N-1} d\bar{u}_k^T R_k d\bar{u}_k$$
$$= \frac{1}{2}\sum_{k=1}^{N-1} \bar{E}_k^T R_k \bar{E}_k + \sum_{k=1}^{N-1} \bar{E}_k^T (B_k^T A_\lambda \bar{c}_\rho + G_k^T \bar{\rho}_k)$$
$$+ \frac{1}{2}\sum_{k=1}^{N-1} \Big(\big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big)^T R_k \big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big)\Big) \qquad \text{Eq. 112}$$
$$+ \frac{1}{2}\sum_{k=1}^{N-1} \Big((R_k^{-1} G_k^T \bar{\rho}_k)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k)\Big)$$
$$+ \sum_{k=1}^{N-1} \Big(\big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k)\Big)$$

By putting the term $\frac{1}{2}\sum_{k=1}^{N-1} d\bar{u}_k^T R_k d\bar{u}_k$ in Eq. 112 into Eq. 109 and evaluating $\frac{1}{2}\sum_{k=1}^{N-1} \bar{E}_k^T R_k \bar{E}_k$ term in the constant $\bar{C}$ term, the expression for J^*gives:

$$
\begin{aligned}
J^* = \bar{C} + \sum_{k=1}^{N-1} \bar{E}_k^T (B_k^T A_\lambda \bar{c}_\rho + G_k^T \bar{\rho}_k) \\
+ \frac{1}{2} \sum_{k=1}^{N-1} \left(\left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \right)^T R_k \left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \right) \right) \\
+ \frac{1}{2} \sum_{k=1}^{N-1} \left((R_k^{-1} G_k^T \bar{\rho}_k)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k) \right) \\
+ \sum_{k=1}^{N-1} \left(\left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \right)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k) \right) \\
- \frac{1}{2} \sum_{k=1}^{N-1} 2 \bar{u}_k^{*T} R_k d\bar{u}_k \\
+ \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho \right)^T \left(A_\lambda^T \sum_{k=1}^{N-1} B_k d\bar{u}_k \right) + \bar{c}_\rho A_\lambda^T d\bar{y}_N \\
+ \left(\sum_{k=1}^{N-1} \bar{\rho}_k^T (G_k \bar{u}_k^* - \bar{W}_k) \right) - \sum_{k=1}^{N-1} \bar{\rho}_k^T G_k d\bar{u}_k
\end{aligned}
\qquad \text{Eq. 113}
$$

Inserting Eq. 110 into Eq. 113 and evaluating $\sum_{k=1}^{N-1} \left(2\bar{u}_k^{*T} R_k \bar{E}_k \right)$ in constant $\bar{C}$ term gives the modified expression for J^*:

$$\begin{aligned}
J^{*} = \bar{C} &+ \sum_{k=1}^{N-1} \bar{E}_k^T \left(B_k^T A_\lambda \bar{c}_\rho + G_k^T \bar{\rho}_k\right) \\
&+ \frac{1}{2} \sum_{k=1}^{N-1} \Big(\big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big)^T R_k \big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big) \Big) \\
&+ \frac{1}{2} \sum_{k=1}^{N-1} \Big((R_k^{-1} G_k^T \bar{\rho}_k)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k) \Big) \\
&+ \sum_{k=1}^{N-1} \Big(\big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\big)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k) \Big) \\
&- \frac{1}{2} \sum_{k=1}^{N-1} \Big(2\bar{u}_k^{*T} R_k \big(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + R_k^{-1} G_k^T \bar{\rho}_k\big) \Big) \qquad \text{Eq. 114} \\
&+ \big(d\bar{y}_N - \bar{b}_\lambda \\
&- \bar{c}_\rho\big)^T \Bigg(A_\lambda^T \sum_{k=1}^{N-1} \big(B_k \bar{E}_k + B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \\
&+ B_k R_k^{-1} G_k^T \bar{\rho}_k\big) \Bigg) + \bar{c}_\rho A_\lambda^T d\bar{y}_N \\
&+ \left(\sum_{k=1}^{N-1} \bar{\rho}_k^T (G_k \bar{u}_k^{*} - \bar{W}_k) \right) \\
&- \sum_{k=1}^{N-1} \Big(\bar{\rho}_k^T G_k \big(\bar{E}_k + R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + R_k^{-1} G_k^T \bar{\rho}_k\big) \Big)
\end{aligned}$$

The term $\big(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\big)^T \big(A_\lambda^T \sum_{k=1}^{N-1} \big(B_k \bar{E}_k + B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + B_k R_k^{-1} G_k^T \bar{\rho}_k\big)\big)$ can be simplified by grouping terms as shown in Eq. 115:

$$\left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T \left(A_\lambda^T \sum_{k=1}^{N-1} \left(B_k \bar{E}_k + B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho + B_k R_k^{-1} G_k^T \bar{\rho}_k \right) \right)$$

$$= \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T A_\lambda^T \left(\sum_{k=1}^{N-1} B_k \bar{E}_k + \sum_{k=1}^{N-1} \left(B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \right) + \underbrace{\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k}_{\bar{c}_\rho} \right) \quad \text{Eq. 115}$$

$$= \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T A_\lambda^T \left(\sum_{k=1}^{N-1} B_k \bar{E}_k + \sum_{k=1}^{N-1} \left(B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho \right) + \bar{c}_\rho \right)$$

The expression for J^* can be further elaborated as follows:

$$
\begin{aligned}
J^* = \bar{C} + \sum_{k=1}^{N-1} (& \underbrace{\bar{E}_k^T B_k^T A_\lambda \bar{c}_\rho}_{Term1\_1 \times Term1\_2} + \underbrace{\bar{E}_k^T G_k^T \bar{\rho}_k}_{Term2}) \\
& + \frac{1}{2} \sum_{k=1}^{N-1} \left(\left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\right)^T R_k \left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\right) \right) \\
& + \frac{1}{2} \sum_{k=1}^{N-1} \left((R_k^{-1} G_k^T \bar{\rho}_k)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k) \right) \\
& + \sum_{k=1}^{N-1} \left(\underbrace{\left(R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\right)^T R_k (R_k^{-1} G_k^T \bar{\rho}_k)}_{Term3} \right) \\
& - \sum_{k=1}^{N-1} \left(\left(\bar{u}_k^{*T} B_k^T A_\lambda \bar{c}_\rho + \underbrace{\bar{u}_k^{*T} G_k^T \bar{\rho}_k}_{Term4} \right) \right) \\
& + \left(d\bar{y}_N - \bar{b}_\lambda \underbrace{-\bar{c}_\rho}_{Term1\_1} \right)^T \left(\sum_{k=1}^{N-1} \underbrace{A_\lambda^T B_k \bar{E}_k}_{Term1\_2} \right. \\
& \left. + \sum_{k=1}^{N-1} A_\lambda^T \left(B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\right) + A_\lambda^T \bar{c}_\rho \right) + \bar{c}_\rho A_\lambda^T d\bar{y}_N \\
& + \left(\sum_{k=1}^{N-1} \left(\underbrace{\bar{\rho}_k^T G_k \bar{u}_k^*}_{Term4} - \bar{\rho}_k^T \bar{W}_k \right) \right) \\
& - \sum_{k=1}^{N-1} \left(\underbrace{\bar{\rho}_k^T G_k \bar{E}_k}_{Term2} + \underbrace{\bar{\rho}_k^T G_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho}_{Term3} \right. \\
& \left. + \bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k \right)
\end{aligned}
\tag{Eq. 116}
$$

The terms indicated in above equations cancels each other and resulting in a simplified form of J^*:

$$\begin{aligned} J^* = \bar{C} + \frac{1}{2}\sum_{k=1}^{N-1}\left(\bar{c}_\rho^T A_\lambda^T B_k (R_k^{-1})^T B_k^T A_\lambda \bar{c}_\rho\right) \\ - \frac{1}{2}\sum_{k=1}^{N-1}(\bar{\rho}_k^T G_k (R_k^{-1})^T G_k^T \bar{\rho}_k) - \sum_{k=1}^{N-1}\left(\bar{u}_k^{*\,T} B_k^T A_\lambda \bar{c}_\rho\right) \\ + \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T A_\lambda^T \left(\sum_{k=1}^{N-1}\left(B_k R_k^{-1} B_k^T A_\lambda \bar{c}_\rho\right)\right) \\ - \sum_{k=1}^{N-1} \bar{\rho}_k^T \bar{W}_k - \left(\bar{b}_\lambda + \bar{c}_\rho\right)^T A_\lambda^T \bar{c}_\rho + 2\bar{c}_\rho^T A_\lambda^T d\bar{y}_N \end{aligned} \qquad \text{Eq. 117}$$

Assuming R_k is a symmetric matrix, it follows that $R_k^{-1} = (R_k^{-1})^T$. The objective function J^*can be expressed as follows:

$$\begin{aligned} J^* = \bar{C} + \frac{1}{2}\bar{c}_\rho^T A_\lambda^T \overbrace{\left[\sum_{k=1}^{N-1}(B_k R_k^{-1} B_k^T)\right]}^{-A_\lambda^{-1}} A_\lambda \bar{c}_\rho \\ - \frac{1}{2}\sum_{k=1}^{N-1}(\bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k) - \overbrace{\left[\sum_{k=1}^{N-1}\left(\bar{u}_k^{*\,T} B_k^T\right)\right]}^{b_\lambda^T} A_\lambda \bar{c}_\rho \\ + \left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)^T A_\lambda^T \overbrace{\left[\sum_{k=1}^{N-1}(B_k R_k^{-1} B_k^T)\right]}^{-A_\lambda^{-1}} A_\lambda \bar{c}_\rho \\ - \left(\bar{b}_\lambda + \bar{c}_\rho\right)^T A_\lambda^T \bar{c}_\rho + 2\bar{c}_\rho^T A_\lambda^T d\bar{y}_N - \sum_{k=1}^{N-1} \bar{\rho}_k^T \bar{W}_k \end{aligned} \qquad \text{Eq. 118}$$

Since $(B_k R_k^{-1} B_k^T)$ is symmetric due to R_k being symmetric, the summation $\sum_{k=1}^{N-1}(B_k R_k^{-1} B_k^T)$ is also symmetric. Therefore, A_λ^{-1} and A_λ is also symmetric, resulting in $A_\lambda = A_\lambda^T$. The expression of J^* in Eq. 118 can be further simplified as:

$$\begin{aligned} J^*(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1}) & \\ &= \bar{C} - \frac{1}{2}\bar{c}_\rho^T A_\lambda \bar{c}_\rho - \left(-d\bar{y}_N^T + \bar{b}_\lambda^T\right) A_\lambda \bar{c}_\rho \\ &- \frac{1}{2}\sum_{k=1}^{N-1} \bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k - \sum_{k=1}^{N-1} \bar{\rho}_k^T \bar{W}_k \end{aligned}$$

Eq. 119

It is possible to further simplify Eq. 119 as follows:

$$\begin{aligned} J^*(\bar{\rho}_1, \bar{\rho}_2, \cdots, \bar{\rho}_{N-1}) & \\ &= \bar{C} - \underbrace{\frac{1}{2}\left(\sum_{k=1}^{N-1} \bar{\rho}_k^T G_k R_k^{-1} B_k^T\right) A_\lambda \left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right)}_{Term1} \\ &- \underbrace{\left(-d\bar{y}_N{}^T + \bar{b}_\lambda^T\right) A_\lambda \left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right)}_{Term2} \\ &- \frac{1}{2}\underbrace{\sum_{k=1}^{N-1} (\bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k)}_{Term3} - \underbrace{\sum_{k=1}^{N-1} \bar{\rho}_k^T \bar{W}_k}_{Term4} \end{aligned}$$

Eq. 120

Term1 in Eq. 120 can be expressed as follows:

$$\begin{aligned} &\left(\sum_{k=1}^{N-1} \bar{\rho}_k^T G_k R_k^{-1} B_k^T\right) A_\lambda \left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right) \\ &\qquad = (\bar{\rho}_1^T G_1 R_1^{-1} B_1^T + \cdots + \bar{\rho}_{N-1}^T G_{N-1} R_{N-1}^{-1} B_{N-1}^T) A_\lambda (B_1 R_1^{-1} G_1^T \bar{\rho}_1 + \cdots + B_{N-1} R_{N-1}^{-1} G_{N-1}^T \bar{\rho}_{N-1}) \end{aligned}$$

$$= \begin{bmatrix} \bar{\rho}_1^T \\ \bar{\rho}_2^T \\ \vdots \\ \bar{\rho}_{N-1}^T \end{bmatrix}^T \begin{bmatrix} (G_1 R_1^{-1} B_1^T) A_\lambda (B_1 R_1^{-1} G_1^T) & (G_1 R_1^{-1} B_1^T) A_\lambda (B_2 R_2^{-1} G_2^T) & \cdots & (G_1 R_1^{-1} B_1^T) A_\lambda (B_{N-1} R_{N-1}^{-1} G_{N-1}^T) \\ (G_2 R_2^{-1} B_2^T) A_\lambda (B_1 R_1^{-1} G_1^T) & (G_2 R_2^{-1} B_2^T) A_\lambda (B_2 R_2^{-1} G_2^T) & \cdots & (G_2 R_2^{-1} B_2^T) A_\lambda (B_{N-1} R_{N-1}^{-1} G_{N-1}^T) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & (G_{N-1} R_{N-1}^{-1} B_{N-1}^T) A_\lambda (B_{N-1} R_{N-1}^{-1} G_{N-1}^T) \end{bmatrix} \begin{bmatrix} \bar{\rho}_1 \\ \bar{\rho}_2 \\ \vdots \\ \bar{\rho}_{N-1} \end{bmatrix}$$

Eq. 121

$$= \bar{\rho}^T Z \bar{\rho}$$

$$Z = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,N-1} \\ Z_{2,1} & Z_{2,2} & & \\ & & Z_{i,j} & \\ & & & \end{bmatrix}$$

$$Z_{m,n} = (G_m R_m^{-1} B_m^T) A_\lambda (B_n R_n^{-1} G_n^T)$$

Term2 in Eq. 120 can be expressed as follows:

$$\left(-d\bar{y}_N{}^T + \bar{b}_\lambda^T\right)A_\lambda\left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right) = \left(-d\bar{y}_N{}^T + \bar{b}_\lambda^T\right)A_\lambda(B_1 R_1^{-1} G_1^T \bar{\rho}_1 + \cdots + B_{N-1} R_{N-1}^{-1} G_{N-1}^T \bar{\rho}_{N-1}) \quad \text{Eq. 122}$$

Since A_λ is symmetric:

$$\left(-d\bar{y}_N{}^T + \bar{b}_\lambda^T\right)A_\lambda\left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right) = \begin{bmatrix} \bar{\rho}_1 \\ \bar{\rho}_2 \\ \vdots \\ \bar{\rho}_{N-1} \end{bmatrix}^T \begin{bmatrix} G_1 R_1^{-1} B_1^T \\ \vdots \\ G_{N-1} R_{N-1}^{-1} B_{N-1}^T \end{bmatrix} A_\lambda\left(d\bar{y}_N + \bar{b}_\lambda\right) = \bar{\rho}^T D \quad \text{Eq. 123}$$

$$\left(-d\bar{y}_N{}^T + \bar{b}_\lambda^T\right)A_\lambda\left(\sum_{k=1}^{N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k\right) = \bar{\rho}^T D \quad \text{Eq. 124}$$

$$D = \begin{bmatrix} G_1 R_1^{-1} B_1^T \\ \vdots \\ G_{N-1} R_{N-1}^{-1} B_{N-1}^T \end{bmatrix} A_\lambda\left(d\bar{y}_N + \bar{b}_\lambda\right) \quad \text{Eq. 125}$$

Term3 in Eq. 120 can be expressed as follows:

$$\sum_{k=1}^{N-1} (\bar{\rho}_k^T G_k R_k^{-1} G_k^T \bar{\rho}_k) = \bar{\rho}^T X \bar{\rho}, \quad X = \begin{bmatrix} G_1 R_1^{-1} G_1^T & 0 & \cdots & 0 \\ 0 & G_2 R_2^{-1} G_2^T & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & G_{N-1} R_{N-1}^{-1} G_{N-1}^T \end{bmatrix} \quad \text{Eq. 126}$$

Term4 in Eq. 120 can be expressed as follows:

$$\sum_{k=1}^{N-1} \bar{\rho}_k^T \bar{W}_k = \bar{\rho}^T \bar{E}, \bar{E} = \begin{bmatrix} \bar{W}_1 \\ \vdots \\ \bar{W}_{N-1} \end{bmatrix} \quad \text{Eq. 127}$$

When the Lagrange multipliers associated with the input inequality constraints at all time steps are combined into a single vector, J^* can be expressed in the form shown below, achieving a concise representation:

$$J^*(\bar{\rho}) = \bar{C} - \frac{1}{2}\bar{\rho}^T \bar{Z} \bar{\rho} - \bar{\rho}^T \overline{D} - \bar{\rho}^T \bar{X} \bar{\rho} - \bar{\rho}^T \overline{E}$$

$$J^*(\bar{\rho}) = \bar{C} - \frac{1}{2}\bar{\rho}^T \underbrace{(\bar{Z} + \bar{X})}_{H} \bar{\rho} - \bar{\rho}^T \underbrace{(\overline{D} + \overline{E})}_{\overline{K}} = \bar{C} - \frac{1}{2}\bar{\rho}^T H \bar{\rho} - \bar{\rho}^T \overline{K} \qquad \text{Eq. 128}$$

$$\bar{\rho} = \begin{bmatrix} \bar{\rho}_1 \\ \bar{\rho}_2 \\ \vdots \\ \bar{\rho}_{N-1} \end{bmatrix}, where\ \ \bar{\rho}_m \in \mathcal{R}^4, \bar{\rho} \in \mathcal{R}^{4(N-1)}\ for\ \ m = 1,2,\cdots,(N-1) \qquad \text{Eq. 129}$$

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,(N-1)} \\ H_{2,1} & \vdots & \vdots & \vdots \\ \vdots & \vdots & H_{m,n} & \vdots \\ H_{(N-1),1} & H_{(N-1),2} & \cdots & H_{(N-1),(N-1)} \end{bmatrix}, H_{m,n} = \begin{cases} (G_m R_m^{-1} B_m^T) A_\lambda (B_m R_m^{-1} G_m^T) + G_m R_m^{-1} G_m^T, & m = n \\ (G_m R_m^{-1} B_m^T) A_\lambda (B_m R_m^{-1} G_m^T), & m \neq n \end{cases} \qquad \text{Eq. 130}$$

$$\overline{K} = \begin{bmatrix} \overline{K}_1 \\ \vdots \\ \overline{K}_m \\ \vdots \\ \overline{K}_{N-1} \end{bmatrix}, \overline{K}_m = (G_m R_m^{-1} B_m^T) A_\lambda \left(-d\bar{y}_N + \bar{b}_\lambda\right) + \overline{W}_m \qquad \text{Eq. 131}$$

$$where\ H_{m,n} \in \mathcal{R}^{4\times4}, H \in \mathcal{R}^{4(N-1)\times4(N-1)}, \overline{K}_m \in \mathcal{R}^4, \overline{K} \in \mathcal{R}^{4(N-1)\times1}\ for\ m,n = 1,2,\cdots,(N-1)$$

The components of the Hamiltonian function that are not dependent on Lagrange multipliers are combined into the term $\bar{C}$. As this term remains constant, it is disregarded in the optimality computations.

$$J^*(\bar{\rho}) = \frac{1}{2}(\bar{\rho}^T H \bar{\rho}) + \bar{\rho}^T(\overline{K}) \qquad \text{Eq. 132}$$

4.2.2 Implementation of Hildreth Procedure

To solve the modified equations and determine the optimal guidance commands, the Hildreth procedure, which leverages the principles of Lagrange multipliers, is utilized. This procedure effectively addresses the input constraint by determining the appropriate values of the input Lagrange multipliers that satisfy the system constraints while optimizing the overall guidance objective. The utilization of the Hildreth procedure further contributes to the rigor and effectiveness of the guidance algorithm design.

Once the matrices H and $\overline{K}$ are constructed, the Lagrange multipliers vector $\bar{\rho}$, the Hamiltonian H and the $\overline{K}$ matrix are reformulated using their scalar elements for ease of computation:

$$\bar{\rho} = \begin{bmatrix} \vdots \\ \rho_i \\ \vdots \end{bmatrix}, for\ i = 1,2,\cdots,4(N-1) \qquad \text{Eq. 133}$$

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,4(N-1)} \\ h_{2,1} & \vdots & \vdots & \vdots \\ \vdots & \vdots & h_{i,j} & \vdots \\ h_{4(N-1),1} & h_{4(N-1),2} & \cdots & h_{4(N-1),4(N-1)} \end{bmatrix}, \overline{K} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_i \\ \vdots \end{bmatrix}, for\ i,j = 1,2,\cdots,4(N-1) \qquad \text{Eq. 134}$$

Since H is a symmetric matrix, it implies that $H_{m,n}$ is symmetric for all values of (m,n). Consequently,$h_{i,j} = h_{j,i}$. By utilizing the optimality conditions for ρ_i:

$$w_i = \frac{\partial J^*(\bar{\rho})}{\partial \rho_i} = 0 \qquad \text{Eq. 135}$$

$$k_i + h_{i,i}\rho_i + [h_{i,1} \quad \cdots \quad h_{(i-1),1}]\begin{bmatrix}\rho_1\\ \rho_2\\ \vdots\\ \rho_{i-1}\end{bmatrix} + [h_{(i+1),1} \quad \cdots \quad h_{4(N-1),1}]\begin{bmatrix}\rho_{i+1}\\ \rho_{i+2}\\ \vdots\\ \rho_{4(N-1)}\end{bmatrix} = 0 \qquad \text{Eq. 136}$$

$$k_i + h_{i,i}\rho_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j = 0 \qquad \text{Eq. 137}$$

$$w_i^{r+1} = -\frac{1}{h_{ii}}\left(k_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j^{r+1} + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j^{r}\right) \qquad \text{Eq. 138}$$

It should be noted that r is $\bar{\rho}$ iteration number. Lagrange multiplier vector at 2nd iteration $\bar{\rho}^2$ is obtained from $\bar{\rho}^1$ by minimizing the cost function with respect to ρ_i^1 for $i = 1,2,\cdots,4(N-1)$. Similarly, $\bar{\rho}^{r+1}$ is obtained from $\bar{\rho}^r$ by setting $\frac{\partial J^*(\bar{\rho})}{\partial \rho_i} = 0$. A continuous function can be defined for the relation between $\bar{\rho}^r$ and $\bar{\rho}^{r+1}$ as shown below:

$$O(\bar{\rho}^r) = \bar{\rho}^{r+1} \qquad \text{Eq. 139}$$

The operator O_i updates the ith element of the Lagrange multiplier vector $\bar{\rho}$ which has $\bar{\rho}^{r+1}$ in the first $i-1$ elements and $\bar{\rho}^r$ in the last $(4N-4)-(i+1)$ elements. In order to complete rth iteration and obtain Lagrange multiplier values at $(r+1)$ iteration step, one complete cycle for $i = 1,2,\cdots,4(N-1)$ must be finished.

$$O_i(\bar{\rho}) = O_i\left(\begin{bmatrix}\rho_1^{r+1}\\ \rho_2^{r+1}\\ \vdots\\ \rho_i^{r}\\ \rho_{i+1}^{r}\\ \rho_{i+2}^{r}\\ \vdots\\ \vdots\\ \rho_{4N-4}^{r}\end{bmatrix}\right) = \begin{bmatrix}\rho_1^{r+1}\\ \rho_2^{r+1}\\ \vdots\\ \rho_i^{r+1}\\ \rho_{i+1}^{r}\\ \rho_{i+2}^{r}\\ \vdots\\ \vdots\\ \rho_{4N-4}^{r}\end{bmatrix} \qquad \text{Eq. 140}$$

$$\rho_i^{r+1} = O_i(\rho_i^r) = \max(0, w_i^{r+1}) \qquad \text{Eq. 141}$$

Hildreth's quadratic programming procedure [10] is employed in an iterative manner to identify and eliminate inactive inequality constraints as outlined below:

$$(\bar{\rho}^r) = \bar{\rho}^{r+1} = \begin{bmatrix} \rho_1^{r+1} \\ \rho_2^{r+1} \\ \vdots \\ \rho_i^{r+1} \\ \vdots \\ \rho_{4N-4}^{r+1} \end{bmatrix} = O_{4N-4}(\bar{\rho}) \ldots\ldots O_2(\bar{\rho})O_1(\bar{\rho}) \qquad \text{Eq. 142}$$

$$O_i(\rho_i^r) = \rho_i^{r+1}$$

$$= \begin{cases} 0, & if -\frac{1}{h_{i,i}}\left(k_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j^{r+1} + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j^r\right) < 0 \\ -\frac{1}{h_{i,i}}\left(k_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j^{r+1} + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j^r\right), & if -\frac{1}{h_{i,i}}\left(k_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j^{r+1} + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j^r\right) \geq 0 \end{cases}$$

$$\rho_i^r \rightarrow Lagrange\ multipliers\ for\ the\ r^{th}\ iteration$$

The initial iterations begin with assuming that all constraints are inactive. Consequently, the Lagrange multipliers are initialized with zero values as follows:

$$\rho_i^0 = 0, for\ \ i = 1,2,\cdots,4(N-1) \qquad \text{Eq. 143}$$

The iteration process halts at the rth iteration step when when the specified condition is met.

$$\rho_i^r = \rho_i^{r-1}, for\ \ i = 1,2,\cdots,4(N-1) \qquad \text{Eq. 144}$$

The specified termination condition, as indicated in Eq. 145, is applied with a certain convergence criterion (ε_ρ) due to numerical errors and the resolutions of data types:

$$\left|\frac{\rho_i^r - \rho_i^{r1}}{\rho_i^{r-1}}\right| \leq \varepsilon_\rho \qquad \text{Eq. 145}$$

4.3 Application

Figure 13 illustrates the flowchart of the midcourse guidance approach, as discussed in Section 4.1. Initially, an input vector history is generated based on PN-based simulation. This history serves as the basis for improving the solution to achieve the objective outlined in Section 4.1. Consequently, an iterative solution procedure is employed, where an error history of the control variable is computed at each iteration to facilitate the achievement of the objective. In this study, the iterations are terminated either upon the convergence of the solution (ε_y), which measures the proximity between the final and desired terminal positions, or when the maximum limit of iteration steps (i_{MPC}^{max}) is reached in terms of the number of iterations.

In cases where convergence to a solution is not attained within the specified error band ε_y, indicating the inability to satisfy the output equality constraint, the computed current inputs are used to proceed to the next recursive step in the unconstrained scenario. If the upper limit of the recursive step count (i_{MPC}) is reached and the output equality constraint still cannot be approached, it signifies the inability to obtain a solution even in the absence of input constraints.

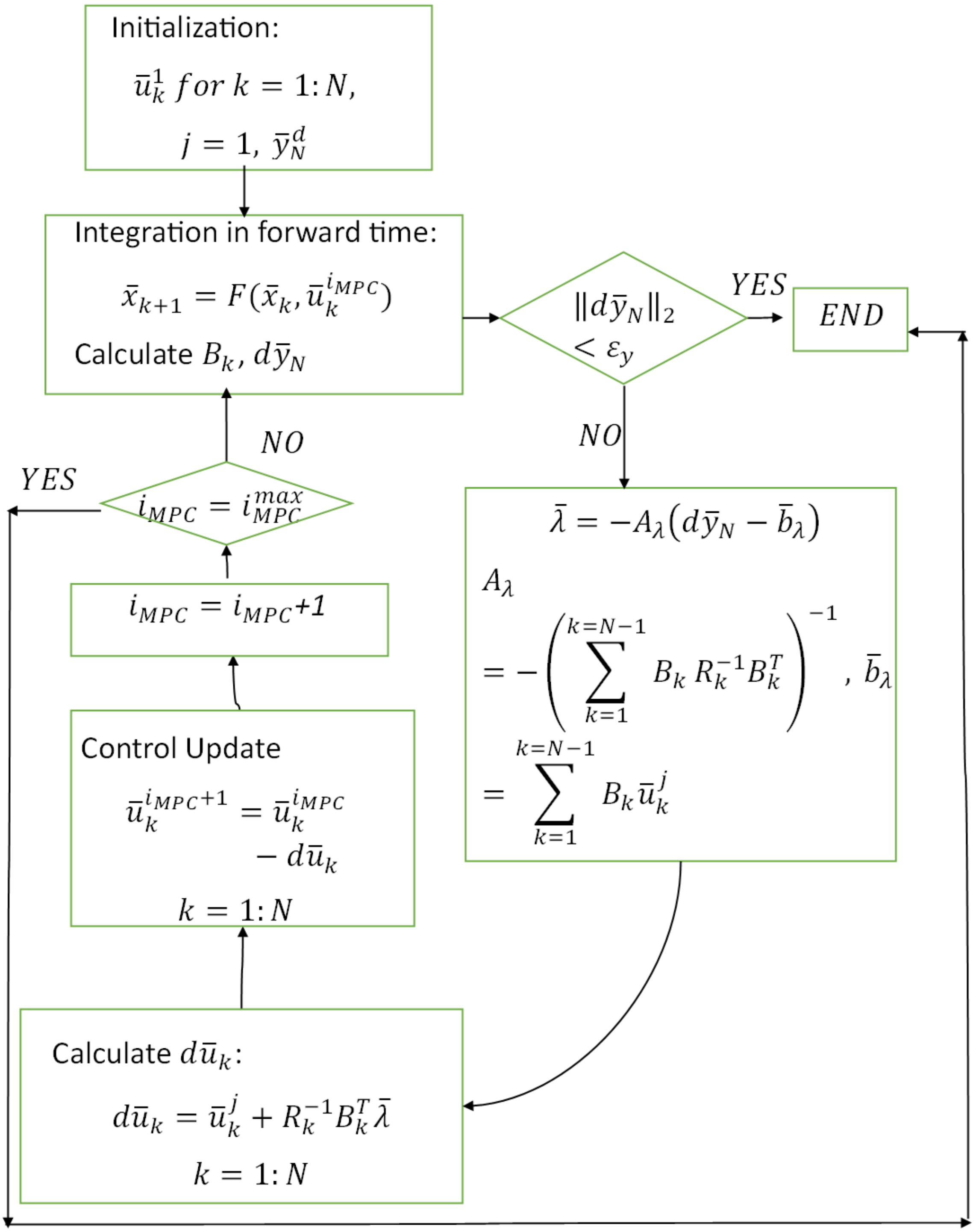

Initialization:
$\bar{u}_k^1$ for $k = 1{:}N,$
$j = 1,\ \bar{y}_N^d$
Integration in forward time:
$\bar{x}_{k+1} = F(\bar{x}_k, \bar{u}_k^{i_{MPC}})$
Calculate $B_k, d\bar{y}_N$
$\|d\bar{y}_N\|_2 < \varepsilon_y$
YES
END
NO
$\bar{\lambda} = -A_\lambda(d\bar{y}_N - \bar{b}_\lambda)$
$A_\lambda = -\left(\sum_{k=1}^{k=N-1} B_k R_k^{-1} B_k^T\right)^{-1}$, $\bar{b}_\lambda = \sum_{k=1}^{k=N-1} B_k \bar{u}_k^j$
NO
YES
$i_{MPC} = i_{MPC}^{max}$
$i_{MPC} = i_{MPC}+1$
Control Update
$\bar{u}_k^{i_{MPC}+1} = \bar{u}_k^{i_{MPC}} - d\bar{u}_k$
$k = 1{:}N$
Calculate $d\bar{u}_k$:
$d\bar{u}_k = \bar{u}_k^j + R_k^{-1} B_k^T \bar{\lambda}$
$k = 1{:}N$

Figure 13: Optimization Steps for Unconstrained Input Case

Figure 14 presents the flowchart of the midcourse guidance approach, which is discussed in Section 4.2. The effectiveness of the linearization processes utilized in this method, applicable to nonlinear systems, diminishes as the solution deviates from the assumed solution. Therefore, the optimization problem primarily focuses on solving the case where input constraints are disregarded. Failure to do so may result in a significantly reduced convergence rate of the numerical solution.

If the control input solution $(\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_{N-1})$ converges to a solution within the specified error band $\varepsilon_y$without activating the input constraints at any time step, it can be concluded that an optimization solution complying with the input constraints has been achieved. In the event that the upper limit of the recursive step count (i_{MPC}^{max}) is reached while the output equality constraint still cannot be approached, it signifies the inability to obtain a solution even in the absence of input constraints. Under such circumstances, it is inferred that achieving an optimization solution under input constraints is infeasible.

In cases where the control input solution obtained without considering input constraints activates the input constraints at one or more time steps, efforts are made to obtain a solution for the input-constrained case using Hildreth's algorithm in the current and subsequent recursive steps.

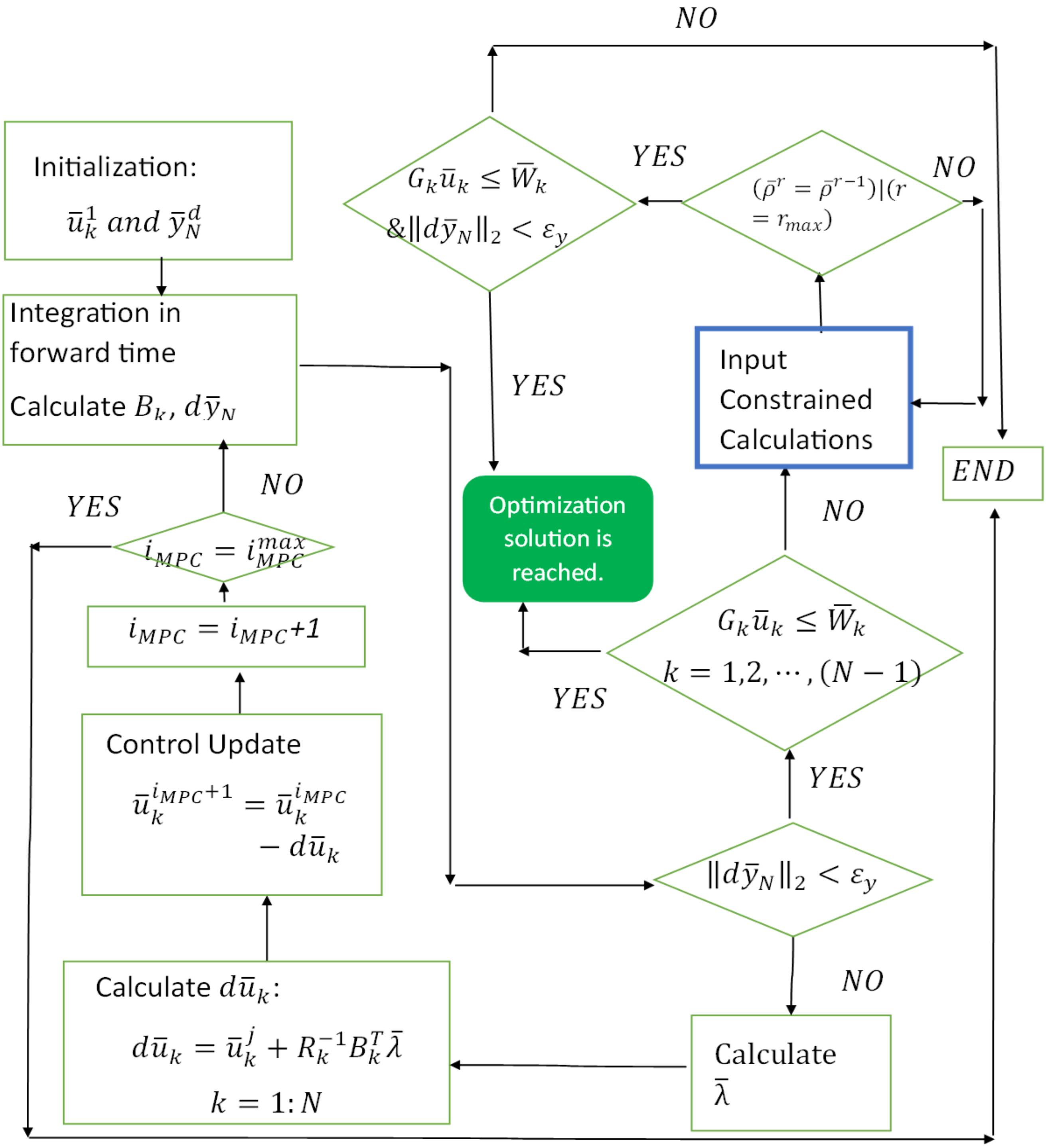

Initialization:
$\bar{u}_k^1$ and $\bar{y}_N^d$
Integration in forward time
Calculate B_k, $d\bar{y}_N$
NO
$G_k\bar{u}_k \leq \bar{W}_k$ & $\|d\bar{y}_N\|_2 < \varepsilon_y$
YES
$(\bar{\rho}^r = \bar{\rho}^{r-1})|(r = r_{max})$
NO
YES
Input Constrained Calculations
END
Optimization solution is reached.
YES
NO
$i_{MPC} = i_{MPC}^{max}$
$i_{MPC} = i_{MPC}+1$
NO
$G_k\bar{u}_k \leq \bar{W}_k$
$k = 1,2,\cdots,(N-1)$
YES
Control Update
$\bar{u}_k^{i_{MPC}+1} = \bar{u}_k^{i_{MPC}} - d\bar{u}_k$
YES
$\|d\bar{y}_N\|_2 < \varepsilon_y$
NO
Calculate $d\bar{u}_k$:
$d\bar{u}_k = \bar{u}_k^j + R_k^{-1}B_k^T\bar{\lambda}$
$k = 1:N$
Calculate $\bar{\lambda}$

Figure 14: Optimization Steps for Input Constraint Case

The equations pertaining to input-constrained calculations are depicted in Figure 15. The comprehensive derivations and explanations of these equations can be found in Section 4.2 of the thesis.

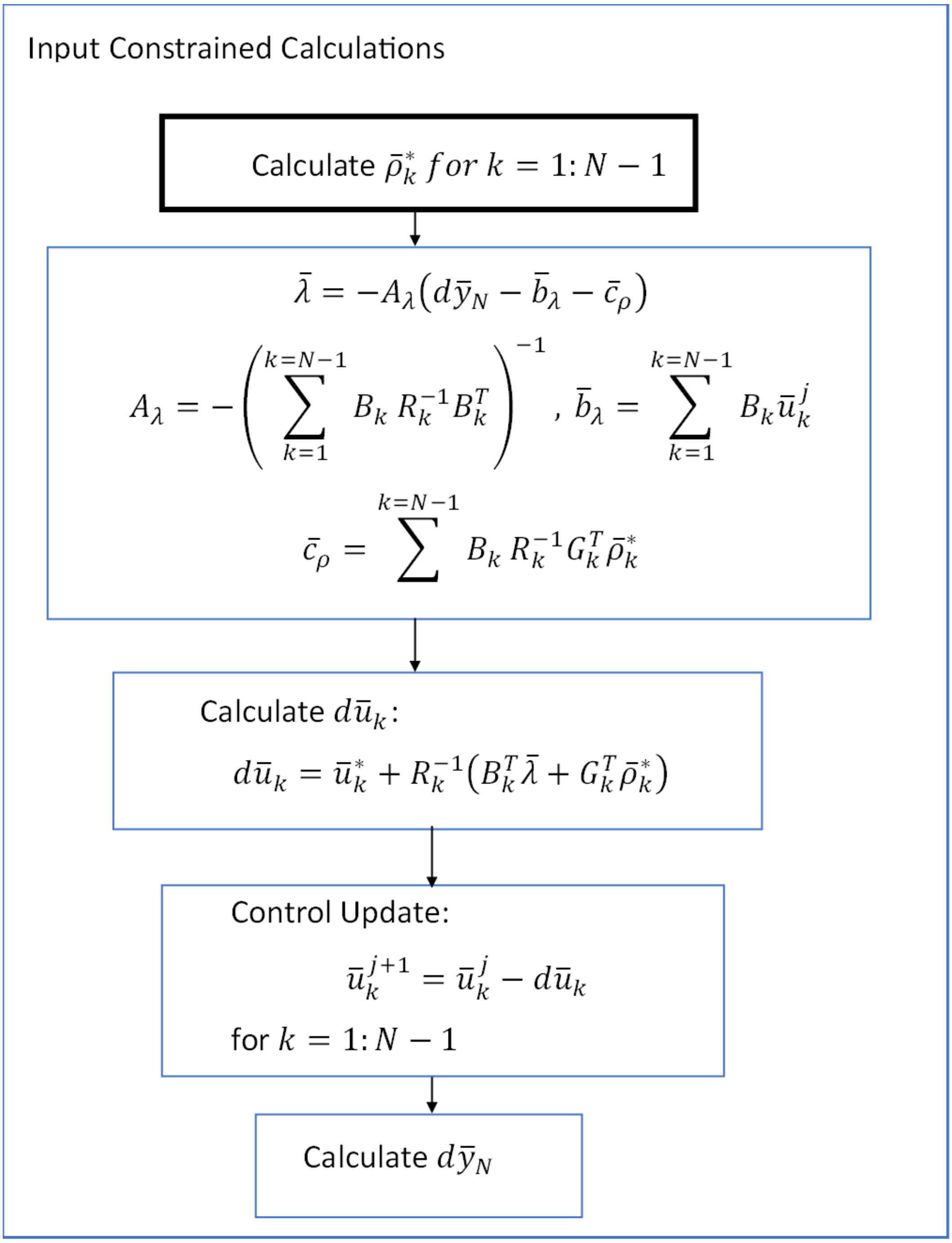

Input Constrained Calculations
Calculate $\bar{\rho}_k^* \; for \; k = 1:N-1$
$\bar{\lambda} = -A_\lambda\left(d\bar{y}_N - \bar{b}_\lambda - \bar{c}_\rho\right)$
$A_\lambda = -\left(\sum_{k=1}^{k=N-1} B_k R_k^{-1} B_k^T\right)^{-1}$, $\bar{b}_\lambda = \sum_{k=1}^{k=N-1} B_k \bar{u}_k^j$
$\bar{c}_\rho = \sum^{k=N-1} B_k R_k^{-1} G_k^T \bar{\rho}_k^*$
Calculate $d\bar{u}_k$:
$d\bar{u}_k = \bar{u}_k^* + R_k^{-1}\left(B_k^T \bar{\lambda} + G_k^T \bar{\rho}_k^*\right)$
Control Update:
$\bar{u}_k^{j+1} = \bar{u}_k^j - d\bar{u}_k$
for $k = 1:N-1$
Calculate $d\bar{y}_N$

Figure 15: Input Constrained Calculations

Figure 16 illustrates the implementation of Hildreth's algorithm in this study for calculating Lagrange multipliers associated with input constraints. It is crucial to terminate the algorithm when the obtained multipliers ρ_i converge to specific values. In each recursive step, the algorithm aims to minimize the cost function and satisfy the output equality constraint by updating the control inputs. To prevent the inability to proceed to the next recursive steps due to the failure of the termination condition of Hildreth's algorithm, an upper limit (r_{max}) is set on the number of iterations within the algorithm.

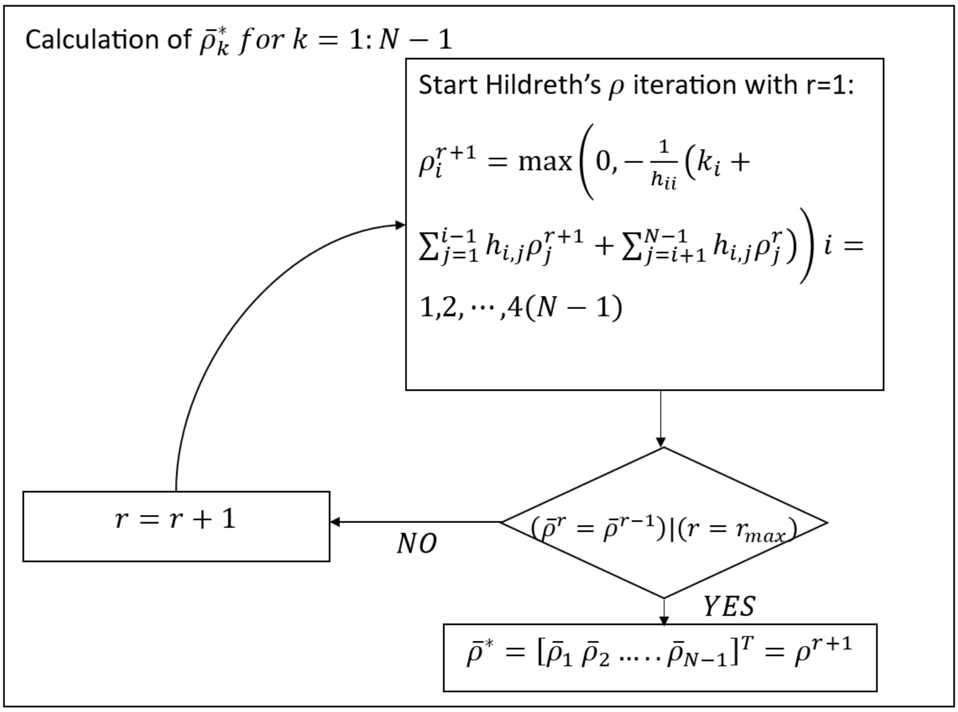

Calculation of $\bar{\rho}_k^* \; for \; k = 1: N - 1$
Start Hildreth's ρ iteration with r=1:
$\rho_i^{r+1} = \max\left(0, -\frac{1}{h_{ii}}\left(k_i + \sum_{j=1}^{i-1} h_{i,j}\rho_j^{r+1} + \sum_{j=i+1}^{N-1} h_{i,j}\rho_j^r\right)\right) i = 1,2,\cdots,4(N-1)$
$(\bar{\rho}^r = \bar{\rho}^{r-1})|(r = r_{max})$
NO
$r = r + 1$
YES
$\bar{\rho}^* = [\bar{\rho}_1 \, \bar{\rho}_2 \ldots\ldots \bar{\rho}_{N-1}]^T = \rho^{r+1}$

Figure 16: Lagrange Multipliers (Related to Input Constraint) Computation

4.4 Results

In this section, the results obtained in several scenario conditions will be examined. In all scenarios, the aerodynamic interceptor possesses the following characteristics:

- The aerodynamic interceptor’s initial position is at the origin of the inertial frame.

- The values of the drag coefficient components are determined based on Mach number.
- The reference area of the aerodynamic interceptor is the same in all scenarios.
- The autopilot dynamics of the aerodynamic interceptors are the same in all scenarios and under all flight conditions.
- The initial acceleration commands required to start the solution are obtained using the proportional navigation guidance law in all scenarios, without any acceleration command constraints.

4.4.1 Unconstrained Input Case

To clarify the approach, an example test case will be presented. The scenario information is shown in Table 1.

Table 1: Scenario Initial Conditions

y_d	$[0\text{ m}, 5167\text{ m}, -2983.2\text{m}]^{\mathrm{T}}$
β_0	30 deg
h_0	0 m
V_{M_0}	200 m/s
t_{go}	15 s

Figure 17 and Figure 18 present outcomes of the guidance algorithm for desired terminal condition $y_d = [0\text{ m}, 5167\text{ m}, -2983.2\text{ m}]$. It is worth noting that a total of 10 iterations were executed, and the output of the final iteration is depicted in the figures. In the figures, the blue and red lines correspond to the results obtained from the proportional navigation guidance (PNG) simulation and the Model Predictive Programming (MPP) approach, respectively. Additionally, the circle symbolizes the desired terminal position. Notably, MPP solution successfully achieves the interception, as demonstrated by the comparison between the desired terminal position and the MPP trajectory.

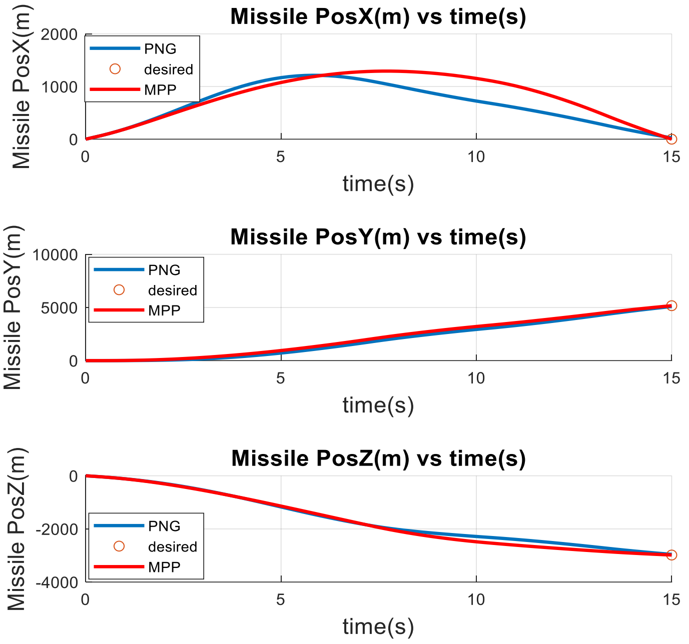


Figure 17: Positions of Interceptor for PNG and MPP Output

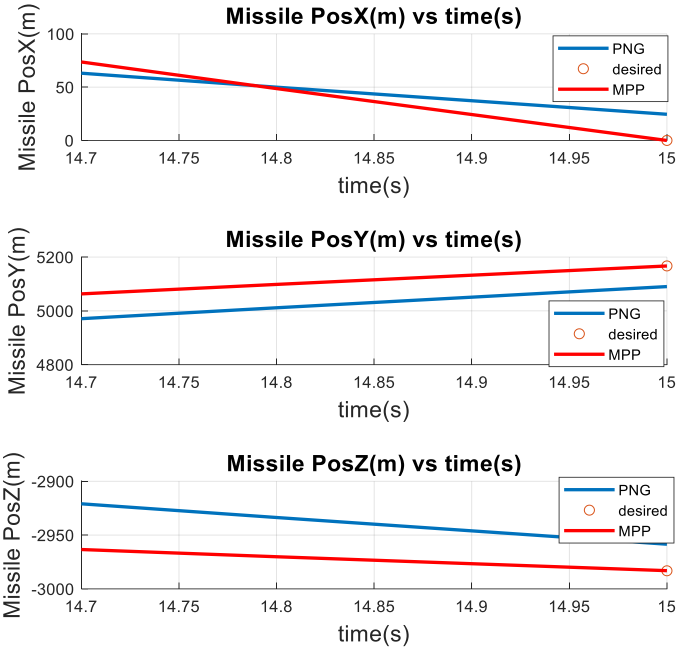


Figure 18: Positions of Interceptor for PNG and MPP Output (Zoomed)

Lateral acceleration commands are illustrated in Figure 19 and Figure 20 for desired terminal condition $\bar{y}_d = [0 \quad 5167 \quad 2983]^{\mathrm{T}}\, m$ after 10 iterations. In the figures, the red and blue lines correspond to the results derived from the proportional navigation guidance (PNG) and MPP methods, respectively. MPP approach demonstrates its capability to adapt the initial input vector to meet the constraint associated with the terminal position.

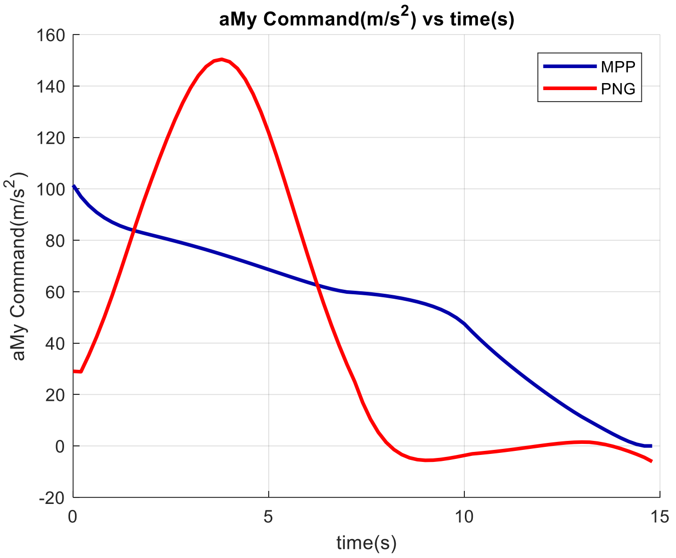


Figure 19: Y Axis Commanded Acceleration for Wind Frame - ${a_M}^W_{y,c}$ (MPP-PNG)

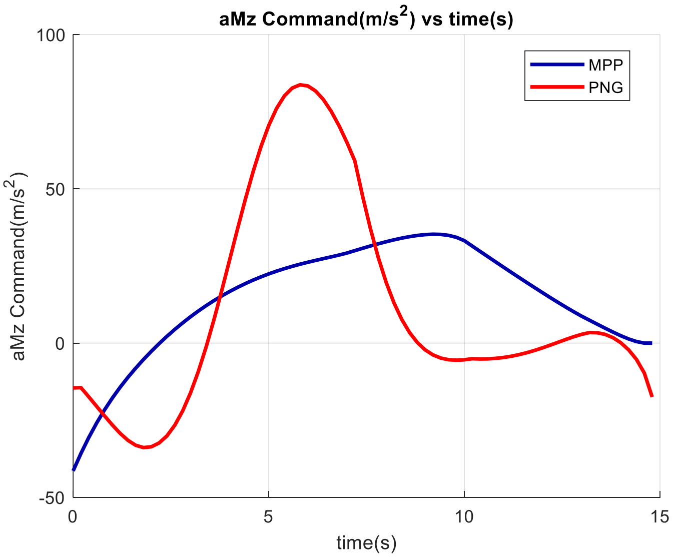


Figure 20: Z Axis Commanded Acceleration for Wind Frame - ${a_M}^W_{z,c}$ (MPP-PNG)

Figure 21 provides an analysis of the convergence process towards the desired terminal position. The first subplot depicted presents the magnitude of the terminal position error, showcasing the deviation from the target position. In the second subplot, the cost function magnitude is depicted. The cost function serves as a measure of the optimization objective. As depicted in the graph, it is evident that the magnitude of the output error converges to nearly zero after five iterations. As the iterations progress, the cost function value decreases, indicating a continuous improvement in achieving the optimization objective.

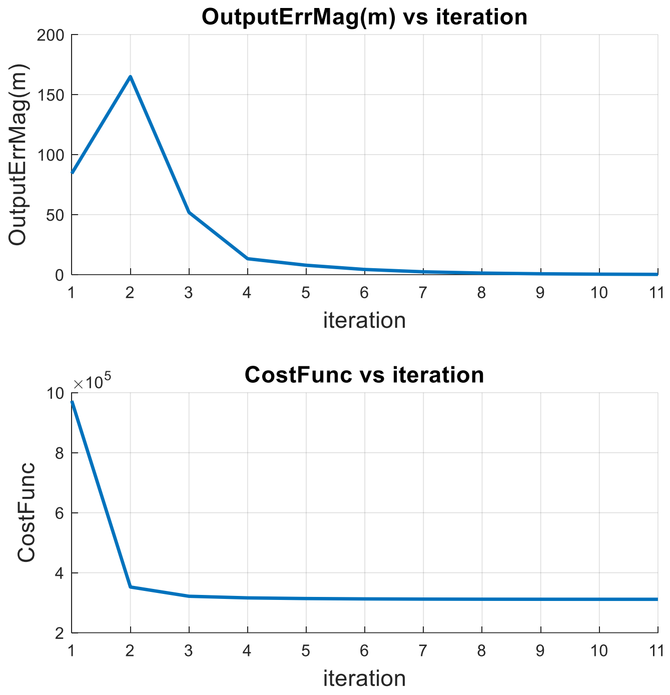


Figure 21: Convergence of Algorithm

Figure 22 and Figure 23 present commanded accelerations $\left(a_{M}{}_{y,c}^{W}, a_{M}{}_{z,c}^{W}\right)$ obtained from the model predictive programming (MPP) iterations. Each figure corresponds to a specific MPP iteration, showcasing the evolution of the acceleration commands throughout the optimization process. It can be observed lateral acceleration commands gradually converge to nearly identical values as the number of iterations increases. This convergence behavior indicates the refinement and optimization of the control inputs over successive iterations, resulting in consistent acceleration commands being applied to the system to achive the desired performance.

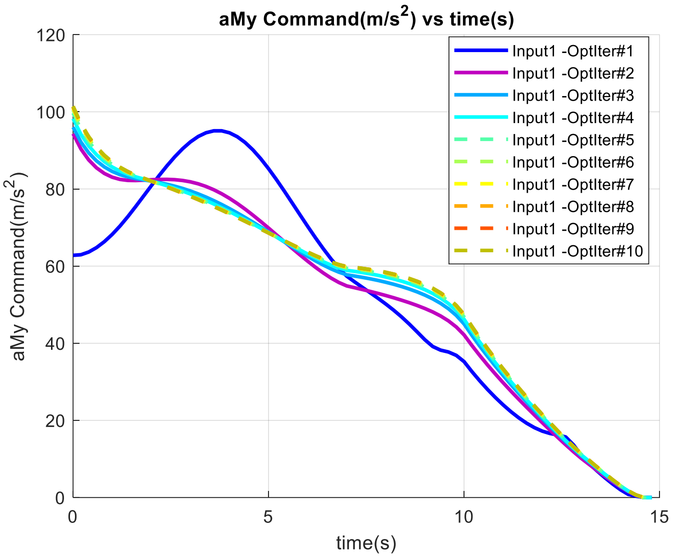


Figure 22: Y Axis Commanded Acceleration for Wind Frame $- {a_M}_{y,c}^W$

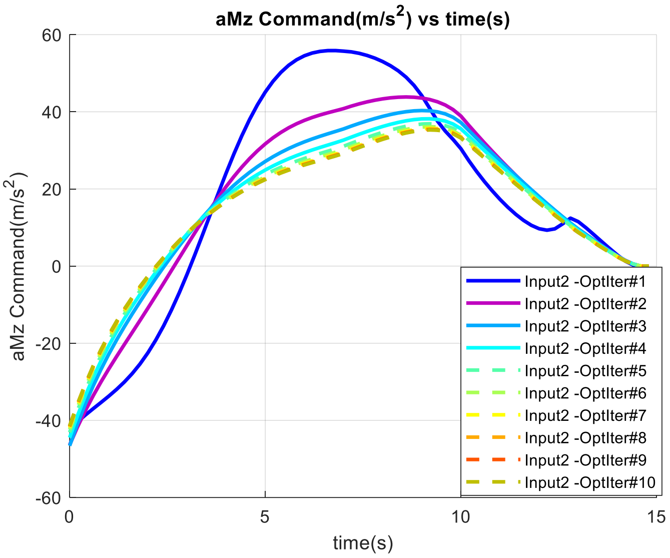


Figure 23: Z Axis Commanded Acceleration for Wind Frame Z Axis - ${a_M}_{z,c}^W$

Figure 24, Figure 25 and Figure 26 exhibit the evolution of the interceptor's position throughout the model predictive programming (MPP) iterations. These figures provide a visualization of the interceptor's trajectory during the optimization process. In each figure, the interceptor's position is depicted, with the final position indicated by a dotted line. The remaining lines represent the initial and intermediate paths that gradually converge to the final trajectory. The desired terminal position is denoted by a circle, representing the intended destination. It is shown that the interceptor's trajectory undergoes refinement and convergence over the course of the MPP iterations. After approximately 5 iterations, the interceptor's path significantly aligns with the desired terminal position. This convergence in the interceptor's position reflects the iterative refinement of control inputs and the optimization of the control strategy during the MPP process.

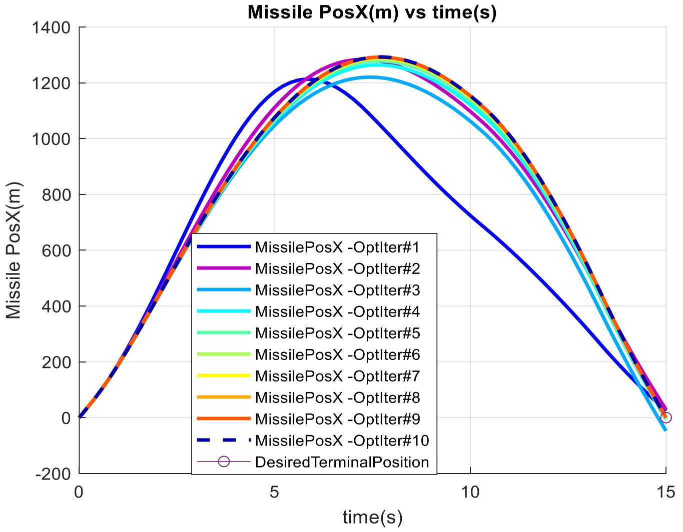


Figure 24: Interceptor Position and Desired Terminal Position in the x-axis

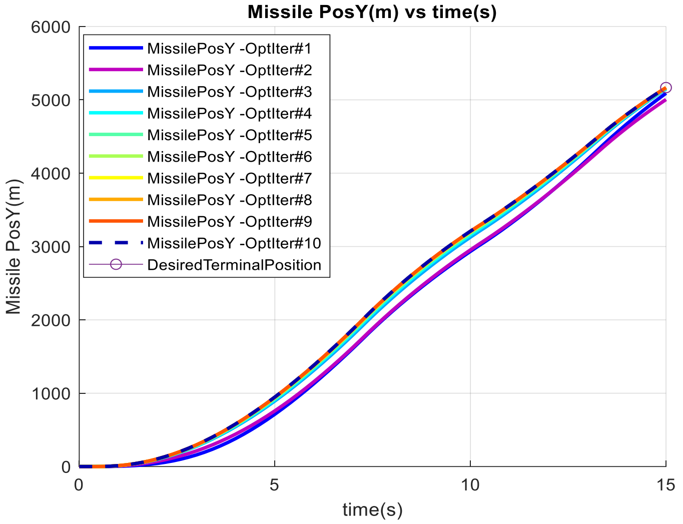


Figure 25: Interceptor Position and Desired Terminal Position in the y-axis

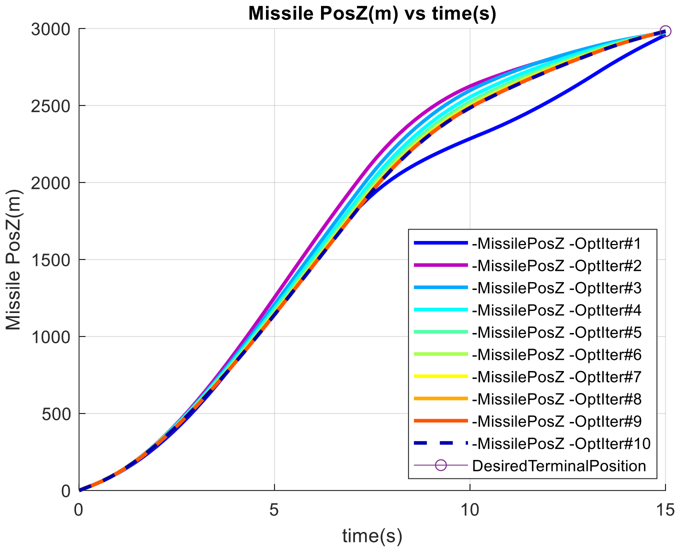


Figure 26: Interceptor Position and Desired Terminal Position in the z-axis

It is critical to emphasize that the accuracy and reliability of these calculations heavily rely on the proximity of the desired output value to the initial output, which is established by the initial control input vector. Specifically, the validity of the MPP approach is contingent upon the acceptability of linearization approximations employed in the optimization process. When the desired output value deviates significantly from the initial output, the assumptions made in linearizing the system may become inadequate, thereby diminishing the validity of the MPP calculations. In such cases, it becomes challenging to descend to a local minimum of the cost function or achieve convergence of iterations towards a feasible solution.

4.4.2 Input Constraint Case

This part presents outcomes obtained when considering acceleration limits throughout the entire flight of the interceptor. The imposed constraints on acceleration values are contingent upon the specific flight conditions of the interceptor, such as Mach number and altitude. In this particular analysis, acceleration limits are incorporated to showcase the activation of constraints at the beginning and middle stages of the flight.

Figure 27 and Figure 28 depict the commanded accelerations derived from the unconstrained optimization problem illustrated in Figure 19 and Figure 20. The red and blue lines represent the simulation results obtained using the proportional navigation guidance (PNG) and the model predictive programming (MPP) approaches, respectively. The black lines represent the maximum and minimum limits of the acceleration values imposed by the constraints. It can be noticed that the MPP solution, when formulated without considering the input constraints, leads to commanded accelerations that surpass the prescribed acceleration limits for this specific test case.

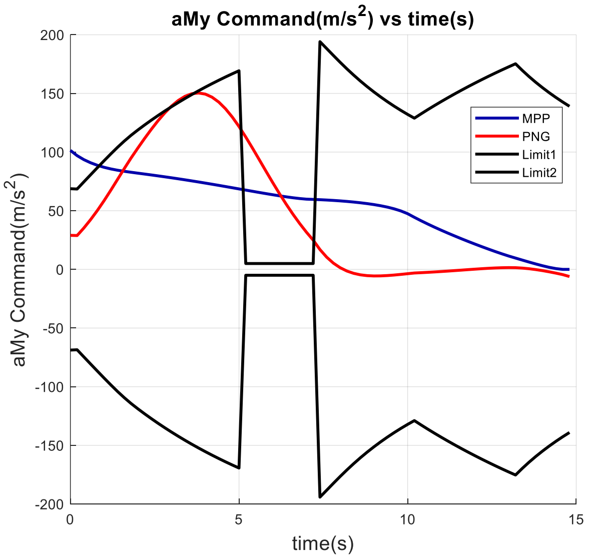


Figure 27: Y Axis Commanded Acceleration for Wind Frame (MPP-PNG)

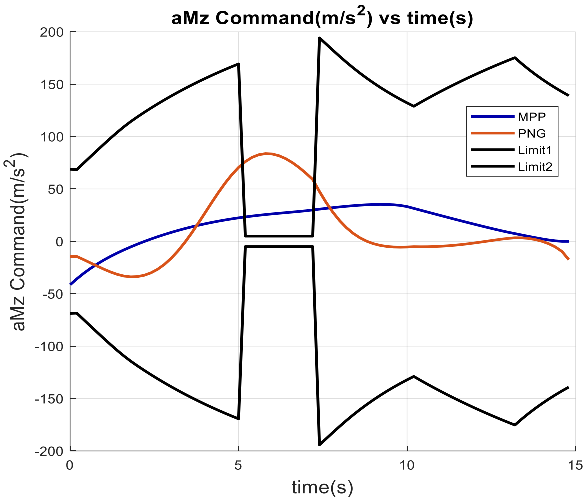


Figure 28 Z Axis Commanded Acceleration for Wind Frame (MPP-PNG)

Figure 29 illustrates results obtained from the input constrained MPP result for desired terminal condition $y_d = [0,5167,-2983.2]\, m$ after 10 iterations. The red and blue lines represent the MPP solutions for the cases with and without input constraints, respectively. The circle denotes the desired terminal position. It can be noticed that the MPP solution considering input constraints generates control commands that facilitate a successful interception of the target.

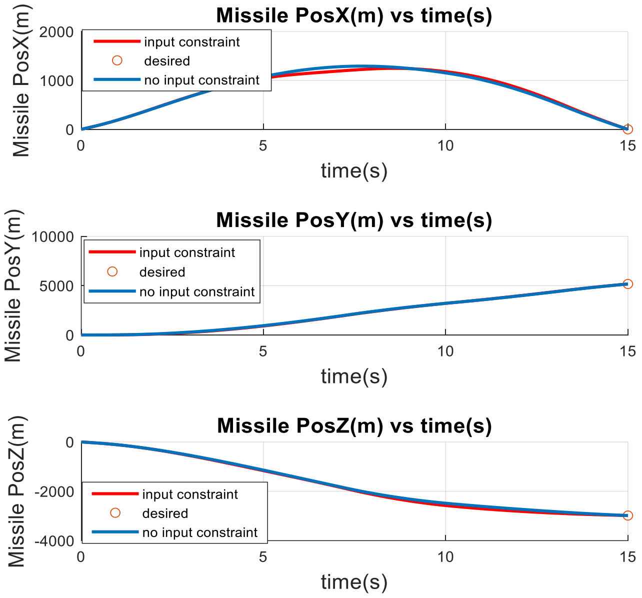


Figure 29: Positions of Interceptor with and without Input Constraints

A comparison of the lateral commanded acceleration obtained from the input constrained optimization problem, unconstrained optimization problem, and proportional navigation guidance (PNG) is presented in Figure 30 and Figure 31. The red, blue, and green lines represent the results obtained from PNG, unconstrained input case solution, and solution of the input-constrained problem respectively. The black lines depict the maximum and minimum limits of the acceleration constraints. Based on the information provided, it can be mentioned that the approach with input constraints alters the initial input vector to meet both the terminal position constraint (Figure 29) and the input constraint. By considering the input limits, the solution ensures that the generated control commands fall within the permissible range, thereby avoiding impractical or unsafe maneuvers.

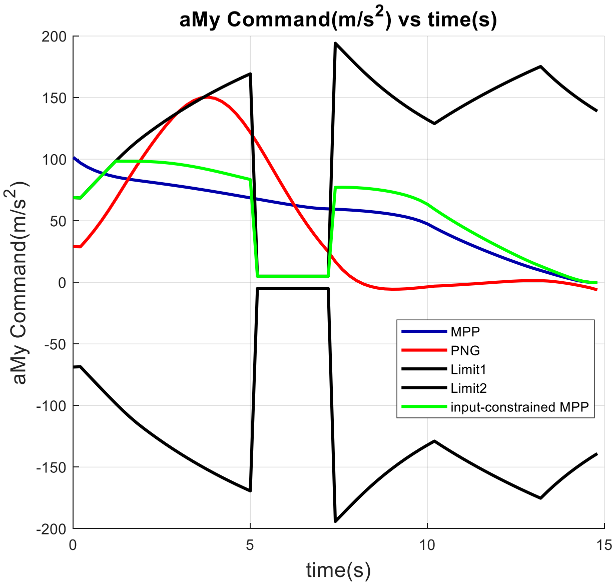


Figure 30: Y Axis Commanded Acceleration for Wind Frame (MPP-PNG)

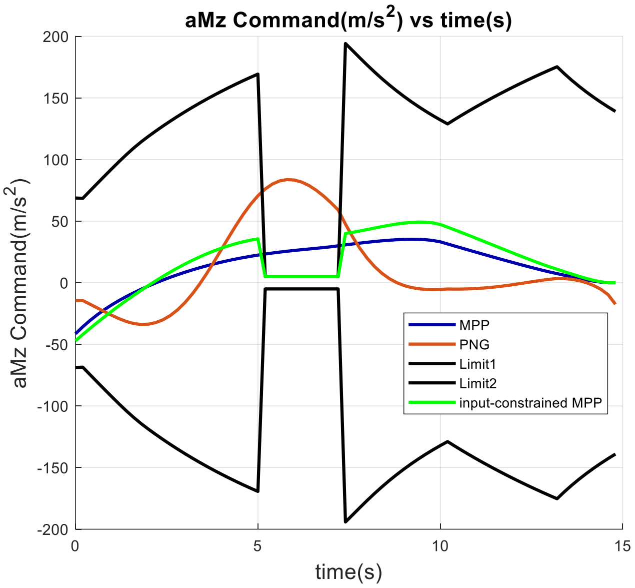


Figure 31: Z Axis Commanded Acceleration for Wind Frame (MPP-PNG)

Figure 32 presents the convergence analysis of iterations towards the desired terminal position for the scenario with input constraints. The first subplot in Figure 32 depicts the error in the terminal position, while the second subplot displays the value of the cost function. Examining the first subplot, it can be observed that the terminal position error decreases progressively as the iterations proceed. The error tends to approach zero after approximately 5 iterations, indicating that the approach effectively guides the interceptor towards the desired terminal position. In the second subplot, the magnitude of the cost function is presented. The cost function represents a measure of the overall optimization objective, which is minimizing energy effort. The plot shows a significant decrease in the cost function after the first iteration. By iteratively refining the control commands, the algorithm achieves the dual objectives of minimizing the terminal position error and optimizing the cost function for this test case.

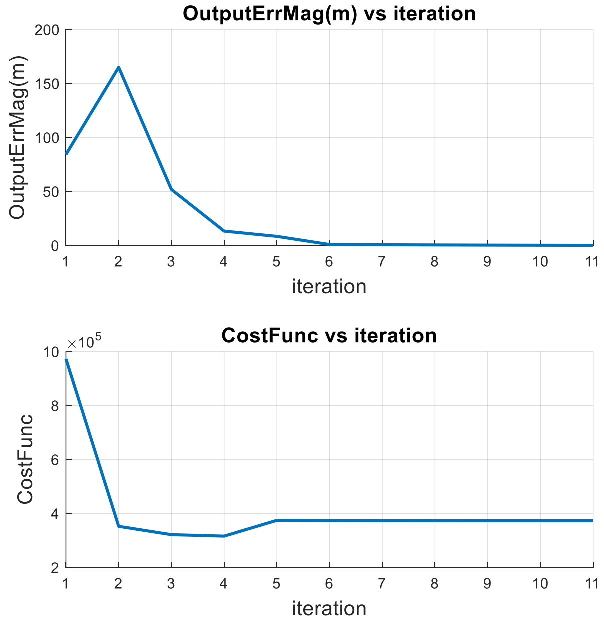


Figure 32: Convergence of Algorithm

Figure 33 and Figure 34 show commanded accelerations during each optimization iteration. It is important to note that the input constrained calculation commences when the terminal position deviation of the unconstrained solution falls below a specified tolerance value, and the resulting input vector surpasses the acceleration limits. Conversely, if the position deviation in the current iteration surpasses the tolerance threshold, the iterations continue until the unconstrained solution approximates the target position within the maximum allowable iterations. The rationale behind this approach is grounded in the understanding that if the interceptor fails to reach the desired position with unconstrained acceleration commands, it is unlikely to achieve the same objective under constrained conditions. Thus, if the interceptor gets close to the desired position during the current iteration, the Lagrange multipliers for input constraints are computed to determine the input constrained solution.

In Figure 32, the terminal position deviation exceeds the specified tolerance threshold of 50 meter during the first three iterations of the optimization process. Consequently, the acceleration commands calculated during these iterations are obtained using the unconstrained approach, leading to acceleration values that surpass the imposed limits, as depicted in the accompanying figures. However, starting from the fourth iteration, when the terminal position deviation falls below the tolerance threshold, the acceleration commands are recalculated considering the input constraints. The algorithm then focuses on finding a feasible solution that satisfies both the terminal position constraint and the input constraints. The rapid convergence of the acceleration commands towards a viable solution, while adhering to the input constraints, can be observed in Figure 33 and Figure 34.

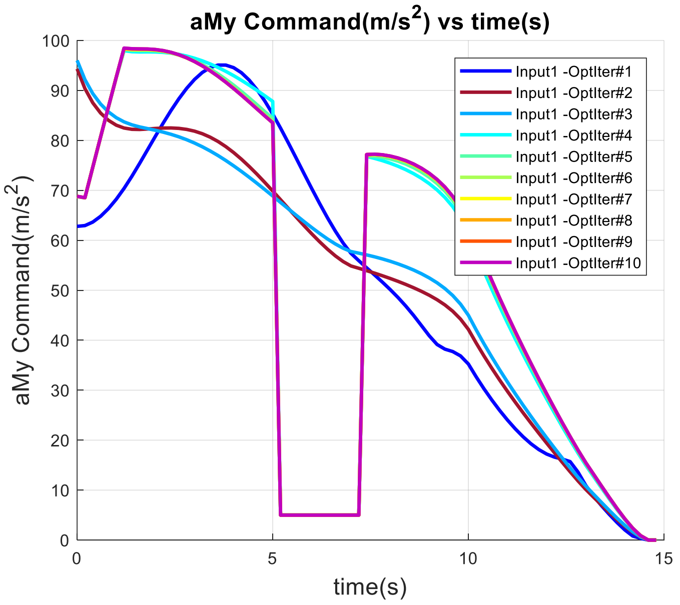


Figure 33: Y Axis Commanded Acceleration for Wind Frame

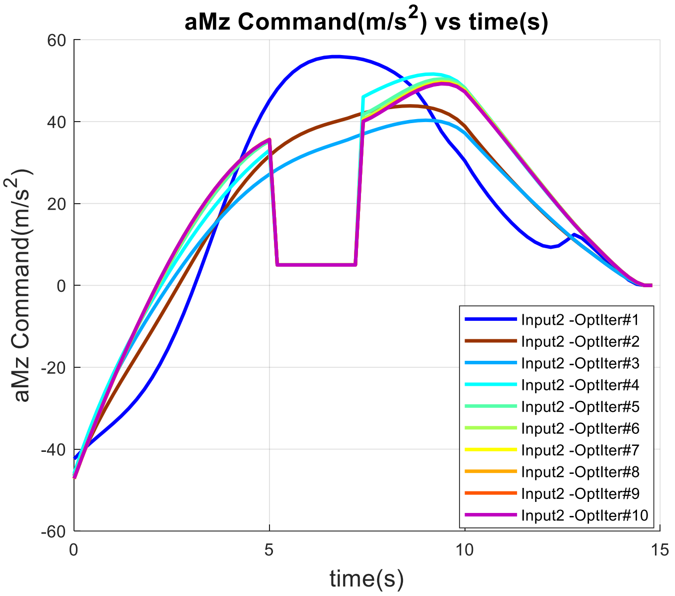


Figure 34: Z Axis Commanded Acceleration for Wind Frame

The MPP iterations in this study are conducted until the terminal position error falls below a specified threshold. Once this criterion is met, the iterations are stopped, and the $\bar{\rho}$ iterations associated with the final MPP iteration are performed until convergence of $\bar{\rho}$ or reaching maximum $\bar{\rho}$ iterations. However, in this particular test case, the MPP iterations are not terminated for the purpose of demonstration, and the iterations for calculating the Lagrange multipliers $\bar{\rho}$ iterations are conducted solely when the terminal position error falls below a pre-defined threshold. The Lagrange multipliers associated with input constraints for each MPP iteration are illustrated in Figure 35 , where distinct colors indicate varying $\bar{\rho}$ values for different MPP iterations. For every time step, acceleration limits are specified for each input component $\left(a_{M_{y,c}}^{W}, a_{M_{z,c}}^{W}\right)$, resulting in the establishment of four constraints for each time step. In the given scenario, considering a sampling time of 0.2 s, the 15-second scenario is partitioned into 75 time steps, resulting in a cumulative count of 300 Lagrange multipliers corresponding to input constraints. The graph employs the x-axis to represent constraint counters (up to 300) and the y-axis to signify Hilderth's $\bar{\rho}$ iterations. The z-axis shows the components of the $\bar{\rho}$ vector. The process of determining Lagrange multipliers associated with input constraints is carried out iteratively using Hildreth's procedure. This iterative process continues until the $\bar{\rho}$ vector converges for each MPP iteration. Based on Figure 35, it can be observed that the $\bar{\rho}$ vector converges within four iterations for the fourth and fifth MPP iterations (turquoise and green), two iterations for the sixth, seventh, and eighth MPP iterations (light green, yellow, orange), and one iteration for the ninth and tenth MPP iterations (red, blue). It can be inferred that, the convergence of the Lagrange multipliers related to the input-constrained algorithm becomes easier when the output error is lower in the unconstrained case. In other words, a smaller discrepancy between the desired output and the unconstrained output facilitates the convergence of the Lagrange multipliers.

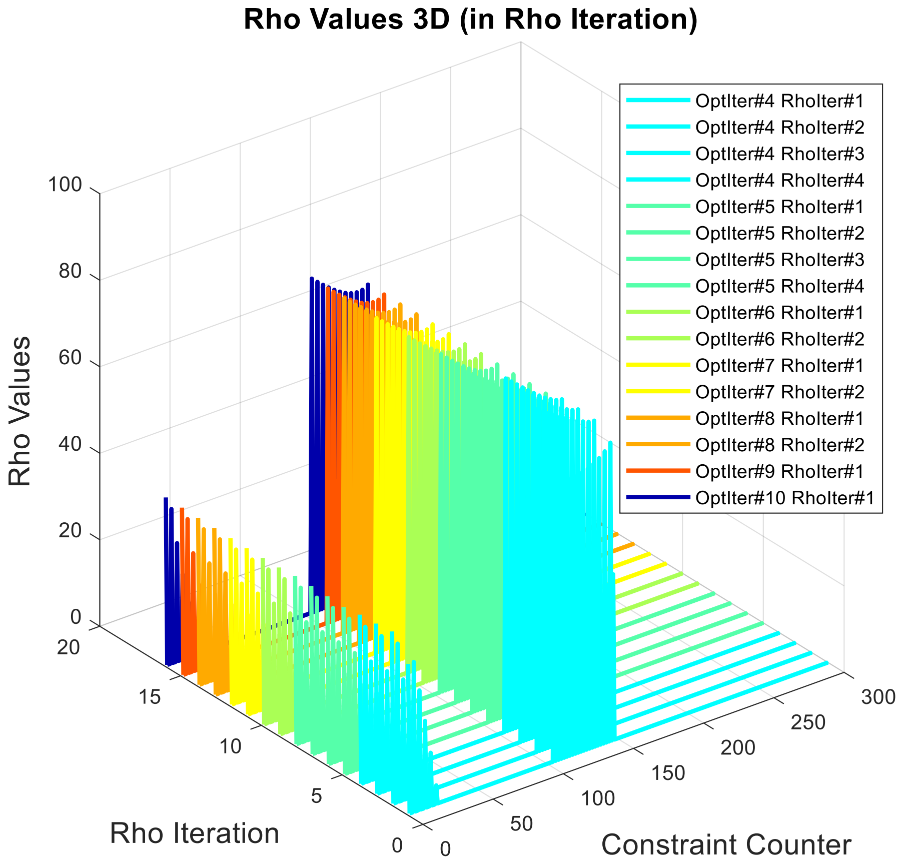


Figure 35: Rho Values for each MPP iteration

4.5 Discussions

There are several approaches in the literature for solving the optimization problem in guidance. A significant majority of these approaches involve significant simplifications or assumptions in the relative kinematic equations or the model of the guided object in order to enable analytical solutions to the problem. As these simplifications or assumptions deviate from describing the real system, it becomes increasingly challenging to make judgments about the optimality of the obtained solution for the relevant system. The general simplifications or assumptions made can be summarized as follows:

- The magnitude of the interceptor's velocity is assumed to be constant.
- The maneuvers performed by the intreceptor are assumed to have no effect on the magnitude of velocity or the rate of increase of velocity.
- The maneuver capability of the interceptor is assumed to be unlimited or to have the same maneuver constraint under all flight conditions.
- The acceleration commands are assumed to be executed by an ideal autopilot dynamics.
- The flight mechanics of the interceptors are completely neglected.
- Atmospheric conditions are not modeled.
- Approximate relative kinematic equations valid for small angle states are used.

Existing approaches for solving model predictive static programming for nonlinear systems have been used for the case of input constraints with inequality bounds. In order to determine which inequality constraints are active or inactive, all Lagrange multipliers need to be solved simultaneously. There are two main challenges for directly performing such a computation:

- The need for high-dimensional matrix operations.
- The requirement for Lagrange multipliers associated with inequality constraints to be either zero or positive.

Due to these reasons, determining which inequality constraints are active and the corresponding Lagrange multipliers is solved recursively following Hildreth's directive [10]. It should be noted that if the inequality constraints on the input make it impossible to satisfy the equality constraint on the output, the convergence of Hildreth's directive to any solution is not possible, regardless of the cost function. Additionally, if both the inequality constraints on the input and the equality constraint on the output result in a very limited solution space, the convergence of Hildreth's procedure to a solution may take a long time. Therefore, an upper limit is imposed on the number of recursive steps in the implementation of Hildreth's procedure, in addition to the convergence criterion specified in Section 4.2. This upper limit can be determined based on the following considerations:

- Acceptable computational time length.
- Acceptable magnitude of error for the optimization solution.

In this study, a guidance solution for the equality constraint on the target position and inequality constraints on commanded acceleration is demonstrated. By modifying the definitions of the outputs, cost function, and inequality conditions or introducing new definitions, solutions for different guidance problems (such as impact angle, linear velocity at arrival, acceleration command at arrival or realized acceleration, reduction of total drag force, avoidance of restricted zones) can also be obtained. The online applicability of this approach depends on the following factors:

- Complexity of the system model (e.g., as the system order increases, the number of computations will increase).
- Time step of the solution and flight duration (e.g., as the time step becomes smaller and the flight duration becomes longer, the number of computations will increase).
- Narrowness of the solution space that satisfies both the inequality constraints on the input and the equality constraint on the output (e.g., if it is almost impossible to reach the solution that satisfies the output constraint due to the

given input constraints, the recursion of Hildreth's directive may take a very long time or may not converge at all).

In addition, it should be noted that this approach does not guarantee finding the globally optimal solution and it is possible to converge to local optima. Therefore, this approach should be considered as a solution method within the class of "best proximity" rather than a globally optimal solution method.

CHAPTER 5

REACHABLE SET ANALYSIS TOOL DEVELOPMENT

The key purpose of performing reachable set calculations for an interceptor is to evaluate its kinematic capabilities throughout a finite flight duration under diverse initial flight conditions. This study examines the kinematic capabilities by defining the lower and upper bounds of reachability for flight ranges which are influenced by the flight duration and initial flight conditions. To accomplish this, a reachable set computation procedure based on input-constrained model predictive programming approach is proposed. The procedure aims to minimize a cost function, enabling the generation of optimal control commands that result in a reduced total control action throughout the entire flight duration. Additionally, the procedure ensures adherence to both inequality constraints associated with control command magnitudes and equality constraints concerning the final position.

To calculate the reachable set, it is crucial to begin by defining search directions and minimum and maximum range reachability boundaries. Specifically, the reachable set is characterized by the boundaries of maximum and minimum reachability, which are computed for different search angles and flight conditions.

The minimum and maximum range reachability boundaries represent the range limits that an interceptor can achieve from its initial position within a specified flight duration under specific flight conditions and for different search directions.

By effectively storing and consolidating the reachable points obtained through the computation process, the boundaries of the reachable set are systematically constructed, accommodating different flight conditions. Subsequently, a logical framework is developed to facilitate the comprehensive exploration of the reachable set boundaries, allowing for a thorough analysis of the interceptor's kinematic capabilities within the given flight duration.

Finally, the reachable set boundaries are determined by applying the proposed reachable set computation procedure, which involves the optimization of the cost function and the satisfaction of both inequality and equality constraints.

5.1 Reachability Boundary and Search Direction Definitions

In this section, definitions for minimum and maximum reachability boundaries and search directions are provided. The minimum range reachability boundary refers to the shortest distances an interceptor can reach from its initial position within a given flight duration, considering different search direction.

Conversely, the maximum range reachability boundary represents the farthest distance that the interceptor can reach from its initial position within the specified flight duration, considering different search direction.

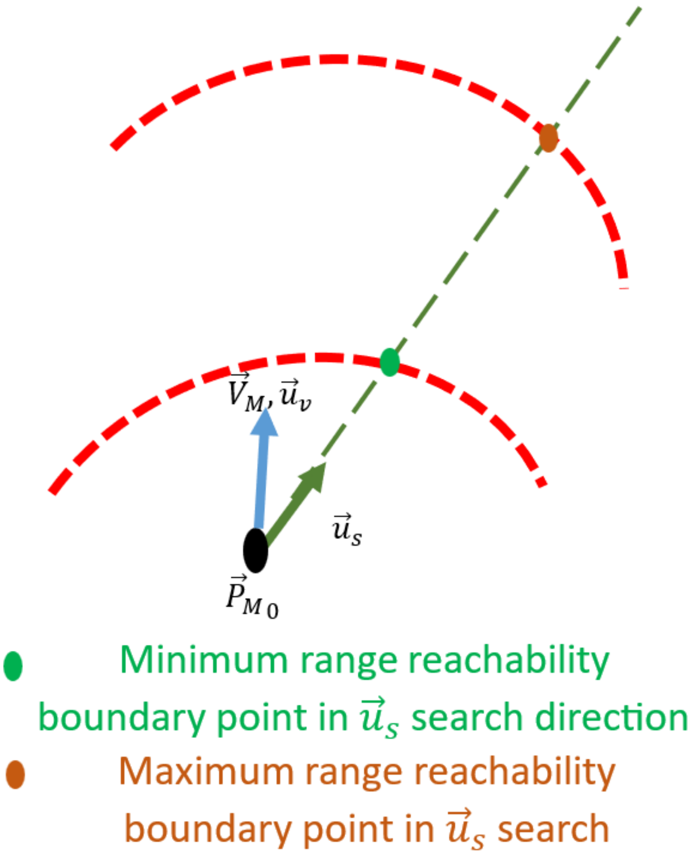
To determine the minimum and maximum range reachability boundaries, a range of desired terminal positions is defined relative to the current position and search directions by the following relation:

$$\bar{y}_N^d = \bar{f}(R) = \bar{P}_{M,0} + R\vec{u}_s, \bar{P}_{M,0} = \begin{bmatrix} 0 \\ 0 \\ -h_0 \end{bmatrix} \qquad \text{Eq. 146}$$

These desired terminal positions are obtained by considering different ranges (R) from the current position and applying search angles with respect to the unit vector of the interceptor's velocity ($\vec{u}_v$).

By exploring the reachability of the interceptor to these desired terminal positions, the minimum and maximum range reachability boundaries are identified.

Figure 36 provides an illustrative example depicting the maximum and minimum boundaries in different search directions. Additionally, it showcases the unit vector of the interceptor's velocity ($\vec{u}_v$) and the search direction unit vector ($\vec{u}_s$) which is obtained by 2 succecessive rotations.



$\vec{V}_M, \vec{u}_v$
$\vec{u}_s$
$\vec{P}_{M0}$
● Minimum range reachability boundary point in $\vec{u}_s$ search direction
● Maximum range reachability boundary point in $\vec{u}_s$ search

Figure 36: Maximum and Minimum Boundaries in Search Directions

The search directions for determining the boundaries of the reachable set are expressed relative to the frame $\mathcal{F}_W$. The terminal position is determined along the first unit vector of the frame $\mathcal{F}_S$ $(\vec{u}_1^S)$ which is acquired through successive rotations about the frame $\mathcal{F}_I$ as described by the Rodrigues formulation.

The rotational sequence for obtaining $\vec{u}_1^S$ is as follows:

$$\mathcal{F}_W \rightarrow \mathcal{R}(\vec{u}_3^I, \alpha_s) \rightarrow \mathcal{F}_H \rightarrow \mathcal{R}(\vec{u}_2^H, \beta_s) \rightarrow \mathcal{F}_S \qquad \text{Eq. 147}$$

Here, $\mathcal{R}(\vec{u}_2^H, \beta_s)$ represents a rotation matrix that defines the transformation from frame $\mathcal{F}_H$ to frame $\mathcal{F}_S$. It is given by the matrix:

$$\mathcal{R}(\vec{u}_2^H, \beta_s) = \begin{bmatrix} cos(\beta_s) & 0 & sin(\beta_s) \\ 0 & 1 & 0 \\ -sin(\beta_s) & 0 & cos(\beta_s) \end{bmatrix} \qquad \text{Eq. 148}$$

Similarly, the rotation matrix $\mathcal{R}(\vec{u}_3^I, \alpha_s)$ represents the rotation matrix obtained by rotating frame $\mathcal{F}_W$ along $\vec{u}_3^I$ of frame $\mathcal{F}_I$. It is computed using the Rodrigues formulation as:

$$\begin{aligned}\mathcal{R}(\vec{u}_3^I, \alpha_s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + sin(\alpha_s) \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ + (1 - cos(\beta_s)) \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\end{aligned} \qquad \text{Eq. 149}$$

By utilizing these rotational transformations, the target terminal position in the inertial frame can be represented in relation to the range (R) , $\mathcal{R}(\vec{u}_3^I, \alpha_s)$, $\mathcal{R}(\vec{u}_2^H, \beta_s)$, α_s and β_s. This representation allows for the determination of the terminal position in relation to the flight parameters and search directions.

Figure 37 shows successive rotations between $\mathcal{F}_W$ and $\mathcal{F}_S$. ,highlighting the transformation process involved.

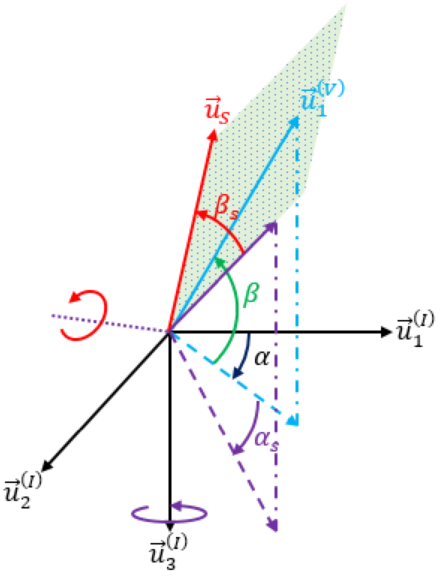


Figure 37: Search Direction

5.2 Reachable Set Boundary Search Procedure

In this section, the search logic which is employed to determine the reachable set boundaries for an interceptor is explained. The search logic follows a systematic procedure, taking into account various flight conditions and constraints, to guide the interceptor towards the desired position while considering the terminal position error and input constraints.

The procedure is summarized in the flowchart shown in Figure 38 and the detailed steps involved in the reachable set boundary search logic are as follows:

1) Define Flight Conditions for Reachable Set Computation: Before initiating the reachable set computation, it is crucial to define the search direction and specific flight conditions under which the analysis will be conducted. These flight conditions encompass a range of parameters, including altitude, Mach number, and other relevant factors that significantly impact the flight characteristics and performance of the interceptor.
2) Define a Terminal Desired Position and Generate Initial Input Vector: A terminal desired position is established as the target location that the interceptor aims to reach within the specified flight duration.The generation of the terminal desired positions is explained in 5.1. Initially, the interceptor is guided towards this desired position by employing the Proportional Navigation Guidance (PNG) approach. This approach provides an initial trajectory for the interceptor, utilizing guidance laws that minimize the line-of-sight (LOS) rate between the interceptor and the target.
3) Compute the Solution using Model Predictive Static Programming: Once the initial input vector is generated through the PNG approach, it is incorporated into the Model Predictive Static Programming (MPSP) approach. The MPSP algorithm optimizes a cost function to compute a refined solution that minimizes the control effort required throughout the complete flight duration.
4) Check Terminal Position Error and Input Constraints: After computing the solution using the MPSP approach, it is essential to evaluate the terminal

position error to determine whether it falls below a predefined threshold. Additionally, the activation of input constraints, such as acceleration limits, is examined. If the terminal position error is within the desired threshold and the input constraints are not violated, it signifies that the computed solution meets the predefined criteria.

5) Execution of Reachability Boundary Calculations: If the conditions specified in step 4 are not satisfied, it is not necessary to execute the calculations for the minimum and maximum reachability boundaries for the current flight condition. In such cases, the analysis proceeds to another flight condition, adjusting the parameters as necessary. However, if the conditions specified in step 4 are satisfied, indicating that the computed solution meets the desired criteria, the calculations for the minimum and maximum reachability boundaries are executed.

By following this search logic, the reachable set boundaries for the interceptor can be determined under different flight conditions. The procedure ensures that the boundaries are calculated only when the terminal position error is within the desired range and the input constraints are not violated. This approach provides a systematic method for exploring the kinematic capabilities of the interceptor and obtaining valuable insights into its performance and limitations.

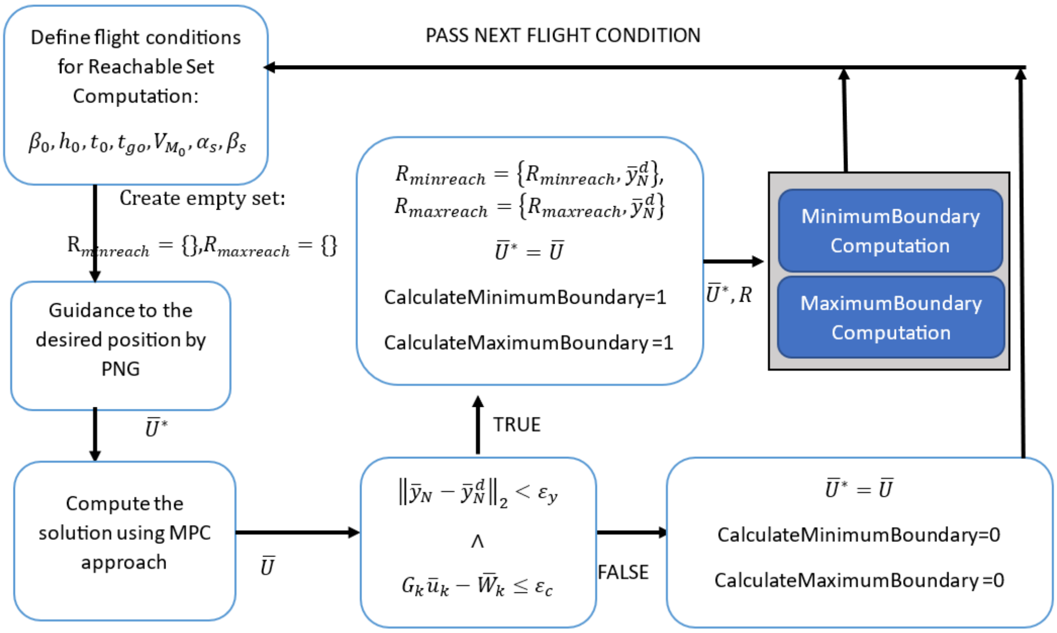

PASS NEXT FLIGHT CONDITION
Define flight conditions for Reachable Set Computation:
$\beta_0, h_0, t_0, t_{go}, V_{M_0}, \alpha_s, \beta_s$
Create empty set:
$R_{minreach} = \{\}, R_{maxreach} = \{\}$
Guidance to the desired position by PNG
$\bar{U}^*$
Compute the solution using MPC approach
$\bar{U}$
$R_{minreach} = \{R_{minreach}, \bar{y}_N^d\}$,
$R_{maxreach} = \{R_{maxreach}, \bar{y}_N^d\}$
$\bar{U}^* = \bar{U}$
CalculateMinimumBoundary=1
CalculateMaximumBoundary=1
$\bar{U}^*, R$
MinimumBoundary Computation
MaximumBoundary Computation
TRUE
$\|\bar{y}_N - \bar{y}_N^d\|_2 < \varepsilon_y$
∧
$G_k \bar{u}_k - \bar{W}_k \le \varepsilon_c$
FALSE
$\bar{U}^* = \bar{U}$
CalculateMinimumBoundary=0
CalculateMaximumBoundary=0

Figure 38: Reachable Set Computation Procedure

5.3 Reachability Boundary Determination for Maximum and Minimum Flight Range

In this section, the procedure employed to determine the reachability boundaries for the maximum and minimum flight range of an interceptor is explained. The procedure involves propagating an initial guess of the interceptor's terminal position, assessing the reachability for the updated desired position, and iterating the process until the range limits are reached. Figure 39 depicts a flowchart illustrating process of reachability boundary computation. The flowchart provides a visual representation of the procedure, highlighting the iterative nature of the process and the various stages involved in determining the reachability boundaries.

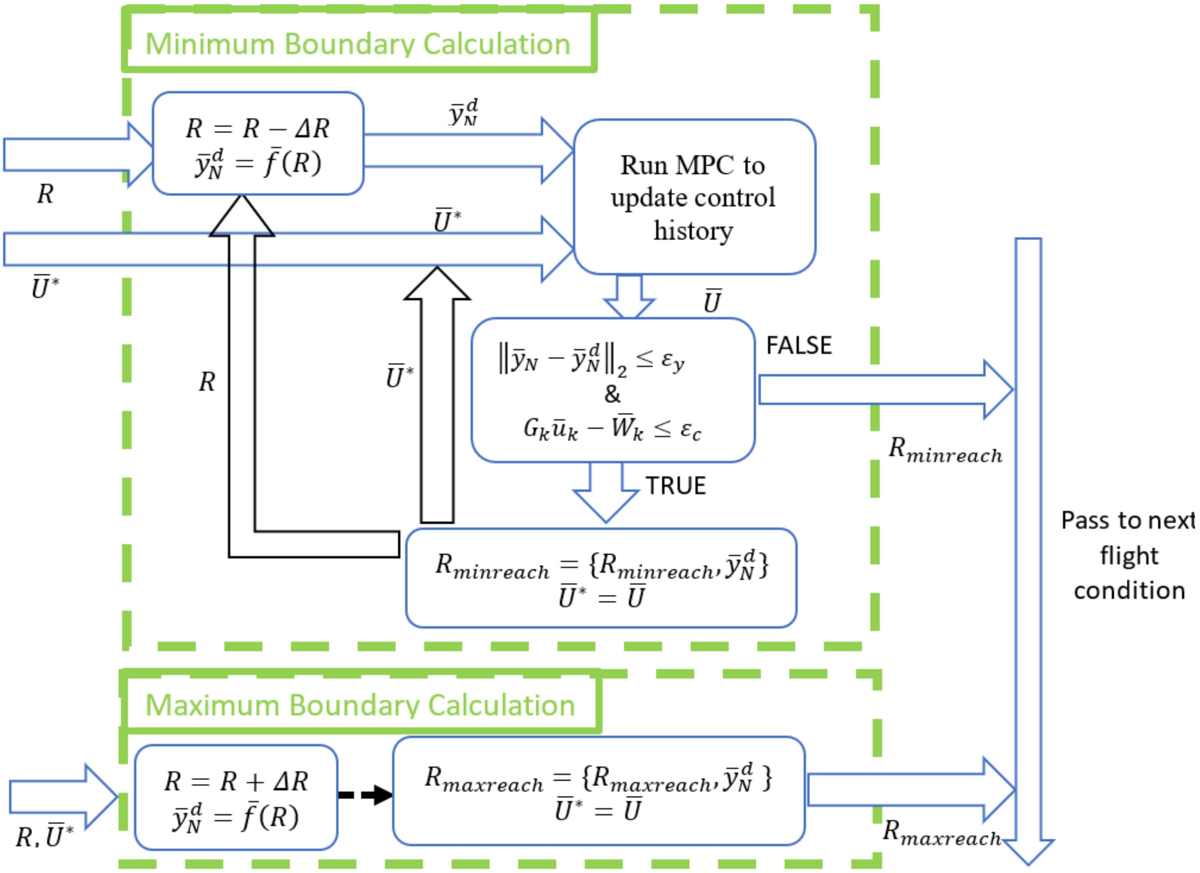

Minimum Boundary Calculation
$R = R - \Delta R$
$\bar{y}_N^d = \bar{f}(R)$
R
$\bar{y}_N^d$
Run MPC to update control history
$\bar{U}^*$
$\bar{U}^*$
$\bar{U}$
R
$\bar{U}^*$
$\|\bar{y}_N - \bar{y}_N^d\|_2 \leq \varepsilon_y$
&
$G_k\bar{u}_k - \bar{W}_k \leq \varepsilon_c$
FALSE
TRUE
$R_{minreach} = \{R_{minreach}, \bar{y}_N^d\}$
$\bar{U}^* = \bar{U}$
$R_{minreach}$
Pass to next flight condition
Maximum Boundary Calculation
$R = R + \Delta R$
$\bar{y}_N^d = \bar{f}(R)$
$R, \bar{U}^*$
$R_{maxreach} = \{R_{maxreach}, \bar{y}_N^d\}$
$\bar{U}^* = \bar{U}$
$R_{maxreach}$

Figure 39: Flowchart of Reachability Boundary Computation

5.3.1 Reachability Boundary for Maximum Flight Range

To determine the reachability boundaries for the maximum flight range, the procedure commences by propagating an initial estimate of the interceptor's terminal position in the forward search direction. This forward search direction corresponds to an increase in flight range. The initial guess is obtained by taking a step in a chosen direction, considering factors such as target location and flight conditions.

Subsequently, the Model Predictive Static Programming (MPSP) approach is employed to assess the feasibility of reaching the updated desired terminal position.

By evaluating the terminal position error of the interceptor, it is determined whether the updated desired position is achievable within the specified flight duration.

In the event that position error at the terminal point of the interceptor falls below a predetermined tolerance threshold and the resulting input vector does not surpass the specified acceleration limit, the updated terminal position is considered a candidate for the reachability boundary for the maximum flight range. This implies that the interceptor can successfully reach the updated position while satisfying the imposed constraints. For each MPP computation, the input vector history that has been recently updated is utilized, ensuring that the calculations are based on the most up-to-date information.

The above process is reiterated until the interceptor is unable to reach the updated terminal position. This iterative approach allows for a thorough exploration of the reachability boundaries for the maximum flight range, taking into account different flight conditions, target locations, and system constraints.

5.3.2 Reachability Boundary for Minimum Flight Range

Similarly, the procedure is also applied to determine the reachability boundary for the minimum flight range. In this case, the backward search direction is considered, corresponding to a decrease in flight range. The same principles and steps outlined for the maximum flight range analysis are applied in this context.

By propagating an initial estimate of the terminal position in the backward search direction, the procedure aims to assess the reachability of the updated desired terminal position for the minimum flight range. The MPSP approach is utilized to optimize the cost function, taking into account input constraints and system dynamics. The terminal position error is evaluated, and if it falls below the predetermined tolerance threshold and the resulting input vector satisfies the acceleration limit, the updated position is considered a candidate for the reachability boundary for the minimum flight range.

Through an iterative process, similar to that described for the maximum flight range, the reachability boundary for the minimum flight range is determined. This iterative approach allows for a comprehensive exploration of the interceptor's capabilities in terms of the minimum flight range under different flight conditions and system constraints.

CHAPTER 6

REACHABLE SET COMPUTATION

6.1 Unconstrained Input Case

In this section, the calculation procedures for obtaining the reachable set boundaries in the unconstrained input case are described. The maximum range reachability boundary is established by gradually advancing the previous reachable point in the search direction, while the minimum range reachability boundary is obtained through a process of regression from the prior reachable point. For this study, a boundary iteration step size of 300 meters is employed, as it allows for acceptable linearization approximations and convergence to a solution. However, in the event that the algorithm fails to attain a satisfactory solution for the desired terminal position during the current boundary iteration, a stepwise reduction of the boundary iteration step size is implemented, reducing it to 100 meters. This adjustment aims to to discern whether the convergence issue arises from limitations in the interceptor's capability or the potential invalidity of the linearization approximations utilized in the optimization process.

In Figure 40 and Figure 41, the points that can be reached in the inertial reference frame are depicted based on the initial conditions specified in Table 1. The red circle and blue marker points depict the points achieved via calculations for the maximum and minimum boundaries, respectively. In this particular test case, the reachable points lie in the y-z plane since the desired terminal position components align with the inertial y and z axes.

The reachability boundary for the maximum range is approximately 6.8 kilometers, while for the minimum range, it is around 16 meters. Comparing these boundaries with those obtained from the PNG scenario, we observe that the maximum range reachability boundary is approximately 1 km greater. Additionally, the minimum range reachability boundary is situated near the initial position. It is worth noting that the distance between two reachable points is reduced as the process advances during the iterations of the maximum range reachability boundary calculation because of the decrease in boundary step sizes, as discussed earlier.

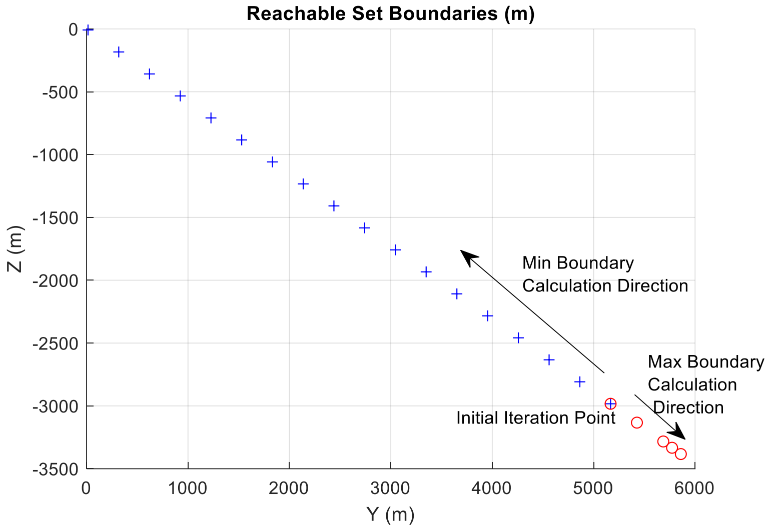


Figure 40: Y-Z plane Reachable Points Along Search Direction

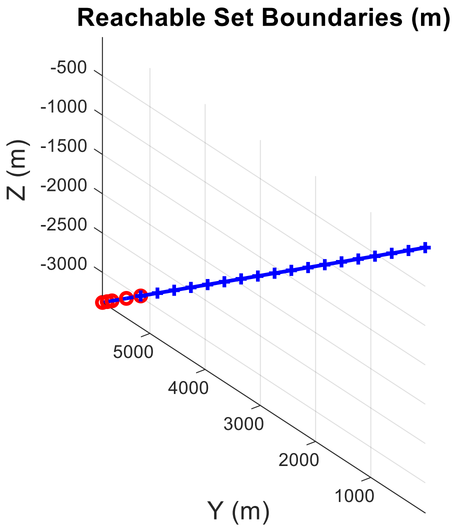


Figure 41: Reachable Points Along Search Direction

Figure 42 consists of two subplots. The first subplot displays the range encompassed by the reachability analysis, considering at most 6 boundary iterations in the specified search direction. The terminal position error at the conclusion of the final MPP iteration is presented in the second subplot for each of the 6 maximum boundary iterations. In the first subplot, the initial point represents the range that the interceptor can reach through PNG. During the second boundary iteration for the maximum range reachability boundary, the first reachable point from the initial iteration is advanced 300 meters further. The ability of the interceptor to reach the new terminal point is then assessed. However, the third boundary iteration presents challenges in finding a viable solution for the interceptor to reach the desired terminal point, as the terminal position error exceeds 50 meters. As a response, in the fourth iteration, the final reachable point from the second boundary iteration is advanced an additional 100 meters, adopting a smaller boundary iteration step size of 100 meters. This adjustment aims to differentiate between the underlying causes of the convergence issue, whether it relates to the interceptor's performance or the linearization approximation. It is important to note that convergence problems may arise when the desired output value is significantly distant from the initial output, as the linearization assumption in calculations may not hold under such circumstances. During the fourth boundary iteration, the MPP gradually approaches a solution with the reduced step size, as the desired terminal position error approaches zero. The subsequent boundary iterations continue with a step size of 100 meters until an MPP solution is no longer achievable, thereby determining the maximum range reachability boundary. For this specific test case, the maximum range reachability boundary is approximately 6800 meters, with a resolution of 100 meters for both the maximum and minimum range reachability boundaries due to the chosen step size.

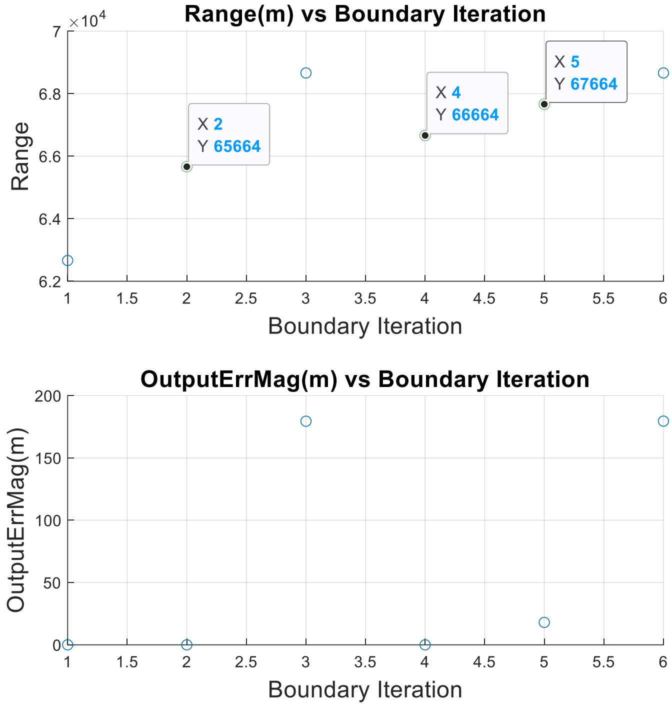


Figure 42: Range and Terminal Position Deviation During Each Maximum Boundary Iteration

Figure 43 and present the cost function values and the desired terminal position error concerning the MPP iterations for each maximum range boundary iteration. Notably, the algorithm exhibits a lack of convergence for the 3rd and 6th boundary iterations. However, for the rest of the boundary iterations, the terminal position error approaches zero after a few iterations.

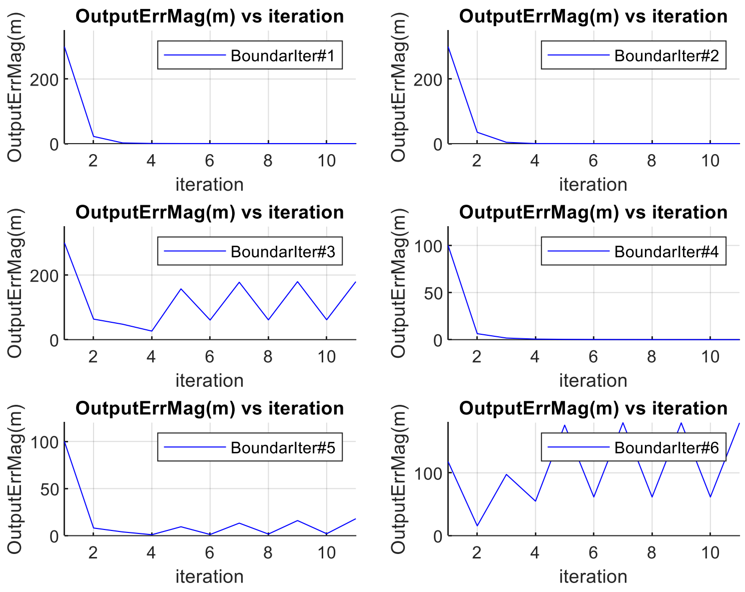


Figure 43: Terminal Position Deviation vs. MPP Iteration

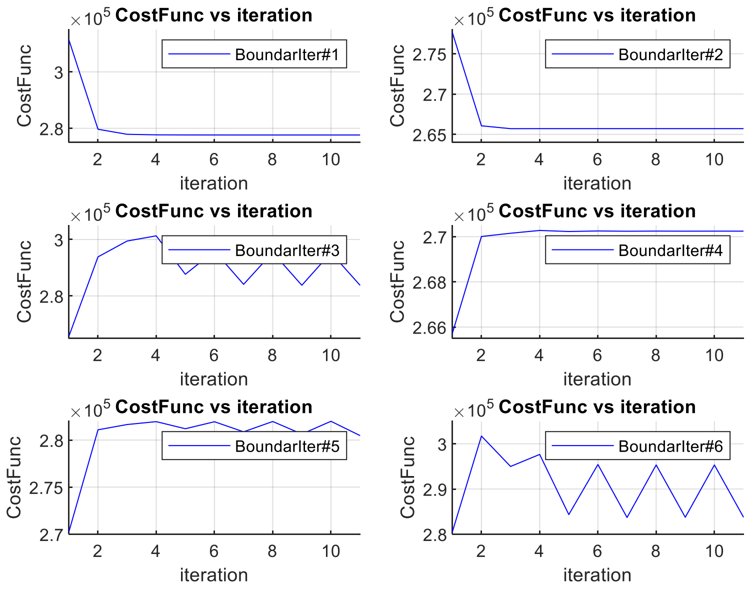


Figure 44: Evolution of Cost Function Across MPP Iterations

Figure 45 and Figure 46 show the cost function values and the desired terminal position error for the 15 MPP iteration case. It is evident that the algorithm fails to converge to a solution for the 3rd and 6th boundary iterations, despite the increased number of iterations. The issue of convergence at the 3rd iteration has been explained previously, and reducing the step size solves the convergence problem up to the 6th boundary iteration.

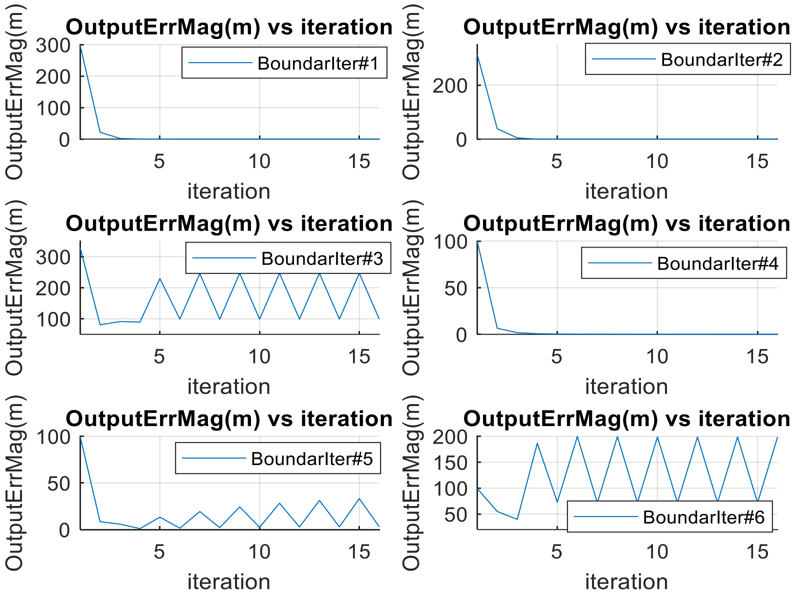


Figure 45: Terminal Position Deviation vs. MPP Iteration

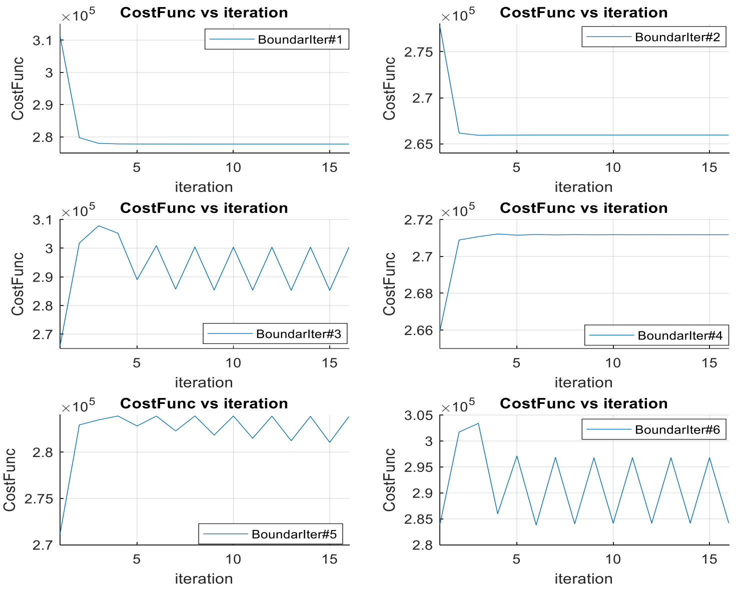


Figure 46: Evolution of Cost Function Across MPP Iterations

To investigate whether the issue of convergence at the 6th iteration is because of the interceptor's capability, the 6th boundary iteration is rerun with the drag force coefficient associated with lateral acceleration $\left(C_{X_a}\right)$ set to zero. Figure 47 demonstrates the convergence results for this case. The results show that the algorithm successfully finds a viable solution when the drag force due to lateral acceleration is neglected. The value of cost function decreases after a few iterations.

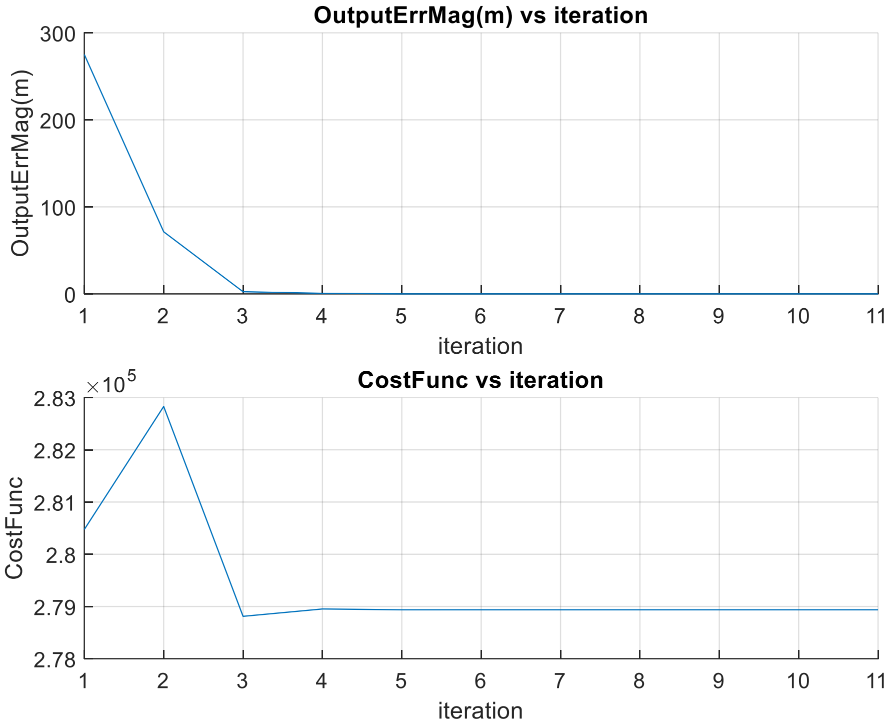


Figure 47: Convergence of Algorithm

Figure 48 and Figure 49 showcase the commanded accelerations on lateral plane derived from the MPP approach during the 6th boundary iteration. The accelerations are represented by red and blue lines, where red indicates the exclusion of drag force impact, and blue represents the inclusion of drag force. The red line indicates that the interceptor can achieve the desired terminal position by utilizing increased acceleration commands. This is because no additional drag force is present due to the greater lateral acceleration, preventing a reduction in the velocity magnitude. The challenge encountered in the 6th maximum range boundary iteration is due to the extra dissipation of the interceptor's kinetic energy as a consequence of the lateral acceleration requirements.

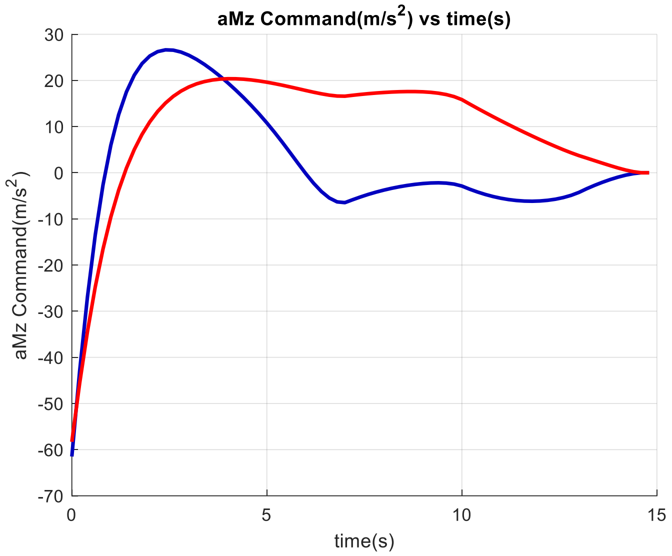


Figure 48: Z Axis Commanded Acceleration for Wind Frame

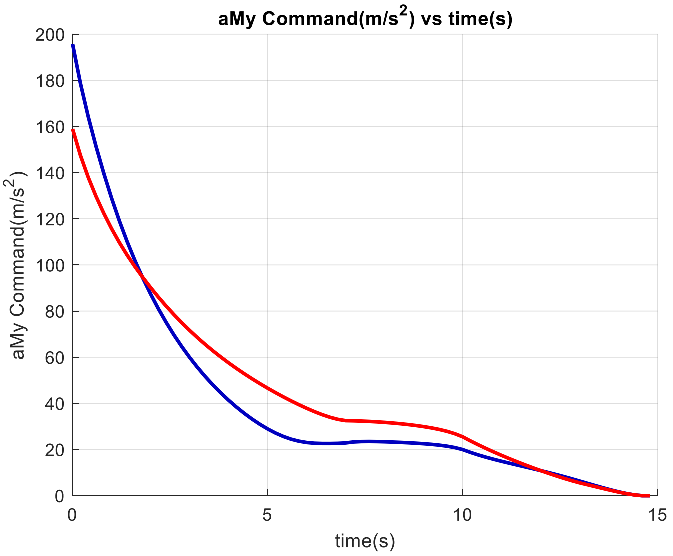


Figure 49: Y Axis Commanded Acceleration for Wind Frame

Figure 50 and Figure 51 present the trajectories of the interceptor in the x-y-z and y-z planes at different minimum boundary iterations. It can be observed that aggressive maneuvering is required to achieve termination points in close proximity to the initial position within a flight duration of 15 seconds.

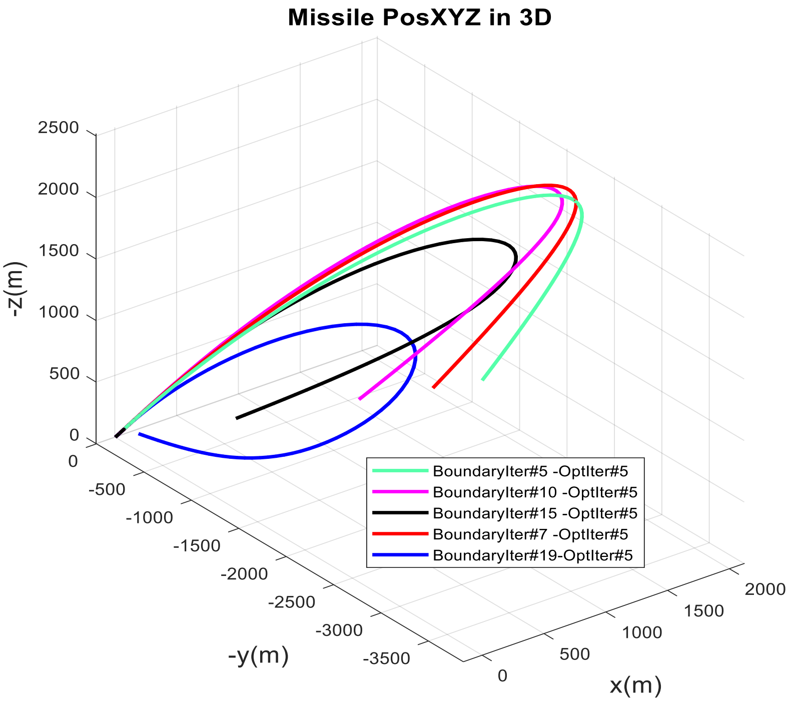


Figure 50: The Trajectory For Different Min. Boundary Iterations (3D)

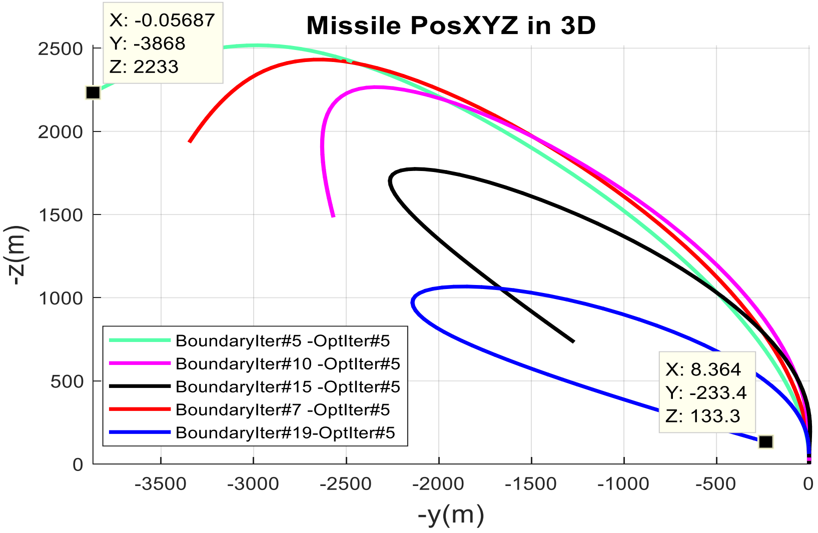


Figure 51: The Trajectory For Different Min Boundary Iterations (Y-Z Plane)

In Figure 52 and Figure 53, the values of the cost function and terminal position deviation for various iterations corresponding to the minimum range reachability boundary are depicted. It can be observed that the terminal position error gradually decreases to zero during the MPP iterations. The cost function value experiences an increase in the second iteration due to the MPP enforcing higher acceleration command magnitudes to satisfy the terminal position equality constraint. In the subsequent iterations, the control effort in terms of acceleration commands gradually decreases while still adhering to the terminal position constraint.

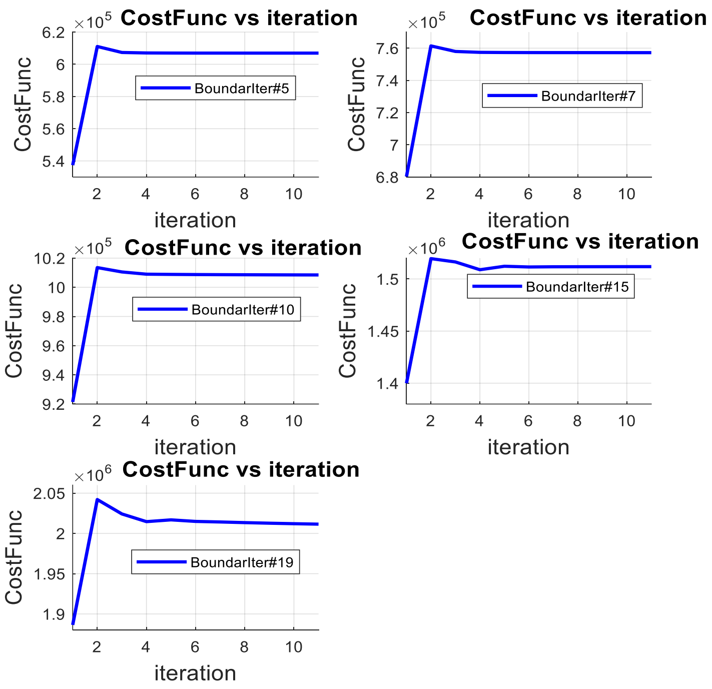


Figure 52: Evolution of Cost Function Across MPP Iterations

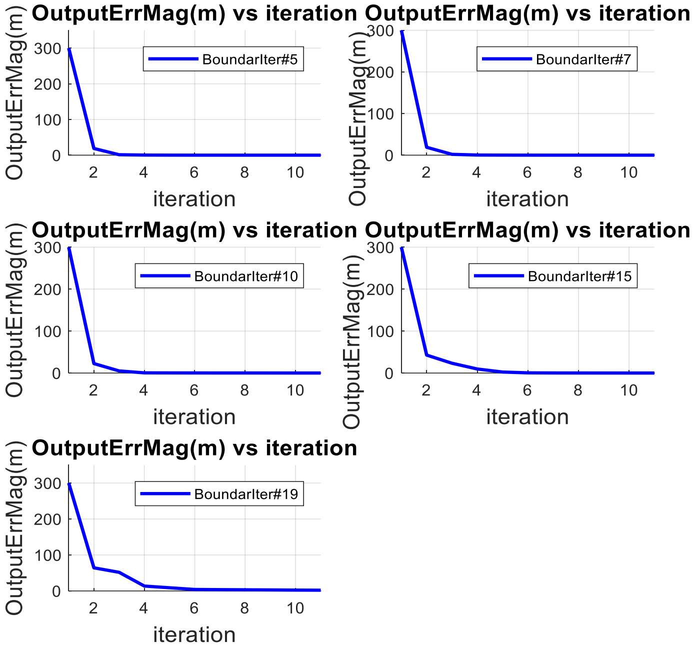


Figure 53: Terminal Position Deviation vs. MPP Iteration

Figure 54 and Figure 55 display the lateral acceleration commands at different minimum boundary iterations. We can observe that higher acceleration commands are required to get to terminal points in close proximity to the initial position. The acceleration commands in the inertial y-axis are higher than those in the z-axis, as the desired terminal points lie along the inertial y-axis. The figures also show that the acceleration commands approach a viable solution within approximately 5 iterations.

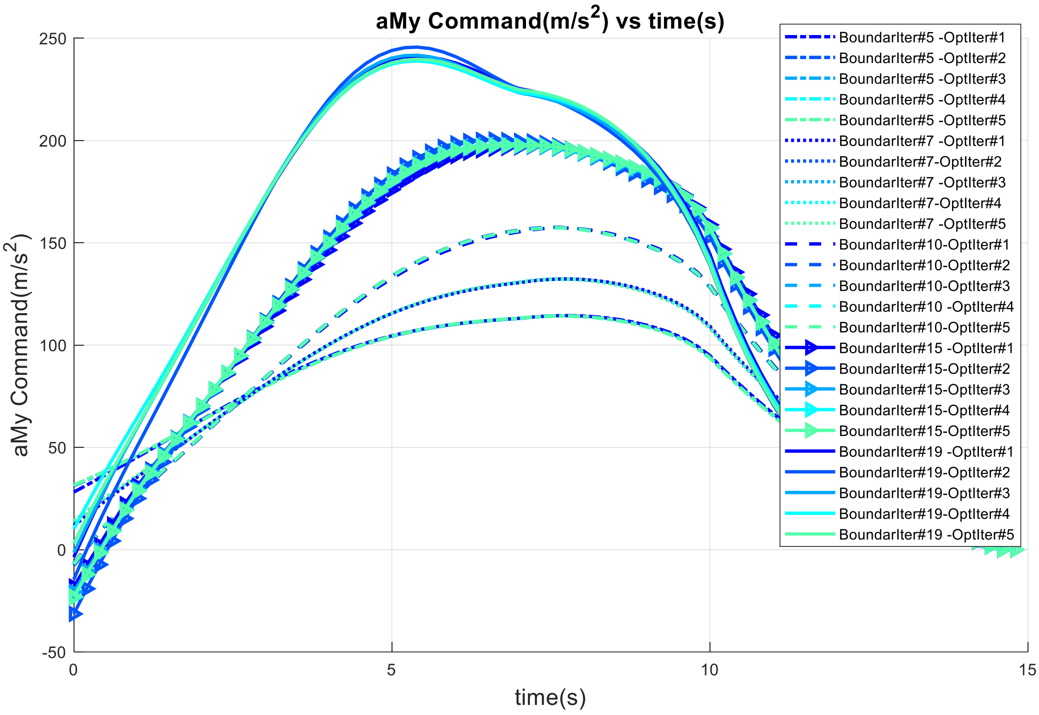


Figure 54: Y Axis Commanded Acceleration for Wind Frame

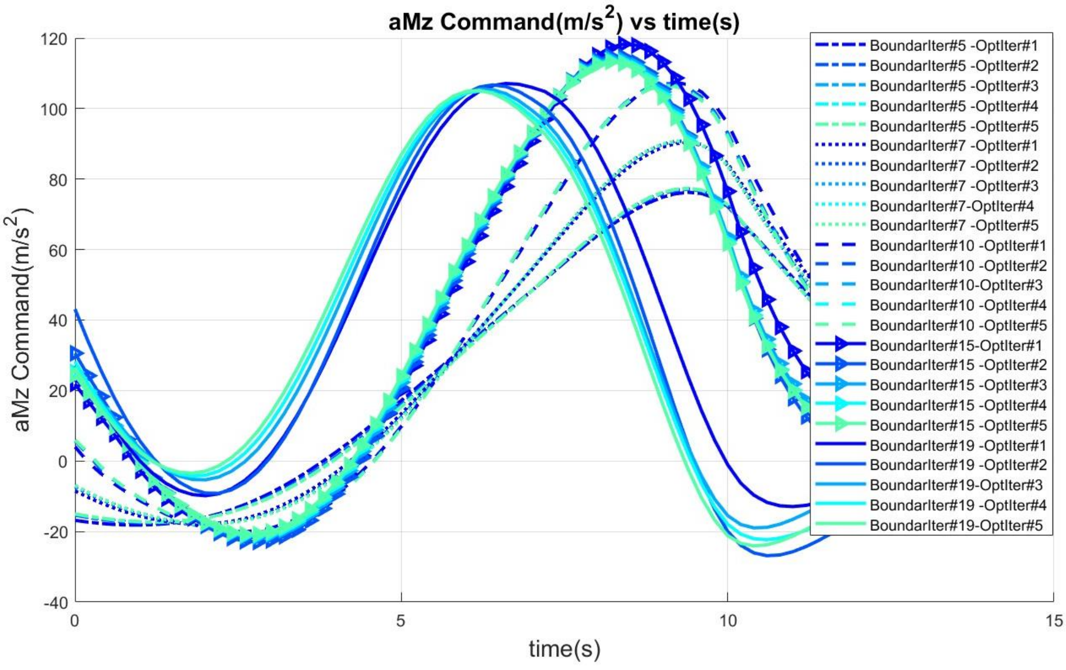


Figure 55: Z Axis Commanded Acceleration for Wind Frame

Figure 56 and Figure 57 demonstrate the flight path angles at different minimum boundary iterations. It is evident that higher path angles are required as the minimum boundary iteration increases to reach terminal points in close proximity to the initial position. After the 10th boundary iteration, the yaw angles exceed 180 degrees, indicating that the interceptor maneuvers in the reverse direction.

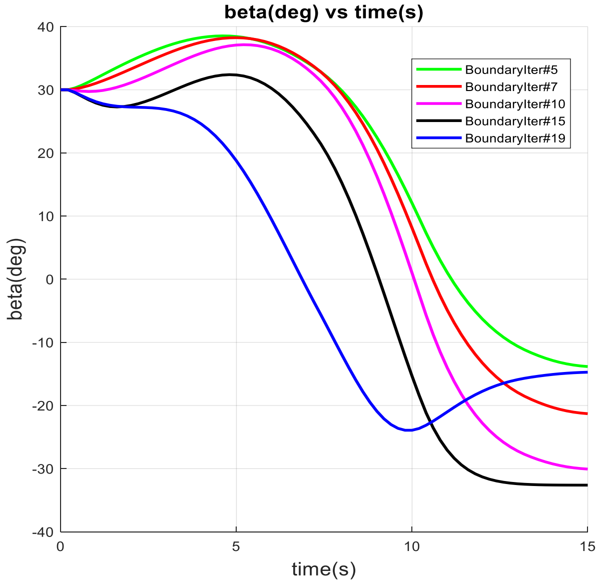


Figure 56: Climb Angle

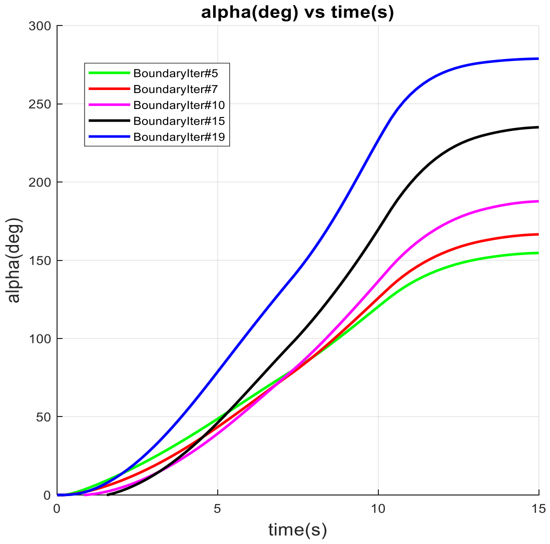


Figure 57: Azimuth Angle

6.2 Input Constraint Case

In this section, the comparison is made between the reachable points achieved using the solution of input constrained problem and those derived in the unconstrained input case. Additionally, the performance evaluation of the input constrained approach includes an analysis of the output error magnitude plot, convergence plot for the Lagrange multipliers of input constraints, and the solution of the input vector (acceleration commands). Notably, the optimization problem for unconstrained input case is initially solved until the desired terminal error decreases below a specified threshold. Subsequently, the resulting input vector is compared against the acceleration limit, which varies based on the Mach number and altitude. Should the input vector exceed the acceleration limit during the flight, the $\bar{\rho}$ vector is iteratively computed using Hildreth's procedure and applied to update the input vector. Finally, the confirmation of the interceptor's ability to reach the desired terminal position is carried out through the implementation of the resultant input vector.

Figure 58 illustrates the points that can be reached in the inertial reference frame for the initial conditions provided in Table 1. Within the figure, the red and blue points portray the achieved positions for the constrained and unconstrained input scenarios, respectively. The figure distinctly illustrates that the kinematic capabilities of the interceptor are influenced by the interceptor's acceleration limit, as anticipated. For the selected search direction, the maximum range reachability boundary is approximately 6.8 km for the unconstrained input case whereas for the constrained input case, it is approximately 6.2 km. Nonetheless, the most notable distinction between the constrained and unconstrained input cases is observed in the minimum range reachability boundary. The minimum range reachability boundary for the constrained input case, within the selected search direction, is approximately 1.7 km farther than that of the unconstrained input case. These results highlight the critical importance of considering input constraints in order to derive realistic reachability boundaries.

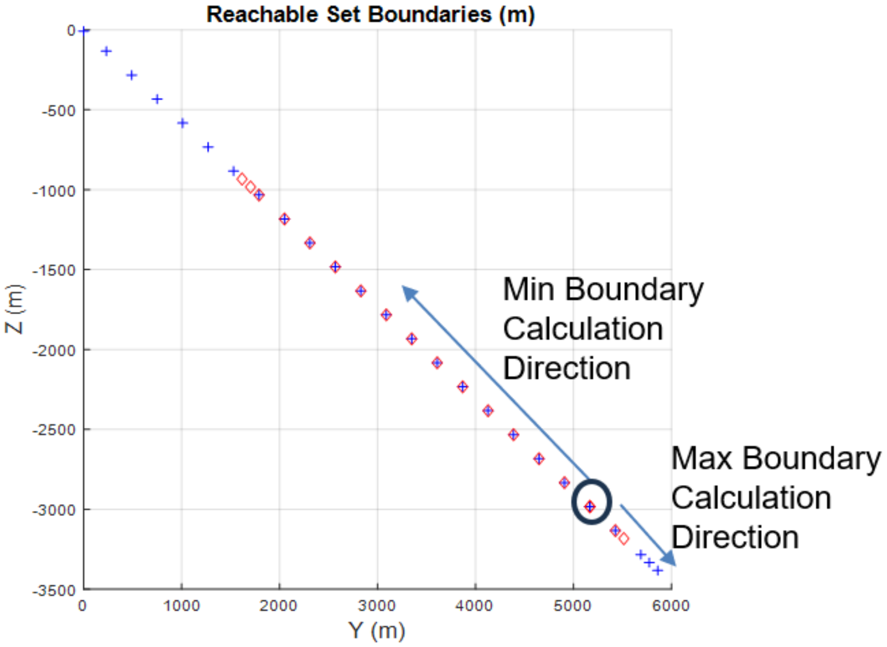


Figure 58: Y-Z plane Reachable Points Along Search Direction

Figure 59 depicts the terminal position deviation for the iterations of the input constrained MPP approach at the 5th, 7th, 10th, and 15th minimum range boundary iterations. Also, the figure shows that the terminal position error meets the chosen tolerance threshold of 50 meters after just 2 MPP iterations for each reachability boundary iteration. The interceptor reaches the desired terminal position during the 5th minimum range reachability boundary iteration without triggering any input constraints. At the 7th, 10th, and 15th reachability boundary iterations, the MPP iteration for the unconstrained input case concludes after the 2nd iteration, as the terminal position error satisfies the defined tolerance. After the second iteration, the input vector solution is used to initialize the calculations for input constrained solution. Subsequently, the input vector solution from the 3rd iteration is applied to validate the terminal position deviation and ensure the reachability.

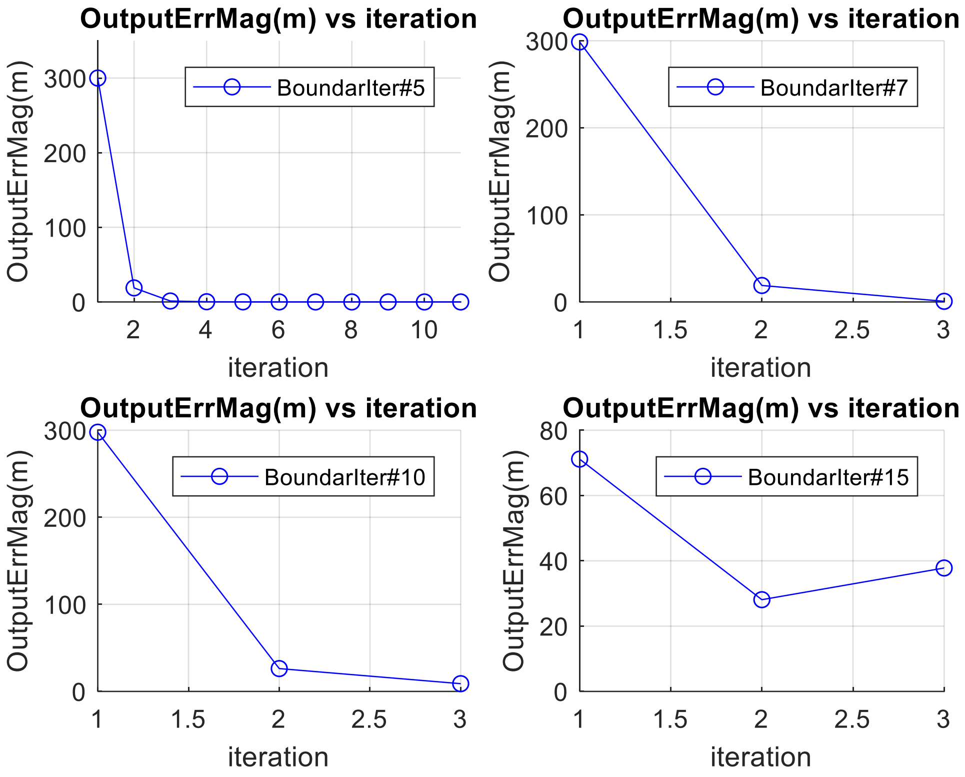


Figure 59: Terminal Position Deviation vs. MPP Iteration

Figure 60 illustrates commanded accelerations at the 5th, 7th, 10th, and 15th minimum range boundary iterations. The red and blue lines represent the lateral and longitudinal accelerations, respectively, while the black lines demonstrate the acceleration limits of the interceptor under the corresponding flight conditions. The figure demonstrates that higher acceleration commands are necessary to achieve points closer to the initial position. Consequently, the input constraints are activated in proximity to the initial points, and the interceptor generates the maximum acceleration command that it is capable of within the constraint activations. As the interceptor approaches the initial points, it generates acceleration commands at the acceleration limit for an extended period.

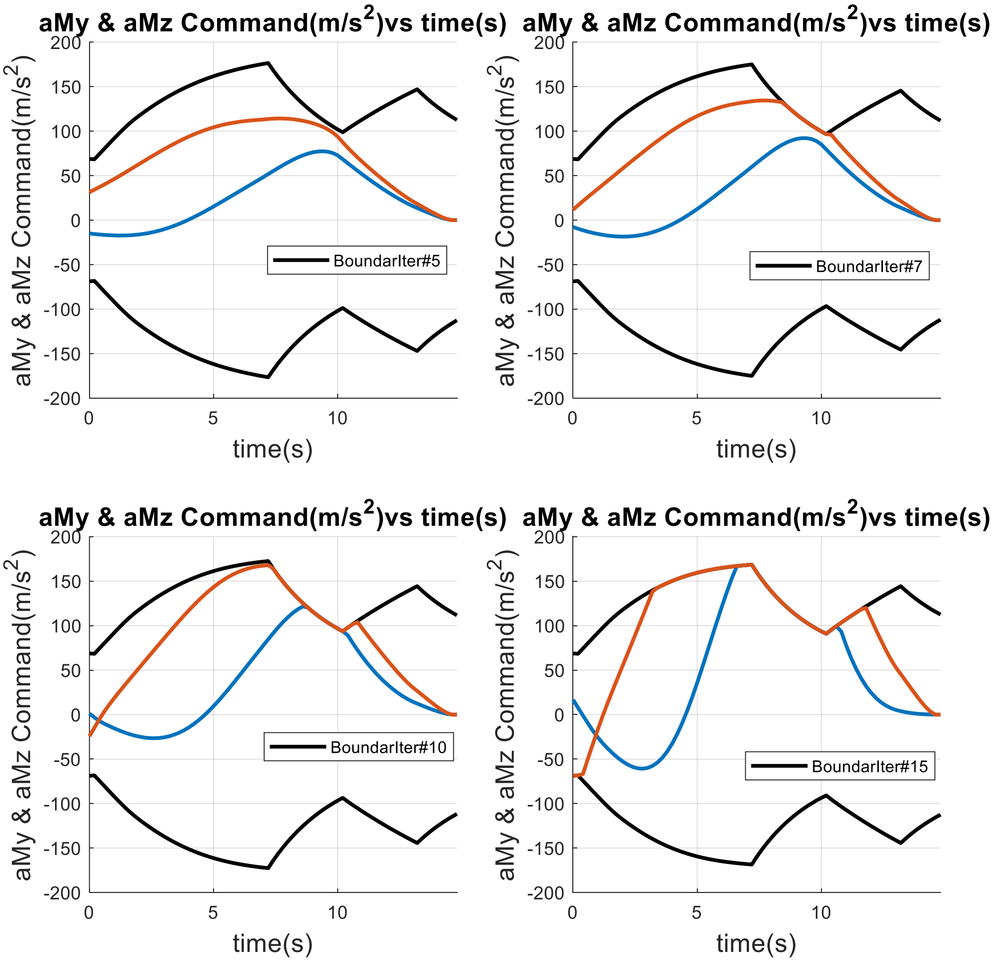


Figure 60: Commanded Accelerations in Wind Frame

To evaluate the convergence behavior of the Hildreth's procedure, the discrepancy between successsive iterations of the $\bar{\rho}$ vector (denoted as $\bar{\rho}_{diff}$) is examined. Figure 61 displays the dot product of $\bar{\rho}_{diff}$ obtained during the iterations of the $\bar{\rho}$ vector for the 7th , 10th , 15th minimum range reachability boundary iterations From the iterations, it can be deduced that the $\bar{\rho}$ vector converges within 4, 6, 29 iterations, for the 7th, 10th and 15th boundary iterations, respectively. The iteration of $\bar{\rho}$ is stopped when it reaches the maximum allowable iteration number at the 16th boundary iteration, which represents the limit of the minimum boundary. At this point, the interceptor struggles to reach the desired final position while meeting the input constraints throughout the flight. As a result, the convergence of $\bar{\rho}$ values slows down at the boundary limits or may not converge to values that indicate the interceptor's capability has already been reached in previous iterations. The $\bar{\rho}$ values at the end of the total $\bar{\rho}$ iteration in the 15th boundary iteration are used in the final calculation of the input vector, and error plot for output and input vector are presented in Figure 59 and Figure 60, respectively.

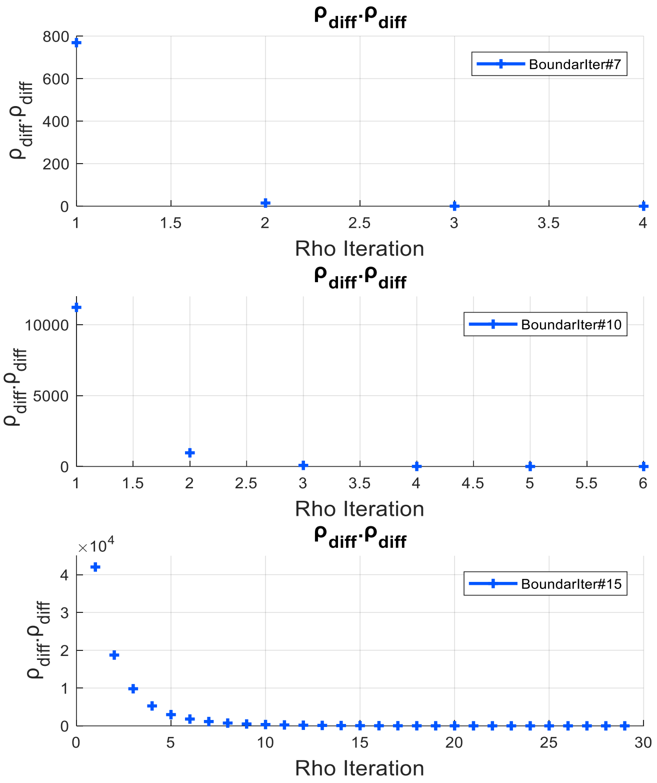


Figure 61: Difference in Rho Values between Consecutive Rho Iterations

Figure 62, Figure 63 and Figure 64 present the progression of $\bar{\rho}$ elements for the 7th, 10th and 15th minimum range reachability boundary iterations respectively. The figures show a reduction in the disparity between successive $\bar{\rho}$ iterations, indicating $\bar{\rho}$ converges to the solution after an adequate number of iterations. Figure 62 to Figure 67 depict the evolution of the elements of the $\bar{\rho}$ vector for the 7th, 10th, and 15th minimum range reachability boundary iterations. These figures demonstrate that the difference between subsequent iterations of the $\bar{\rho}$ vector decreases, indicating the convergence of $\bar{\rho}$ to the solution after a considerable number of iterations. The $\bar{\rho}$ values for each iteration are presented in Figure 65 to Figure 67, revealing that the $\bar{\rho}$ values become closer with each iteration.

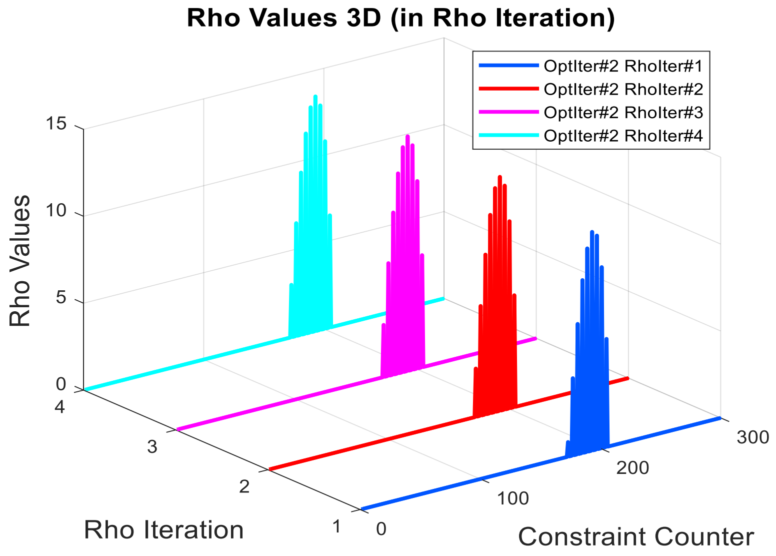


Figure 62: Three-Dimensional Rho Values at 7th Boundary Iteration

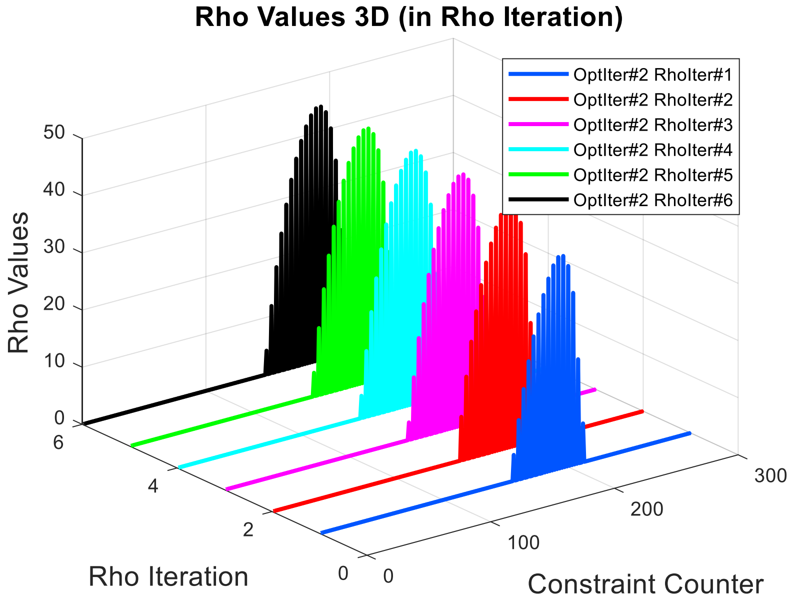


Figure 63: Three-Dimensional Rho Values at at 10th Boundary Iteration

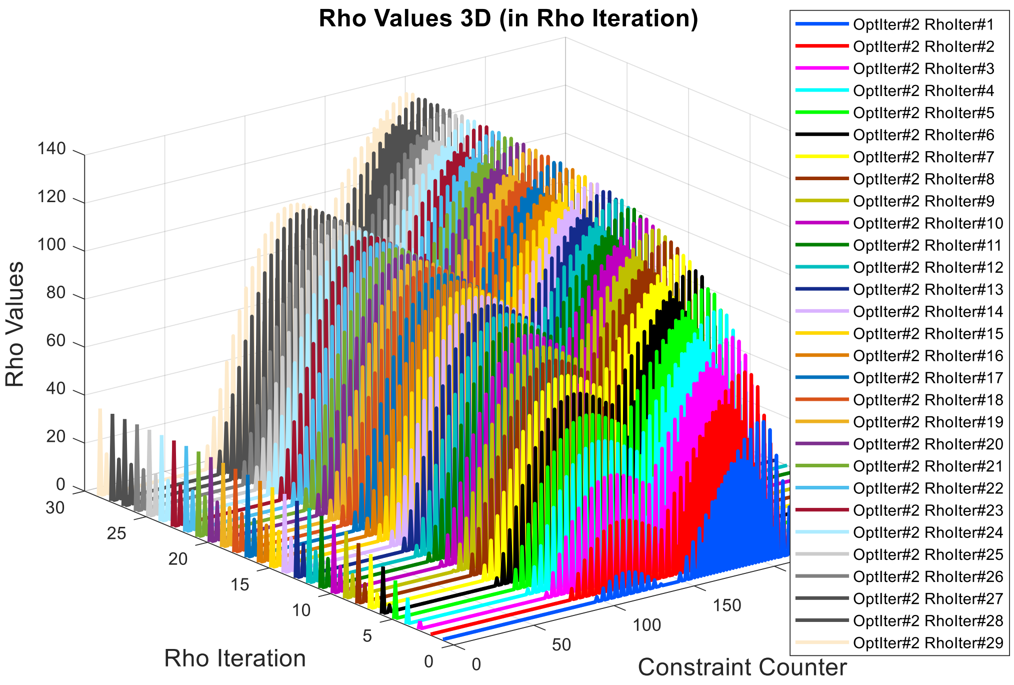


Figure 64: Three-Dimensional Rho Values at 15th Boundary Iteration

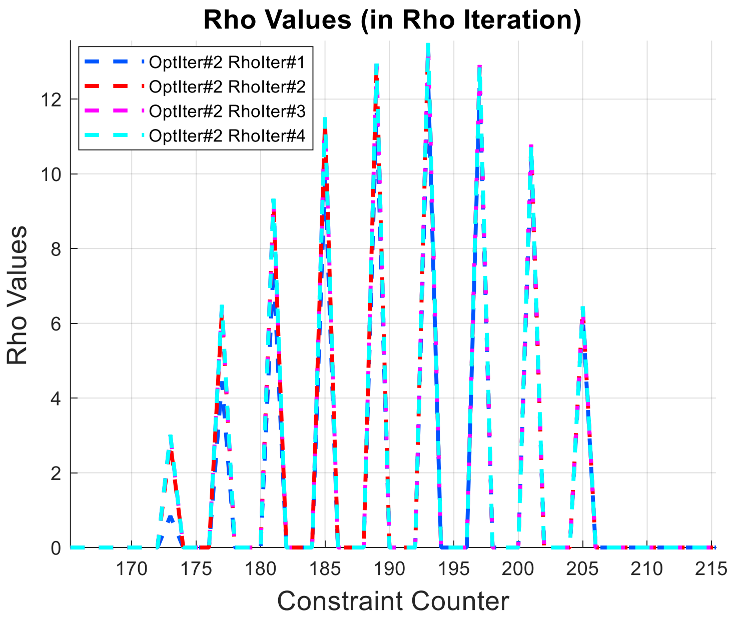


Figure 65: Rho Values for each Rho Iteration at 7th Boundary Iteration

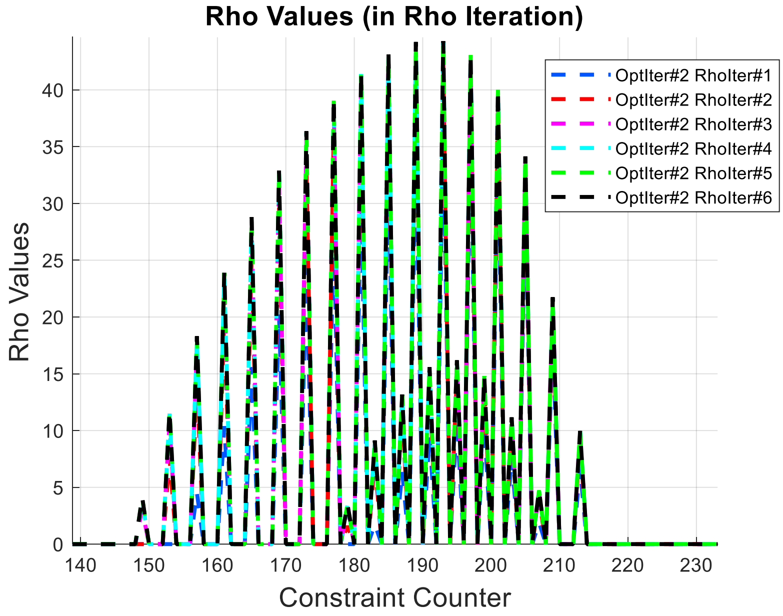


Figure 66: Rho Values for each Rho Iteration at 10th Boundary Iteration

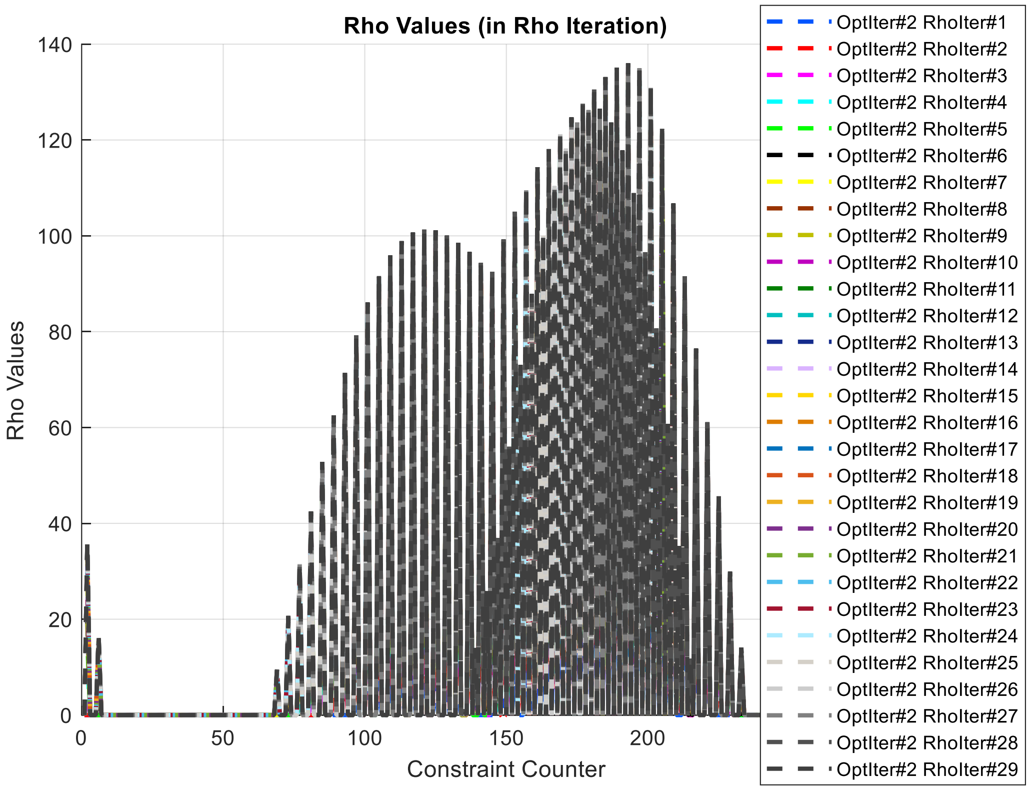


Figure 67: Rho Values for each Rho Iteration at 15th Boundary Iteration

Figure 68 and Figure 69 showcase the reachable set boundaries and reachable set points for the scenario provided in Table 2. Within the left figures, the red and blue points portray the achieved positions for the minimum and maximum range reachability boundaries, respectively. The reachable points are indicated by the blue and black lines in the middle figures for different search directions. Within the right figure, the trajectories related to minimum range reachability boundary computation is illustrated for $\alpha_s = \beta_s = 0°$ and related acceleration responses are displayed in Figure 69 .

Table 2: Scenario Initial Conditions (Input Contsraint Case)

β_0	40 deg
h_0	0 m
V_{M_0}	200 m/s
t_{go}	15 s
α_s	$0°, 20°, 40°, 60°, 80°$
β_s	$-30°, -15°, 0°, 15°, 30°$

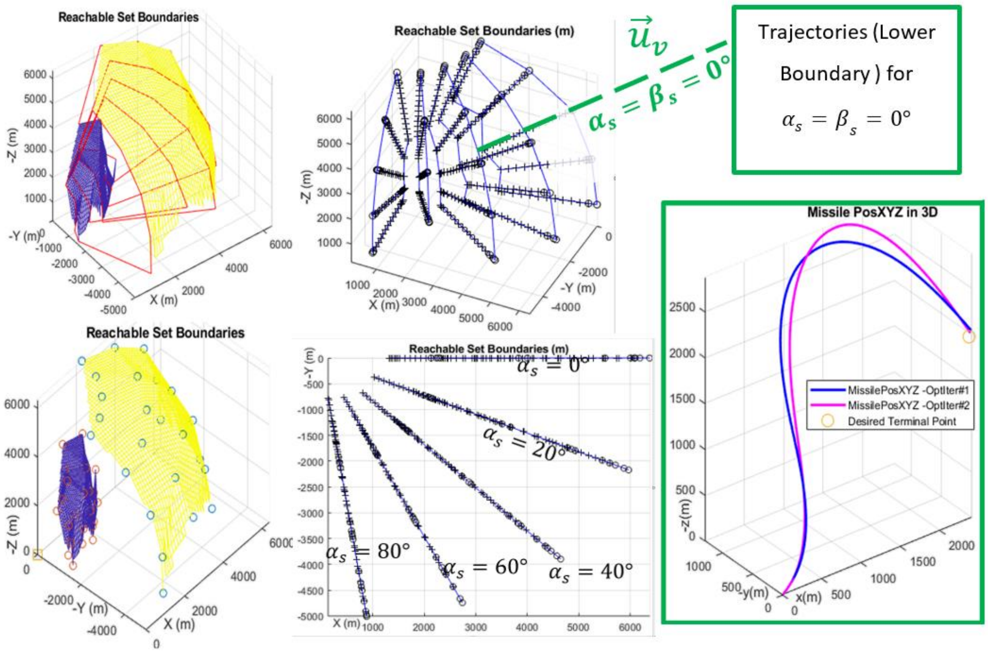


Figure 68: Visualization of Reachable Set and Trajectories for a Specific Search Direction

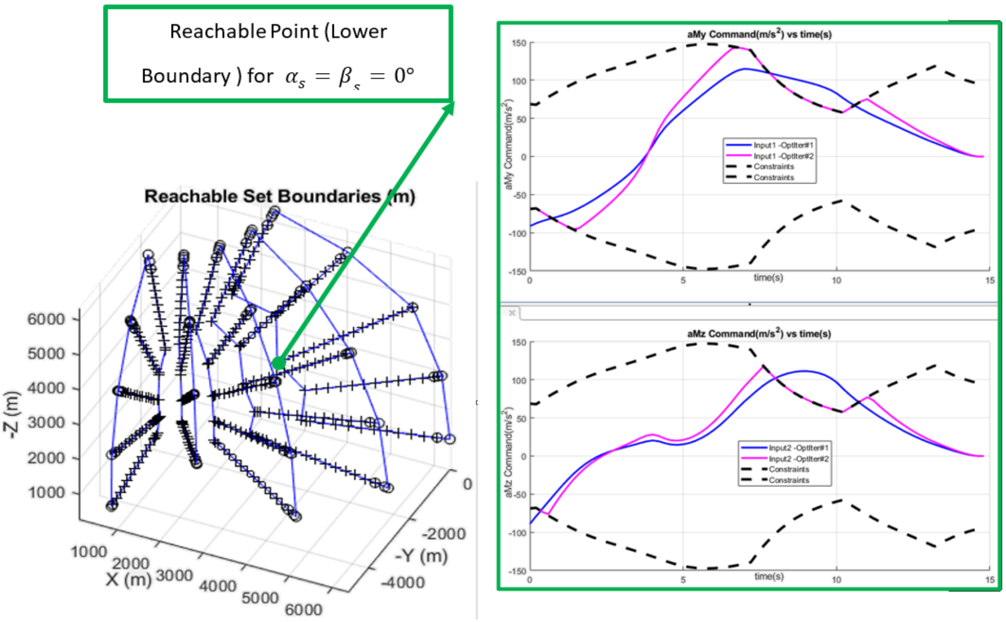


Figure 69: Visualization of Reachable Set Boundaries and Acceleration Response for a Specific Search Direction

6.3 Initial Condition Variations

In this section, the reachability boundaries for different combinations of initial conditions, namely t_{go}(flight duration), h (altitude) and β (climb angle), are examined. By exploring these variations, insights into how the kinematic capabilities of the interceptor are affected by changes in these parameters are gained.

6.3.1 Unconstrained Input Case

In this section, the analysis of reachability boundaries in the absence of input constraints is presented. The reachability boundaries are examined for different test cases and search angles.

To conduct the analysis, five test cases are defined, each representing a unique combination of initial conditions. Table 3 provides a summary of the initial conditions for each test case, including the values of β , h and t_{go}.

Table 3: Initial Conditions

Test Case	β_0	h_0	V_{M_0}	t_{go}
1	30 deg	0 m	200 m/s	15 s
2	30 deg	0 m	200 m/s	25 s
3	30 deg	3 km	200 m/s	15 s
4	30 deg	3 km	200 m/s	25 s
5	10 deg	0 m	200 m/s	15 s

Figure 70 and Figure 71 showcase the reachable points for test cases 1 and 2 when $\alpha_s = 90°$ and $\beta_s = 0°$. The reachable points are indicated by the blue and black lines. The figures illustrate that the maximum reachable boundary for the 25-second case extends approximately 2-2.5 kilometers beyond the boundary of the 15-second test case. Evidently, as the flight duration increases, the interceptor can cover longer distances as long as it possesses sufficient energy. However, reaching terminal positions in close proximity to the initial position becomes more challenging as the flight duration increases.

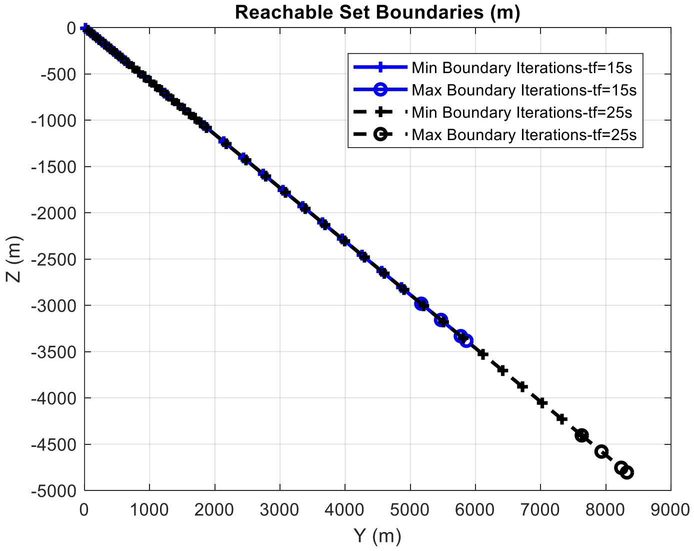


Figure 70: Visualization of Reachable Points

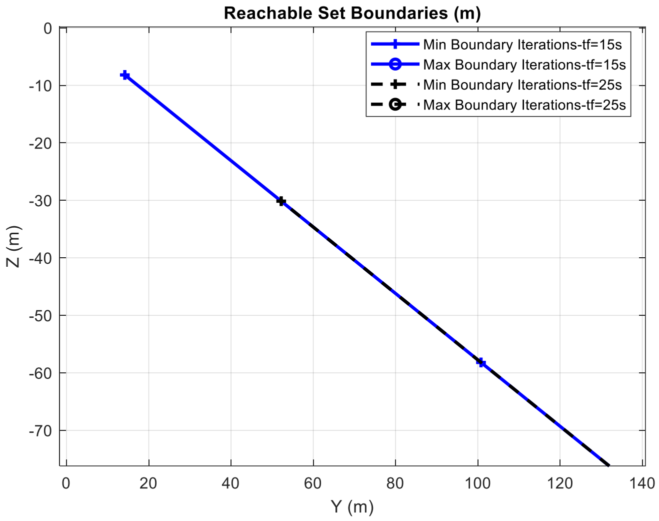


Figure 71: Visualization of Reachable Points (Zoomed)

Moving on to test cases 3 and 4, Figure 72 display reachable points for $\alpha_s = 90°$ and $\beta_s = 0°$. Comparing the reachability boundaries in Figure 72 to those in Figure 70, it is evident that the maximum range reachability boundary expands with an increase in initial altitude. This observation indicates that the interceptor's reachability is influenced by the initial altitude, enabling the interceptor to cover larger distances when operating at higher altitudes.

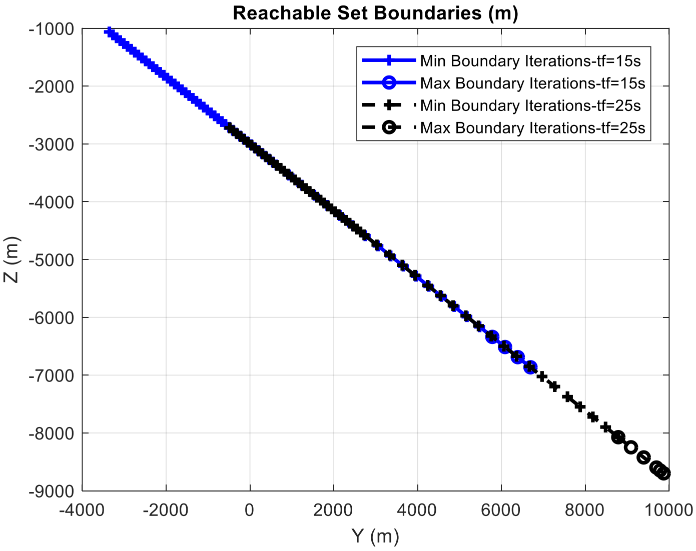


Figure 72: Visualization of Reachable Points

Shifting the focus to a different set of search angles, Figure 73 to Figure 77 depict the reachable sets and reachability boundaries for test case 1. Figure 73 showcases the reachable points for $\alpha_s = 0°$, $\beta_s = 30°$, providing insights into the reachability in the X-Z plane.

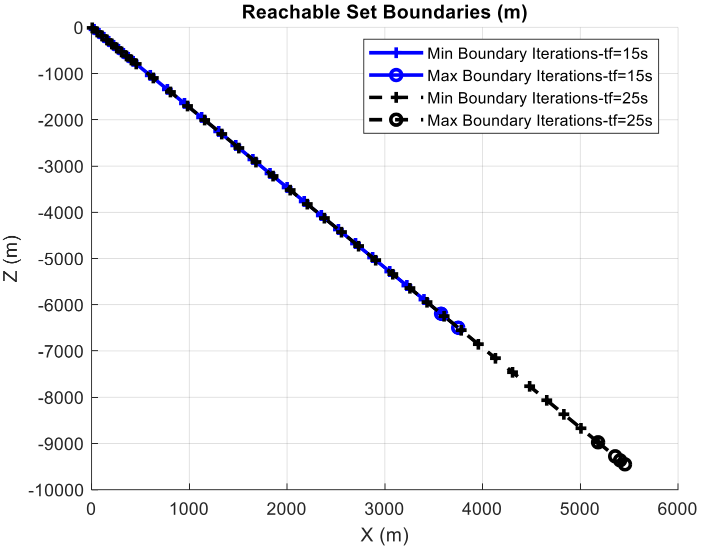


Figure 73: Visualization of Reachable Points

Figure 74 presents the overall reachability zone, while Figure 75 specifically displays the reachability boundary for test case 1. These figures demonstrate the reachability boundaries obtained for different search angle values ($\alpha_s = 0°, 15°, \cdots, 75°, 90°$ and $\beta_s = -90°, -75°, \cdots, 75°, 90°$). The reachable points are represented by black markers, while the blue line indicates the reachability boundaries.

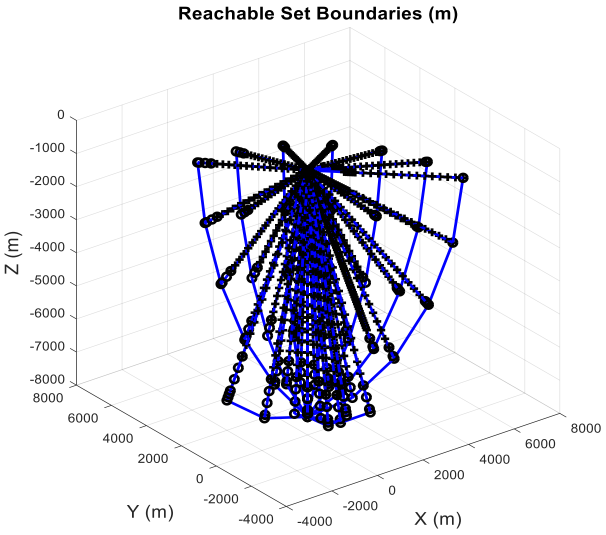


Figure 74 Visualization of Reachable Set

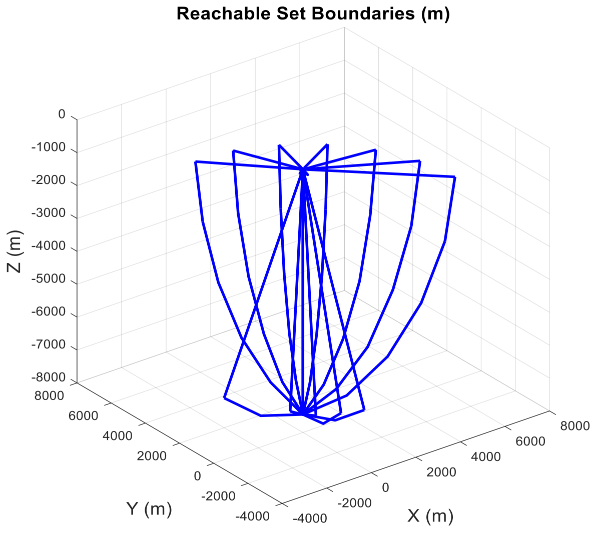


Figure 75: Visualization of Reachable Set Boundary

Furthermore, Figure 76 to Figure 81 present the reachable sets and reachability boundaries from different perspectives, such as the inertial X-Y, X-Z, and Y-Z planes. Each line in these figures represents the reachability boundary for a specific α_s value, with the markers representing the reachable points for different β_s values. It is notable that the limits of reachability within the inertial X-Y plane exhibit symmetry in terms of the search direction line of $\alpha_s = \beta_s = 0°$, as expected. However, the reachability boundaries in the X-Z and Y-Z planes deviate from symmetry due to the influence of β_s values, altitude changes, and aerodynamic drag forces. The variations in altitude affect atmospheric properties and aerodynamic drag forces, thereby altering the path along the Z-axis and impacting the reachability boundaries.

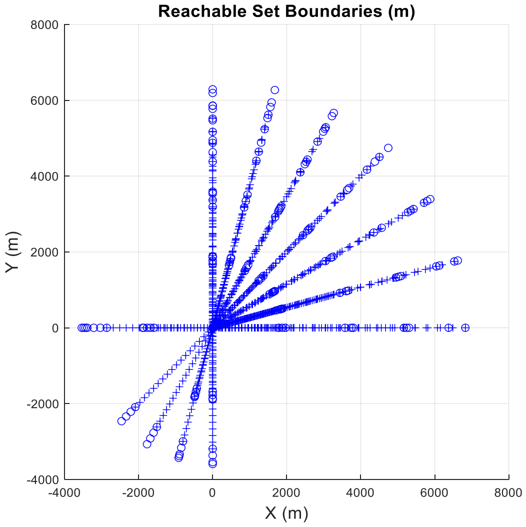


Figure 76: Visualization of Reachable Points

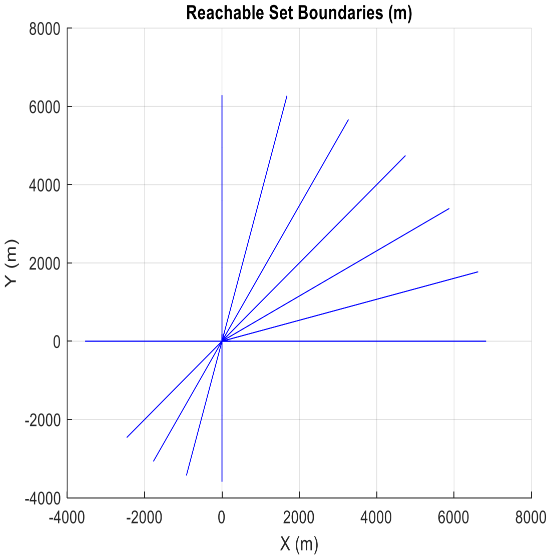


Figure 77: Visualization of Reachable Set Boundary

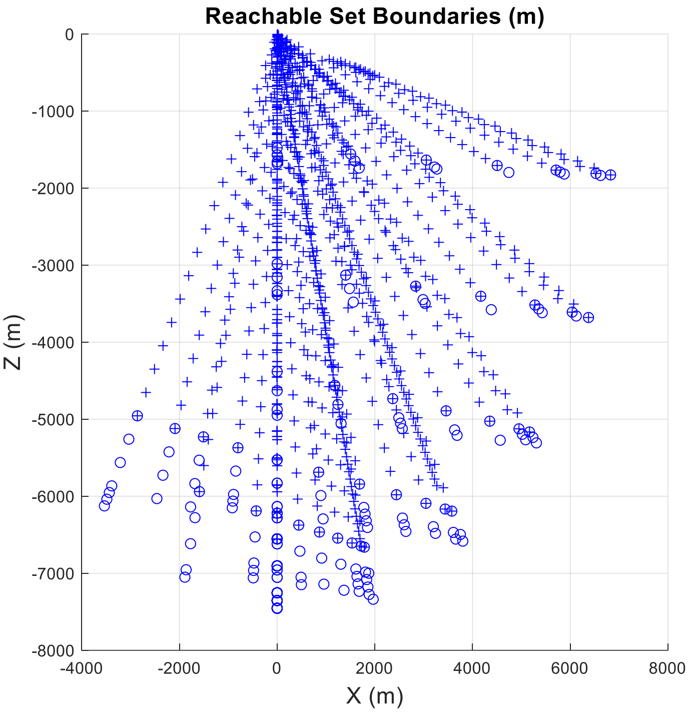


Figure 78: : Visualization of Reachable Points (X-Z plane)

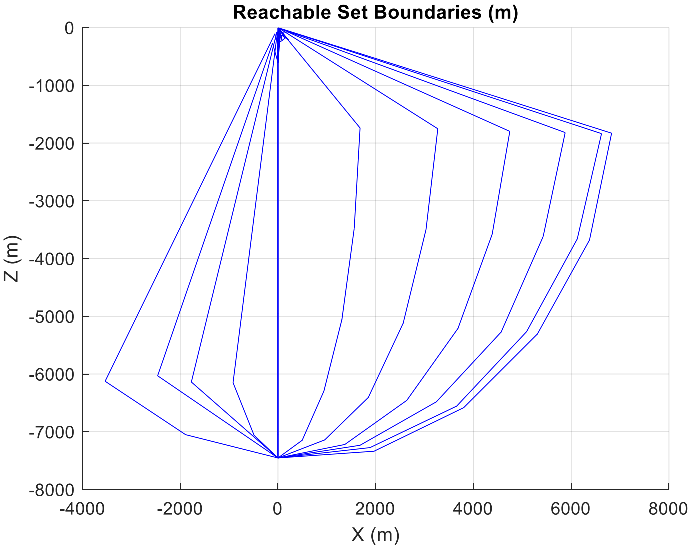


Figure 79: : Visualization of Reachable Set Boundary (X-Z plane)

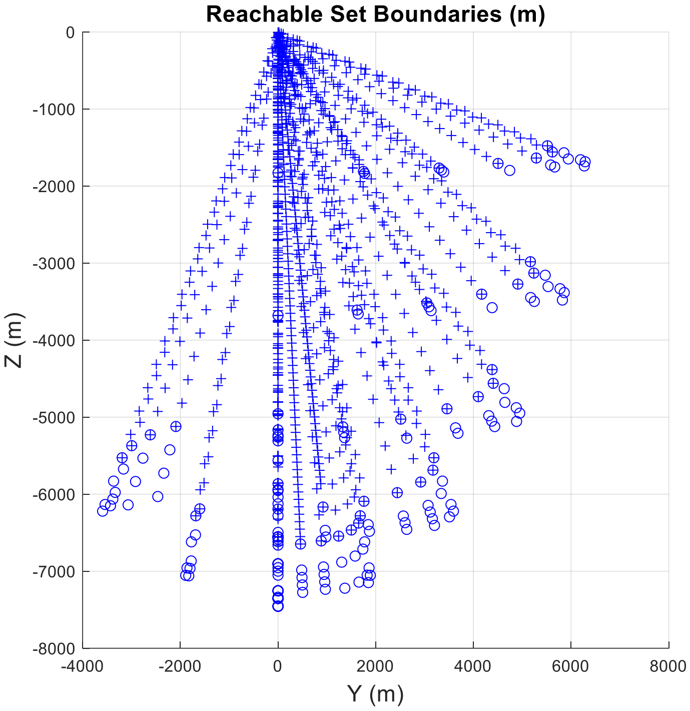


Figure 80 : Visualization of Reachable Points (Y-Z plane)

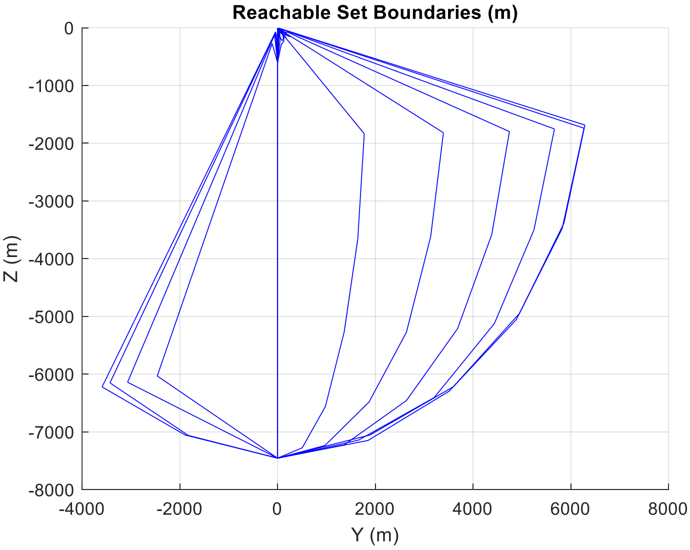


Figure 81: : Visualization of Reachable Set Boundary (Y-Z plane)

Finally, Figure 82 provides an overview of the reachability boundaries for test cases 1 and 5, represented by the blue and red lines corresponding to $\beta_0 = 30°$ and $\beta_0 = 10°$, respectively.

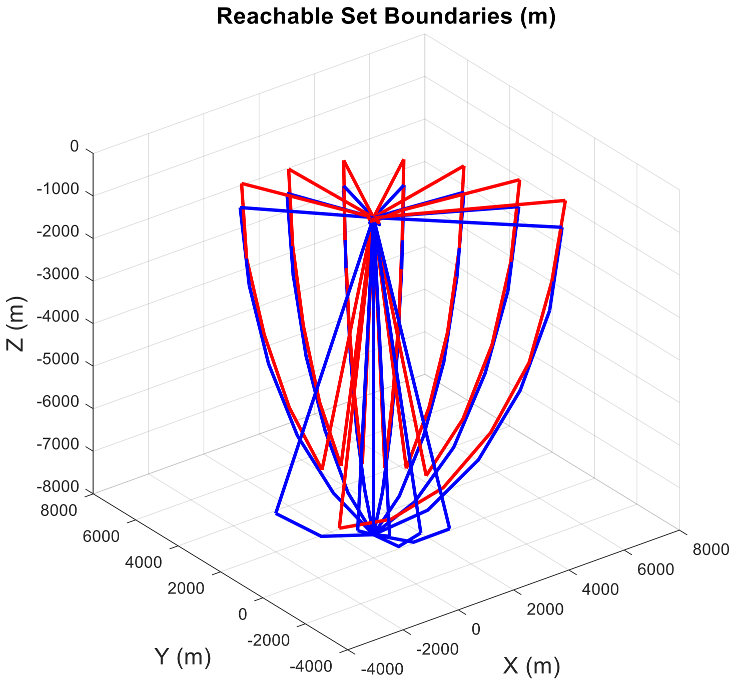


Figure 82: : Visualization of Reachable Set Boundary For Different Initial Path Angle

Subsequently, Figure 83 to Figure 86 present the reachability boundaries for specific α_s values separately. These figures highlight the impact of different α_s angles on the reachability boundaries, with the blue lines indicating the 30-degree cases and the red lines representing the 10-degree cases. The reachability boundaries expand with higher β_0 angles, particularly in terms of reaching higher altitudes.

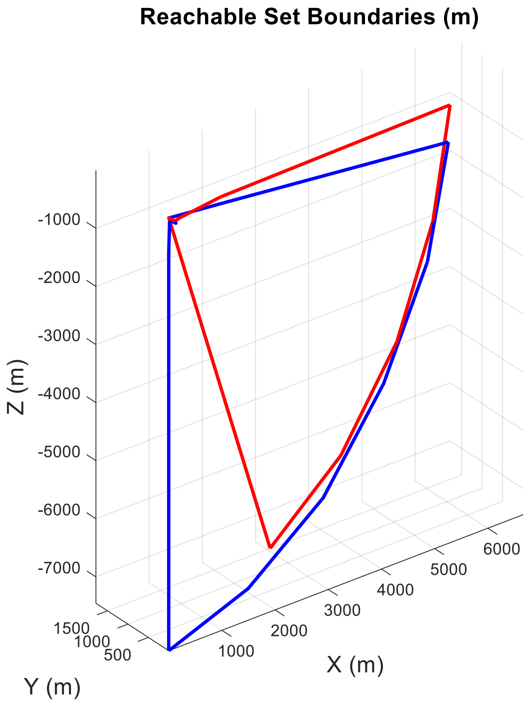


Figure 83: Visualization of Reachable Set Boundary For $\alpha_s = 15°$

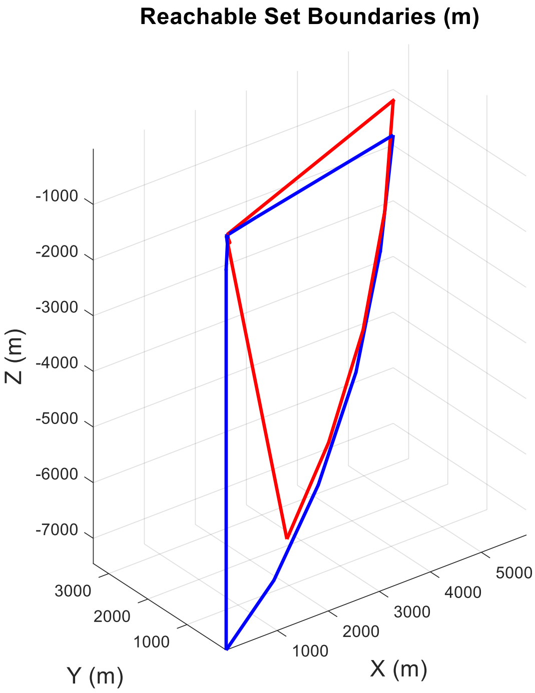


Figure 84: Visualization of Reachable Set Boundary For $\alpha_s = 30°$

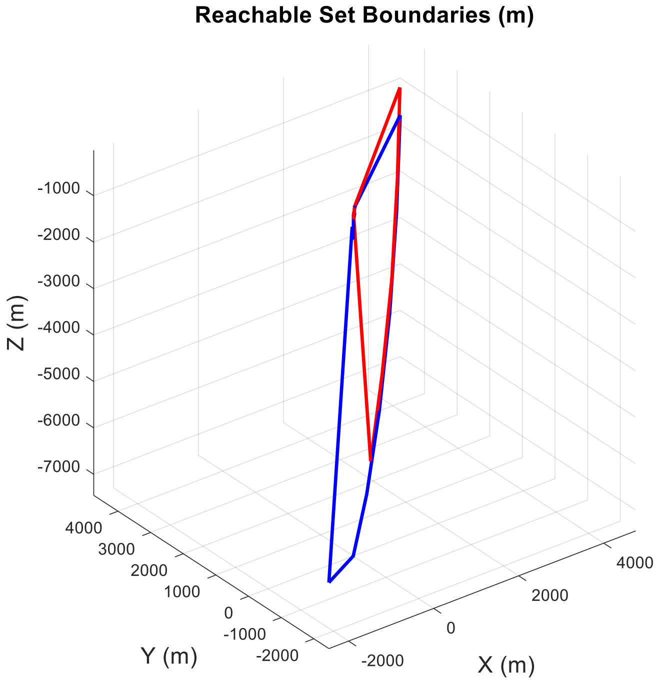


Figure 85: Visualization of Reachable Set Boundary For $\alpha_s = 45°$

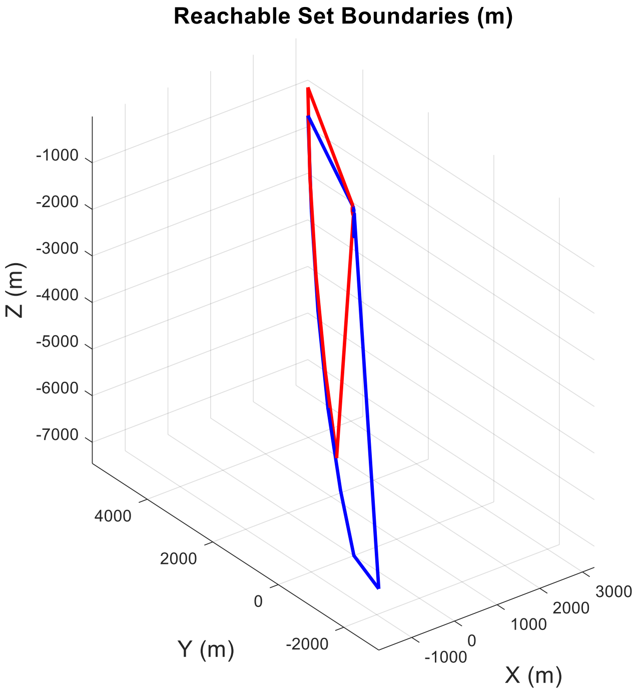


Figure 86: Visualization of Reachable Set Boundary For $\alpha_s = 60°$

6.3.2 Input Constraint Case

In this section, the analysis of reachability for input constraint case is presented. The reachability boundaries are examined for different test cases and search angles.

To conduct the analysis, five test cases are defined. Table 4 provides a summary of the initial conditions for each test case, including the values of β_0, h_0,${V_M}_0$and t_{go}.

Table 4: Initial Conditions (Input Constraint Case)

Test Case	β_0	h_0	${V_M}_0$	t_{go}
1	40 deg	0 m	200 m/s	15 s
2	40 deg	0 m	200 m/s	10s s
3	40 deg	0 m	200 m/s	7.5 s
4	30deg	0 km	200 m/s	15 s
5	45 deg	0 km	200 m/s	15 s
6	55 deg	0 m	200 m/s	15 s
7	40 deg	7 km	200 m/s	15 s
8	40 deg	7 km	600 m/s	15 s

In this section, the results of the analysis of maximum and minimum range reachability boundaries for various flight durations (test cases 1,2,3) are presented, as shown in Figure 87. These figures demonstrate the reachability boundaries obtained for different search angle values ($\alpha_s = 0°, 20°, \cdots ,80°$ and $\beta_s = -30°, -15°, \cdots ,30°$). As previously observed in Section 6.3.1, it is noted that the interceptor's capability to travel longer distances increases with the extension of flight time, provided it possesses sufficient energy. Consequently, the maximum range reachability boundary expands correspondingly with the rise in flight duration. Additionally, Figure 88 and Figure 89 depict the reachability boundaries for different initial path angles (test cases 4,5,6). These figures demonstrate the reachability boundaries obtained for different search angle values ($\alpha_s = 0°, 15°, \cdots ,80°$ and $\beta_s = -25°, -10°, 10°, 25°, 30°$). It is evident from these figures that the reachability boundaries expand significantly with higher β_0 angles, particularly concerning reaching greater altitudes. Furthermore, it is essential to emphasize that both the area and shape of the reachable set are subject to substantial influence by variations in t_{go}

(time to go) and β_0. Hence, it becomes imperative to define the reachable set for different initial β_0 values, as well as time to go parameters when integrating it into the guidance computer.

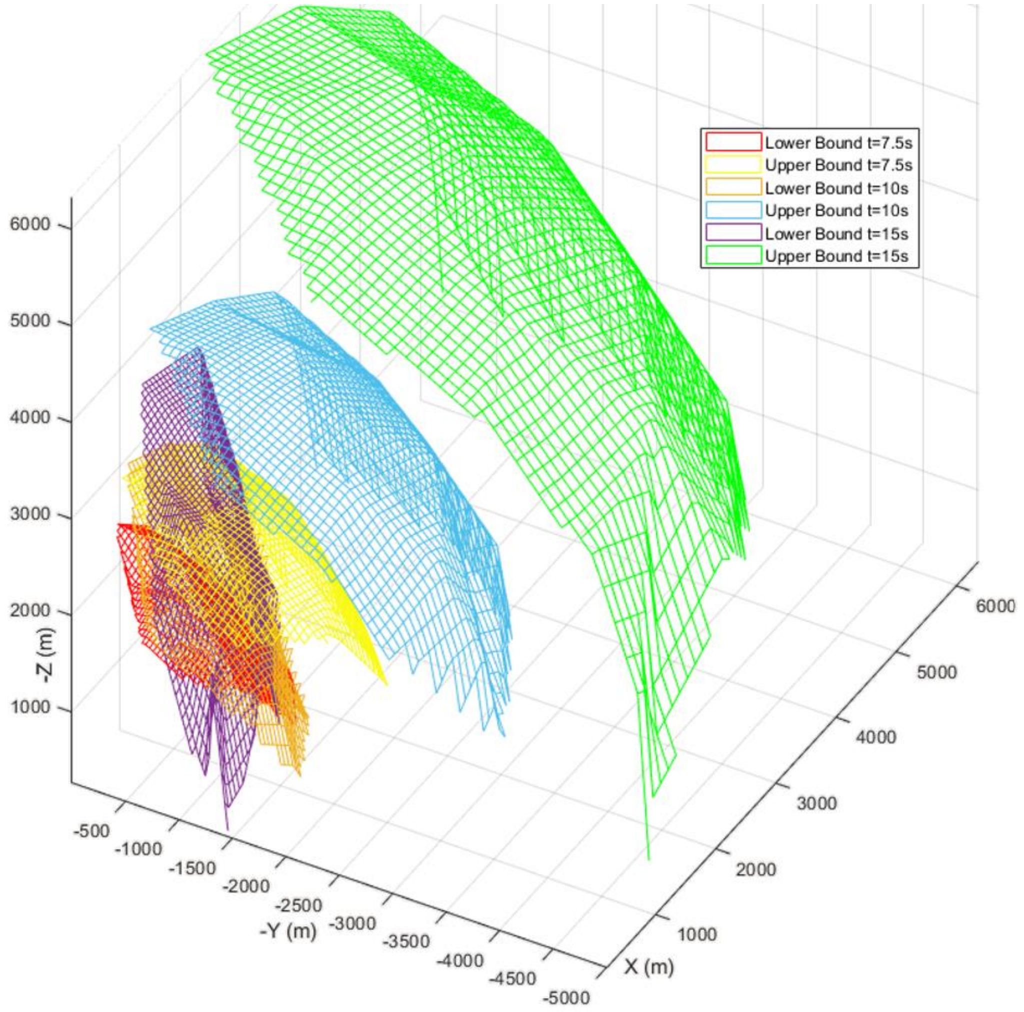


Figure 87: Visualization of Reachable Set Boundary for Different Flight Duration

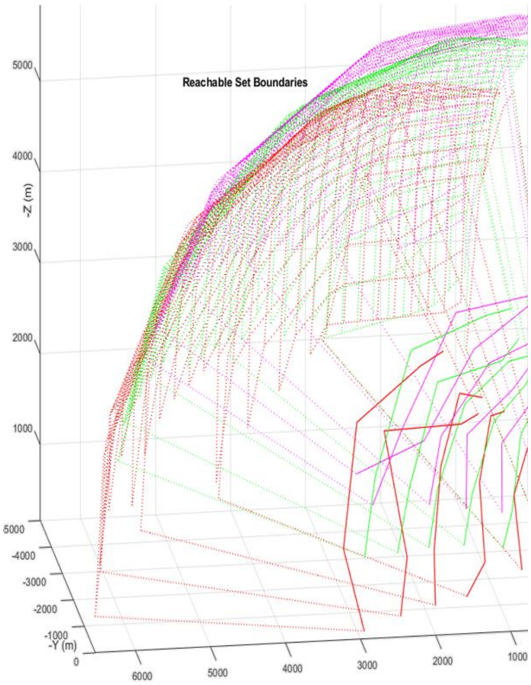


Figure 88: Visualization of Reachable Set Boundary for Different Initial Flight Path Angle

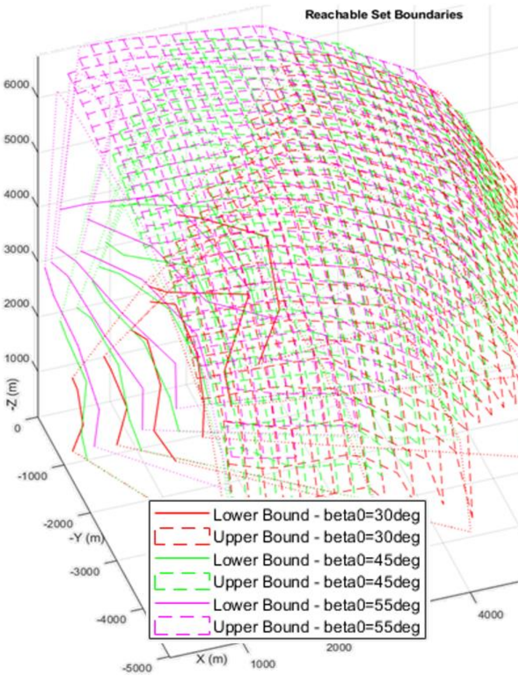


Figure 89: Visualization of Reachable Set Boundary for Different Initial Flight Path Angle

Figure 90 and Figure 91 demonstrates the reachability boundaries for different initial altitudes (test cases 1,7) and initial velocities (test cases 7,8). These figures demonstrate the reachability boundaries obtained for different search angle values ($\alpha_s = 0°, 15°, \cdots, 80°$ and $\beta_s = -25°, -10°, 10°, 25°, 30°$). It is evident from these figures that the reachability boundaries shift and expand with higher initial atitude and velocity. Furthermore, it is essential to emphasize that both the area and shape of the reachable set are subject to substantial influence by variations intial altitude and velocity.

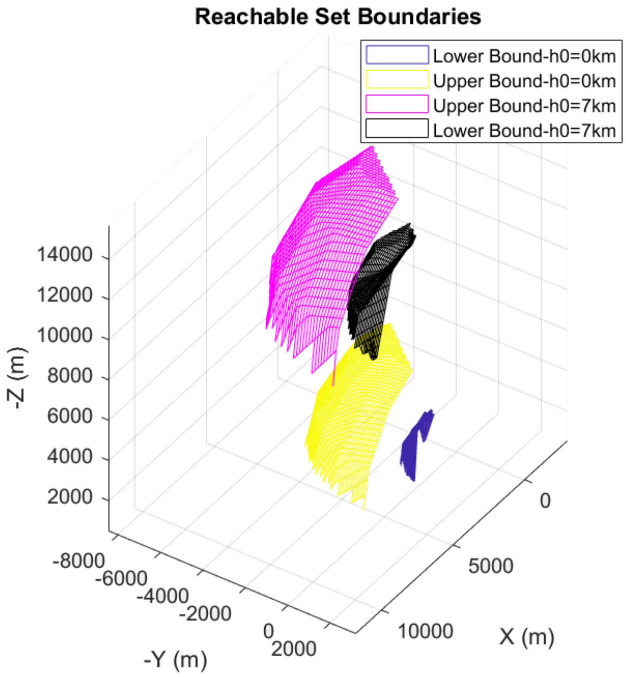


Figure 90: Visualization of Reachable Set Boundary for Different Initial Altitudes

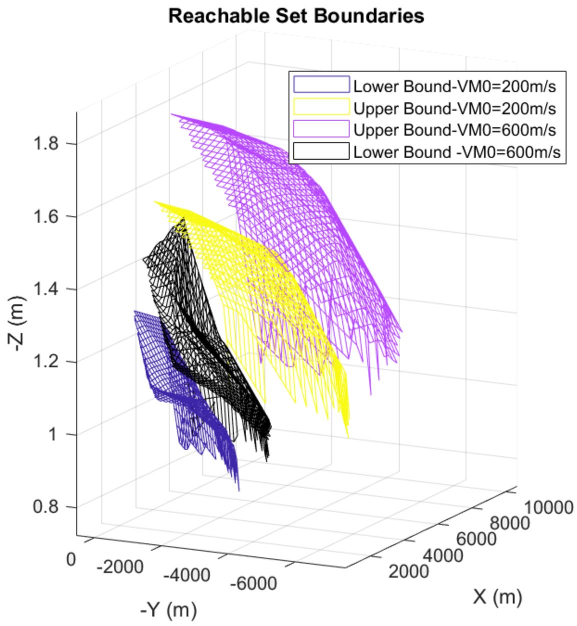


Figure 91: Visualization of Reachable Set Boundary for Different Initial Speeds

6.4 Reachable Set Comparison Between Unconstrained Input Case and Input Constraint Case

In this section, a comparison is made between test cases 1 and 5 (Table 2) to investigate how the presence or absence of input constraints affects the reachability boundaries. The reachability boundaries are for various search angles, specifically $\alpha_s = 0°, 15° \dots 90°$ and $\beta_s = -90°, -75°, \dots 75°, 90°$.

For test case 1, Figure 92 llustrates the reachability set boundary. The red line represents the reachable boundary acquired for the constrained input case with different α_s values , while the the blue line depicts the reachability boundaries in the scenario without input constraints. The figures highlight that the minimum range reachability boundaries contract when considering the acceleration limits. Furthermore, the difference between the constrained and unconstrained input cases in terms of minimum range reachability boundaries can reach up to 3500 meters for these specific test cases. These results emphasize the necessity of considering the acceleration capability of the interceptor to obtain a realistic reachability boundary.

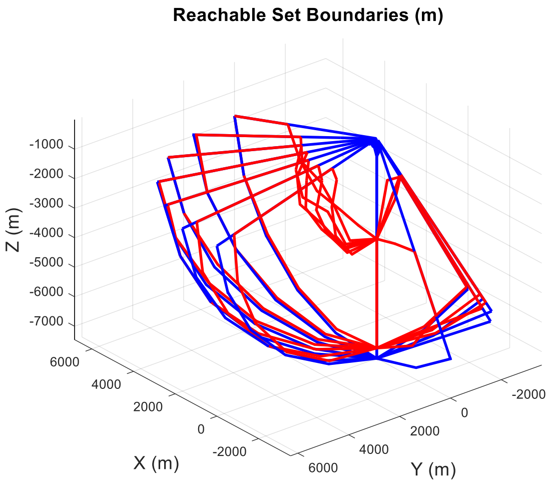


Figure 92: Visualization of Reachable Set For Test Case 1 and 5

Subsequently, the reachability set boundary for test case 1 is examined individually for each α_s value in Figure 93 to Figure 95. These figures reveal that, in addition to the shrinkage in the minimum reachability boundary, the maximum reachability boundary narrows, particularly at higher β_s angles, as expected. The interceptor requires greater acceleration commands to reach the desired terminal position at higher β_s values. Consequently, the maximum reachability boundary obtained through the input-constrained MPP approach differs from the boundary obtained through the unconstrained input case due to the activation of acceleration limits at higher β_s values.

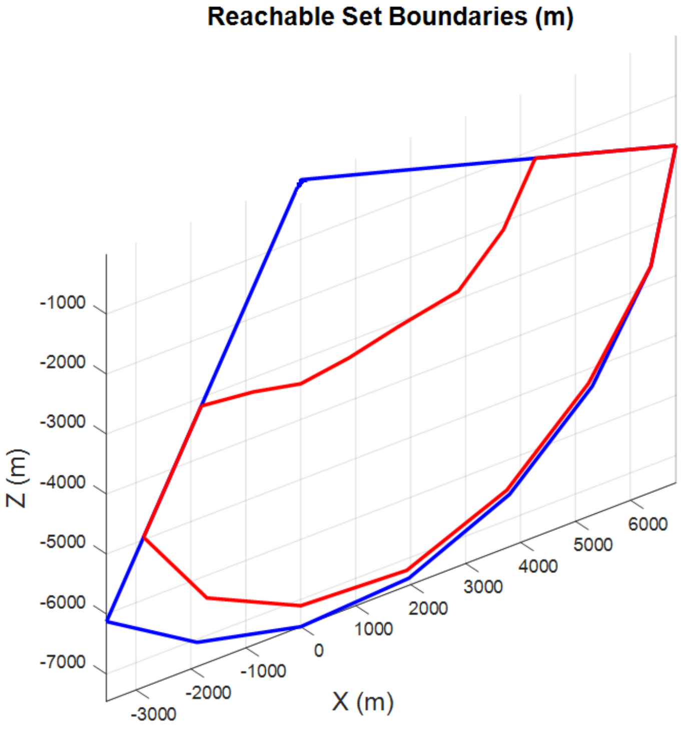


Figure 93: Visualization of Reachable Set Boundary For $\alpha_{\mathrm{search}} = 0°$

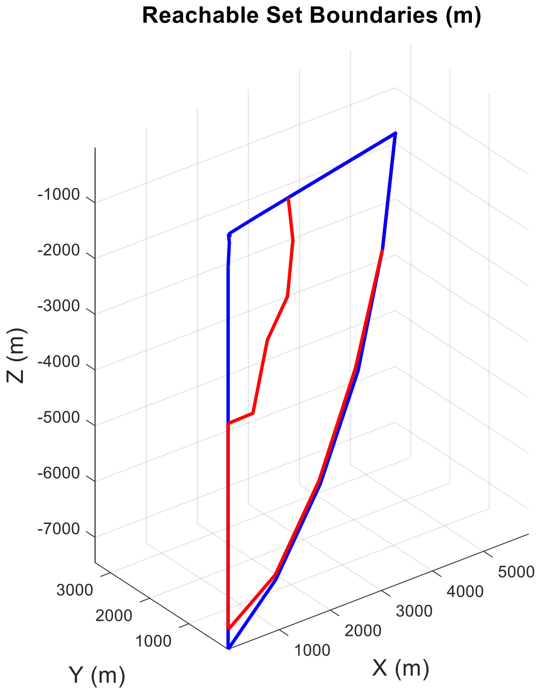


Figure 94: Visualization of Reachable Set Boundary For $\alpha_{\mathrm{search}} = 30°$

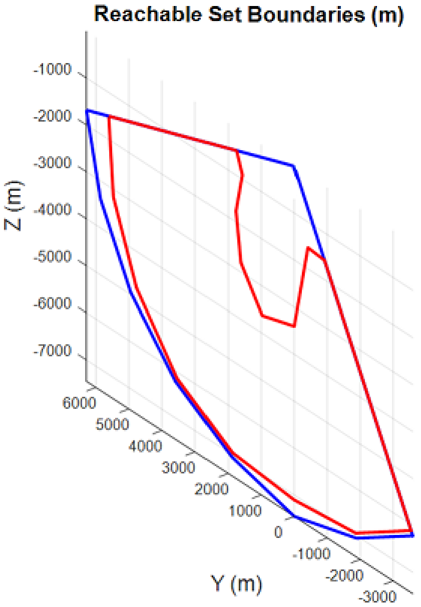


Figure 95: Visualization of Reachable Set Boundary For $\alpha_{\mathrm{search}} = 90°$

Figure 96 presents the reachability boundaries for test case 5, where the red line represents the reachability boundaries acquired for the constrained input case with different α_s values, and the blue line represents the reachability boundaries for the unconstrained input case. The figures illustrate that both the minimum and maximum range reachability boundaries exhibit significant discrepancies when considering the acceleration limits. The disparity between the constrained and unconstrained input cases concerning the minimum range reachability boundaries can reach up to 3500 meters. Additionally, for the maximum range reachability boundaries, the difference may escalate to 1000 meters, particularly for higher α_s values. These findings underscore the importance of considering input constraints to ensure a realistic calculation of the reachability boundary.

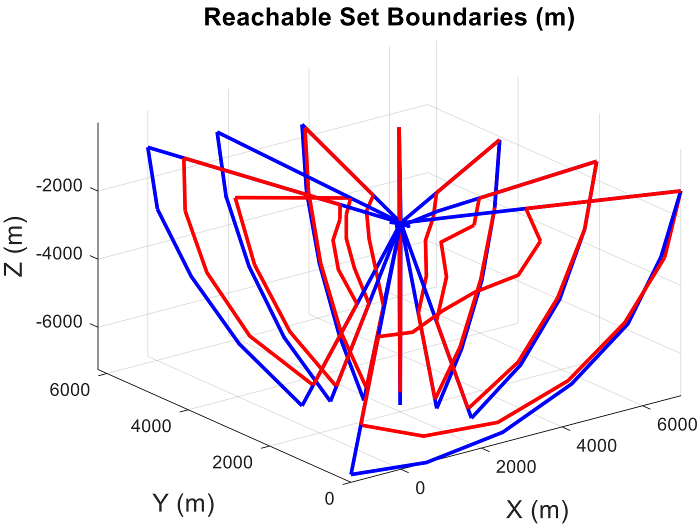


Figure 96: Visualization of Reachable Set Boundary for the Test Case 5

6.5 Discussions

The analysis of the constrained input cases in this study has revealed the significant influence of input constraints on the reachability boundaries. A comparison between the unconstrained and constrained input cases for test cases 1 and 5 (Table 2) clearly demonstrates the notable variations in both the minimum and maximum range reachability boundaries when considering acceleration limits. These findings underscore the critical importance of incorporating input constraints into the reachability boundary calculations to ensure accurate and realistic results.

The effect of parameter variation on the reachability set can also be discussed in relation to the presented results. By examining the variations in the initial conditions, such as the angle β_0 and initial altitude h_0, one can observe their impact on the reachability boundaries. These variations can influence the interceptor's kinematic capabilities, ultimately affecting the achievable flight ranges within the specified flight duration. Furthermore, the time parameter t_{go} also plays a crucial role in determining the reachability boundaries, as different flight durations yield distinct ranges that the interceptor can cover.

CHAPTER 7

SENSITIVITY ANALYSIS

In this chapter, a sensitivity analysis and reachability set computation are presented, considering parameter perturbations in an aerodynamic interceptor system. The objective is to explore and address the variations in system dynamics, which can potentially lead to substantial changes in the reachable set. By conducting simulations that consider different scenarios with variations in system parameters, it is aimed to generate robust reachable maps that can capture variations in the system. The main emphasis of this study is to understand how parameter variations influence the minimum and maximum boundaries of the reachable set, which are key indicators of the interceptor's kinematic capabilities. By tabulating the sensitivity of the reachable set with respect to system parameters, it might be possible to quantify the extent to which each parameter affects the reachable set, providing a understanding of the system's behavior under different operating conditions.

7.1 Perturbed Parameters

A sensitivity analysis was conducted to evaluate the impact of different parameters on the reachable set. The study involved a series of simulations with perturbations introduced to drag coefficients, thrust, and the autopilot time constant, as outlined in Table 5. These variations enable an examination of the impact of parameter changes on the characteristics of the reachable set.

Table 5: Different Perturbation Cases

Scenario Description	Perturbed Parameter	Scenario #	Perturbation Percentage
Nominal	-	1	-
Base Drag Coefficient Variation	$C_{X,0}$	2	-20%
		3	-10%
		4	+10%
		5	+20%
Acceleration Related Drag Coefficient Variation	$C_{X,a}$	6	-20%
		7	-10%
		8	+10%
		9	+20%
Pulse 1 Thrust Variation	T_{p1}	10	-20%
		11	-10%
		12	+10%
		13	+20%
Autopilot Time Constant Variation	τ_{AP}	14	-20%
		15	-10%
		16	+10%
		17	+20%

7.2 Investigation of Base Drag Coefficient and Acceleration Related Drag Coefficient Effects on Reachable Sets

In this section, the impact of base drag coefficient and acceleration related drag coefficient variations on the reachable sets of the aerodynamic interceptor is analyzed. Figure 97 and Figure 98 present the the variations in reachable sets resulting from different base drag and acceleration related drag values, respectively. The figures demonstrate that the maximum reachability boundary expands for specific scenarios denoted as Sc2 and Sc3 in Figure 97 and Sc6 and Sc 7 in Figure 98. This expansion is attributed to lower drag values, which reduce the drag experienced by the interceptor, thus enabling it to maintain higher energy levels.

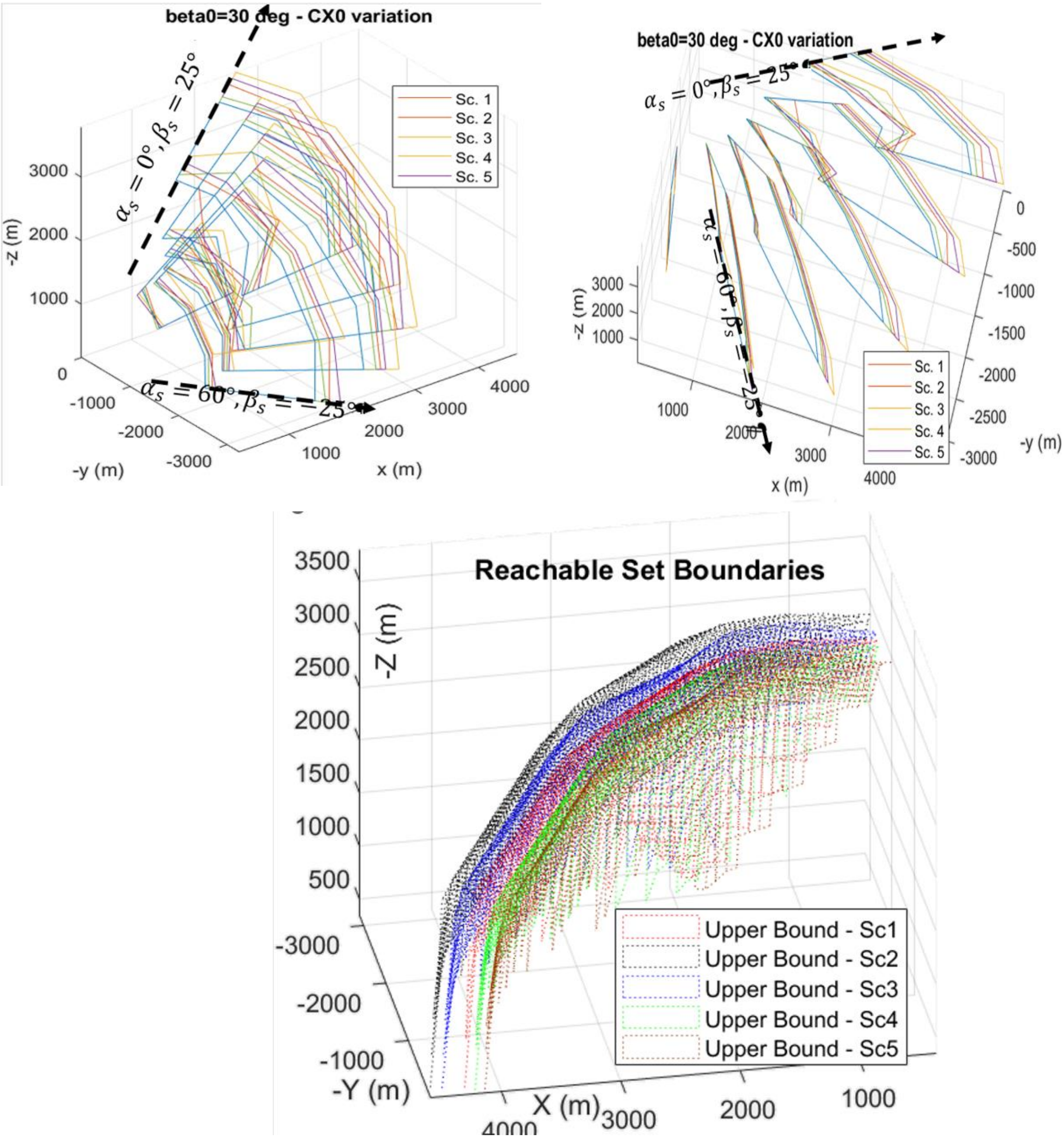


Figure 97: Effect of Base Drag Coefficient Variation on Reachability Boundaries

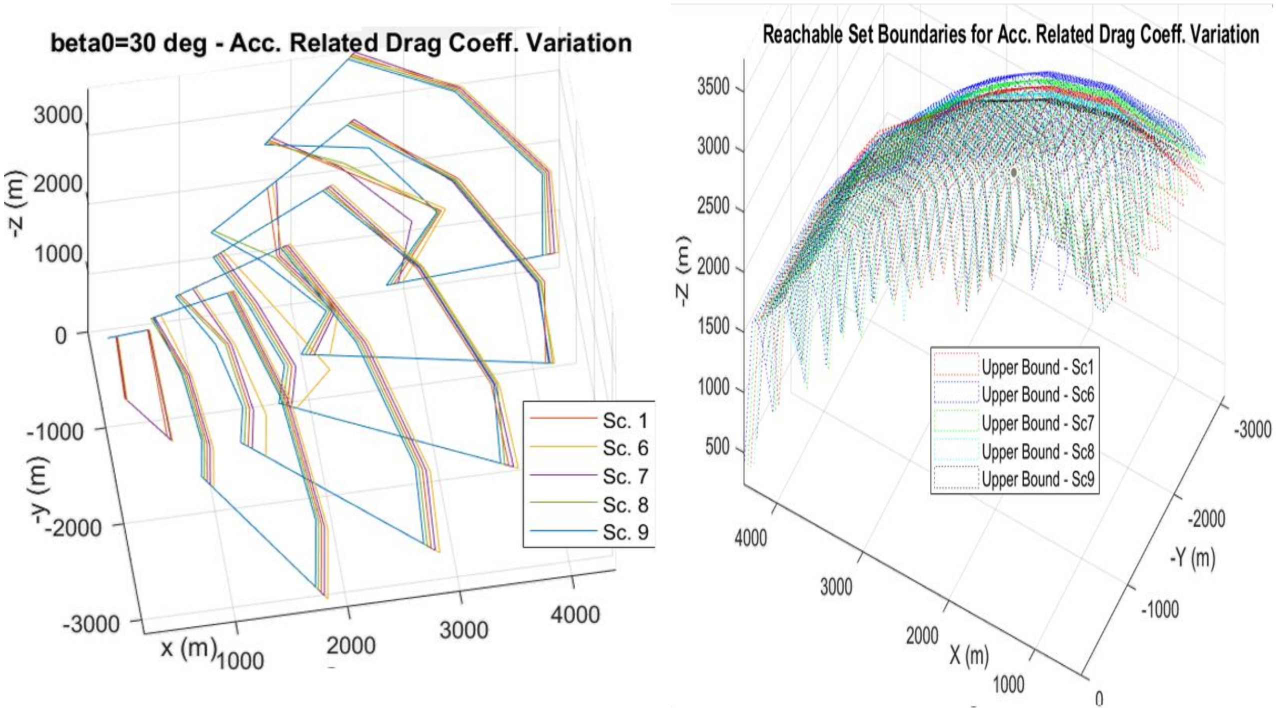


Figure 98: Effect of Acceleration Related Drag Coefficient Variation on Reachability Boundaries

To quantify the deviations from the nominal condition (Sc1), Table 6 and Table 7 present the mean percentage variations of the minimum and maximum boundary values, respectively. It is important to note that parameter variations significantly impact both minimum and maximum range reachability boundaries, with the maximum boundaries experiencing relatively smaller effects. This phenomenon can be attributed to the activation of inequality input constraints more frequently for the minimum range reachability boundaries. As a result, the system generates higher acceleration commands to reach closer terminal positions (with respect to the initial position) for the same flight duration. Furthermore, the study reveals that the minimum and maximum boundary variations are nearly identical in all directions of the inertial frame. This finding suggests that the minimum and maximum range reachability boundaries undergo uniform scaling in all directions, preserving their shapes while expanding or contracting the overall area of the reachability boundaries.

Table 6: Percentage of Minimum Boundary Variations for Drag Coefficient Perturbation Cases

Sc	x	y	z
2	6.58	6.58	6.58
3	5.10	5.10	5.10
4	5.02	5.02	5.02
5	6.49	6.49	6.49
6	6.49	6.49	6.49
7	2.59	2.59	2.59
8	3.95	3.95	3.95
9	5.81	5.81	5.81

Table 7: Percentage of Maximum Boundary Variations for Drag Coefficient Perturbation Cases

Sc	x	y	z
2	5.50	5.50	5.50
3	2.80	2.80	2.80
4	2.24	2.24	2.24
5	4.98	4.98	4.98
6	2.17	2.17	2.17
7	1.29	1.29	1.29
8	1.23	1.23	1.23
9	1.80	1.80	1.80

7.3 Effect of Thrust Variation

Figure 99 illustrates the reachable set boundaries for different thrust magnitudes, providing information about the interceptor's kinematic capabilities under varying propulsion conditions. The reachability boundaries are represented by the maximum and minimum limits within which the interceptor can maneuver while satisfying the defined constraints. Upon close examination, it becomes evident that the maximum reachability boundary expands significantly for Sc12 and Sc13 scenarios, where higher thrust magnitudes are applied. This observation aligns with the anticipated behavior, as higher thrust levels empower the interceptor to accumulate greater energy during flight, resulting in an extended reachability boundary. Consequently, the interceptor gains the ability to cover longer distances and reach more distant targets with increased thrust, thus enhancing its mission effectiveness in long-range engagements. The insights obtained from Figure 99 are crucial for optimizing the interceptor's performance and mission planning, as they provide valuable information on the impact of thrust variations on the achievable reachability boundaries. By considering different thrust scenarios, designers can better understand the trade-offs between energy consumption and range, enabling them to make informed decisions to enhance the interceptor's overall operational capabilities.

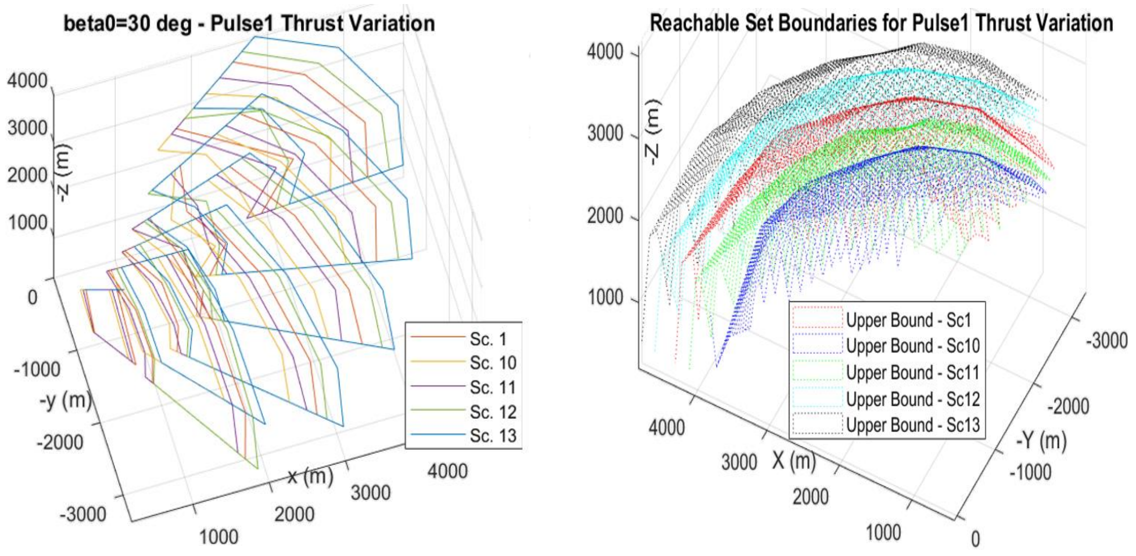


Figure 99: Effect of Thrust Variation on Reachability Boundaries

Table 8 and Table 9 shows the variations in boundary values expressed as percentages compared to the nominal condition (Sc1). The results reveal the sensitivity of the reachable set's minimum and maximum boundaries to parameter perturbations. The analysis demonstrates that parameter variations significantly affect both the minimum and maximum range boundaries of the reachable set. Each scenario (Sc10, Sc11, Sc12, and Sc13) is assessed, and the mean percentage deviations of the minimum boundary values in the x, y, and z directions are reported, shedding light on the system's response to different perturbations. Furthermore, another observation is that the influence of parameter variations is uniformly distributed across all directions of the inertial frame. This phenomenon implies that the minimum and maximum range reachability boundaries experience proportional adjustments in all directions, retaining their fundamental shapes while adapting to the changes in parameter values. As a result, the reachability boundaries exhibit consistent enlargement or contraction, determined by almost identical scale factors, ensuring the preservation of their overall shape during variations.

Table 8: Percentage of Minimum Boundary Variations for Thrust Perturbation Cases

Sc	x	y	z
10	6.88	6.88	6.88
11	6.08	6.08	6.08
12	6.90	6.90	6.90
13	7.84	7.84	7.84

Table 9: Percentage of Maximum Boundary Variations for Thrust Perturbation Cases

Sc	x	y	z
10	10.12	10.12	10.12
11	5.46	5.46	5.46
12	5.51	5.51	5.51
13	9.85	9.85	9.85

7.4 Effect of Autopilot Dynamics

In, Figure 100 , the impact of the autopilot time constant on the reachable set boundaries for different autopilot configurations is presented. It is observed that the effect of the autopilot time constant is relatively smaller compared to the influences of thrust and drag parameters. The system's responsiveness to changes in the autopilot time constant is highlighted, although its impact is not as pronounced as that of other critical parameters.

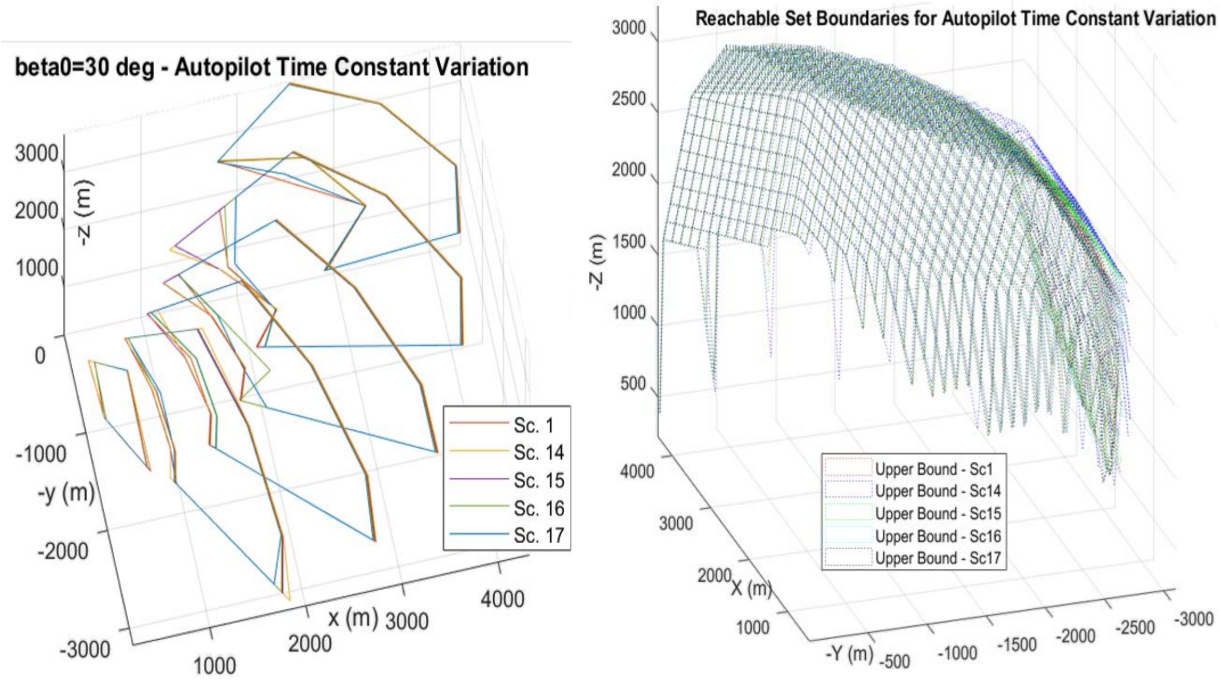


Figure 100: Effect of Autopilot Time Constant Variation on Reachability Boundaries

An analysis of the percentage variations of the minimum and maximum boundary values is provided in Table 10 and Table 11 , respectively, in comparison to the nominal condition (Sc1). Notably, the results reveal that variation in autopilot time constants has a limited impact on the maximum range reachability boundaries. However, its influence becomes more apparent when considering the minimum range reachability boundaries. This observation is attributed to the inherent difficulty in compensating for variations in the autopilot dynamics concerning maximum range reachability cases. The saturation of acceleration commands to limits occurs more frequently for the minimum range reachability boundaries, leading to a relatively higher sensitivity to autopilot time constant variations.

Table 10: Percentage of Minimum Boundary Variations for Different Autopilot Dynamic Cases

Sc	x	y	z
14	4.91	4.91	4.91
15	3.94	3.94	3.94
16	2.96	2.96	2.96
17	3.65	3.65	3.65

Table 11: Percentage of Maximum Boundary Variations for Different Autopilot Dynamic Cases

Sc	x	y	z
14	0.89	0.89	0.89
15	0.38	0.38	0.38
16	0.21	0.21	0.21
17	0.58	0.58	0.58

7.5 Effect of All Variations for a Specific Test Case

In Figure 101**,** a presentation is made of the yaw and pitch channel acceleration commands for specific conditions of $\alpha_s = 60°$ and $\beta_s = -25°$. The solid lines in the figure indicate the acceleration commands for a wide range of parameter variation scenarios (Sc1, Sc2, …, Sc16, Sc17). Black dashed lines in the figure demonstrate upper and lower limits of acceleration command.

The analysis reveals an observation, wherein the input constraints are consistently more active when dealing with the minimum range reachability boundary cases. Furthermore, for the specific test condition considered in this study the acceleration commands are found to be distributed within a distinct region around the nominal case. This behavior suggests that the interceptor's response to parameter variations remains confined to a certain range, signifying a certain level of robustness in its performance under these conditions.

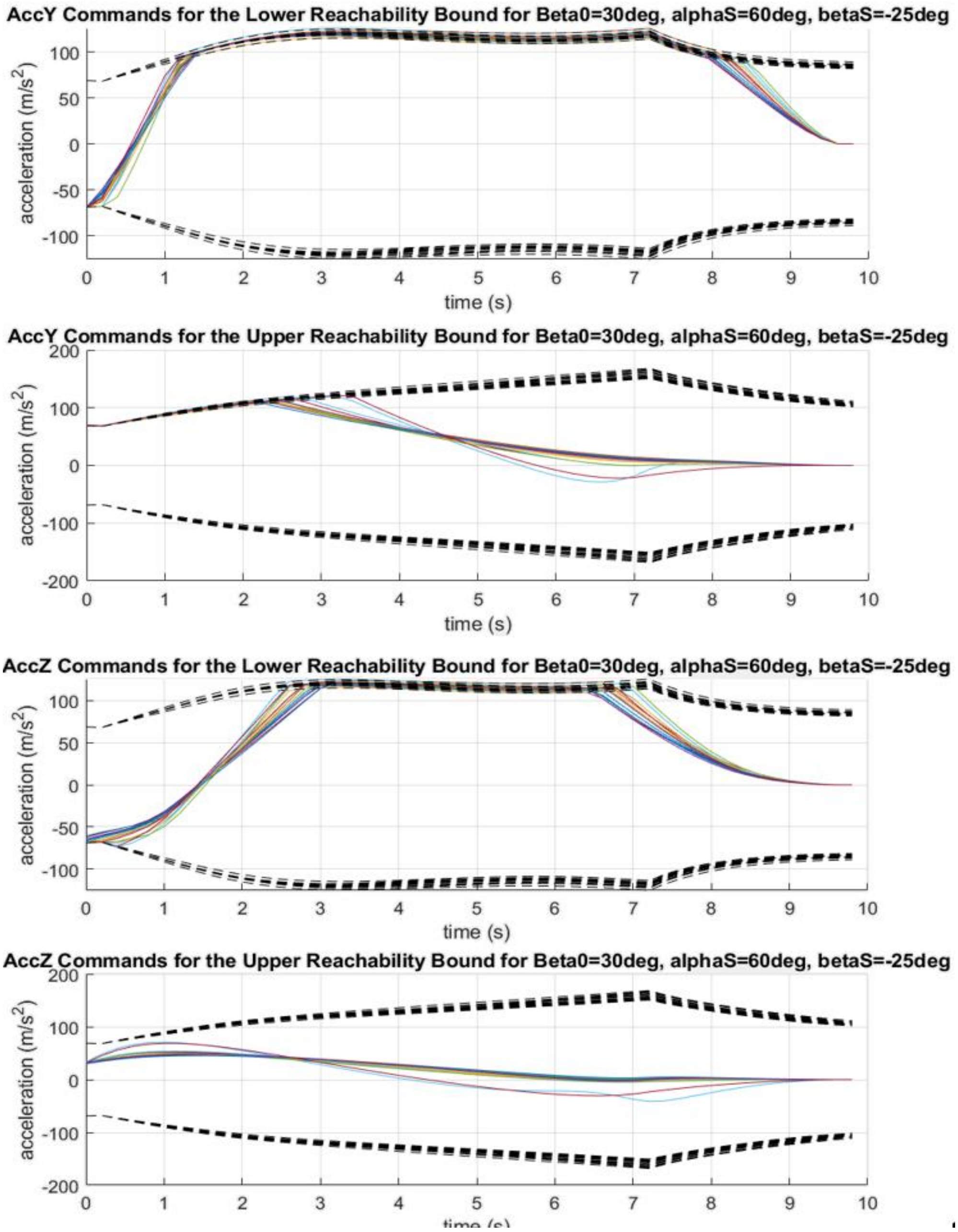


Figure 101:Acceleration Commands for Minimum and Maximum Range Reachability Boundaries

CHAPTER 8

CONCLUSION

In conclusion, this study has presented an analysis of the optimization-based method for guidance algorithm and the computation of reachability boundaries. The optimization problem in guidance algorithms has been addressed through the use of model predictive static programming, considering both equality and inequality constraints.

The optimization problem in guidance algorithms has been extensively studied in the literature. However, many existing approaches rely on simplifications and assumptions in the kinematic equations and models of guided objects to make the problem analytically solvable. While these simplifications may allow for mathematical tractability, they deviate from accurately representing the real system, making it challenging to assess the optimality of the obtained solutions. The common simplifications and assumptions include assuming constant velocity or rate of velocity increase, neglecting the effects of maneuvers on velocity, assuming unlimited maneuverability, ideal autopilot dynamics, neglecting flight mechanics, not modeling atmospheric conditions, and using approximate kinematic equations for small angles. The suggested approach overcomes the limitations of existing methods that rely on simplifications and assumptions, providing a more accurate and realistic modeling of the system.

To address the optimization problem with input constraints, model predictive static programming has been employed, specifically for cases involving inequality constraints on the inputs. However, directly solving for all Lagrange multipliers simultaneously poses challenges due to high-dimensional matrix operations and the requirement for the multipliers to be either zero or positive values. Therefore, a recursive solution based on Hildreth's procedure is utilized to determine active inequality bounds and their corresponding Lagrange multipliers. In cases where the input constraints prevent the satisfaction of the output equality constraint, convergence to a solution may not be possible regardless of the cost function. Moreover, when both the input inequality bounds and the output equality constraint are satisfied within a limited solution space, convergence may take a considerable amount of time. To address these issues, an upper limit on the number of iteration steps is imposed in addition to convergence criteria.

The applicability of the approach described in this thesis depends on various factors, including the complexity of the system model, the solution time step and flight duration, the width of the solution space for optimization, and the update frequency of the solution during the flight. These factors affect the computational requirements and the ability to converge to a solution within acceptable time limits. Additionally, it is important to note that the proposed approach does not guarantee finding the globally optimal solution across the entire solution space. Instead, it may converge to local optimal solutions. Therefore, this approach should be considered as a method that provides solutions in the “best proximity” class.

The reachable set algorithm developed in this study leverages the model predictive static programming technique and effectively incorporates input constraints. By employing a directional search approach, the algorithm computes the minimum and maximum boundary points of the reachable set, eliminating the need for predefined geometries or grid point positions.

The results obtained from the reachability analysis highlight the significance of considering input constraints. Comparisons between reachability sets computed with and without input constraints demonstrate the impact of acceleration limits on the shape and area of the reachability boundaries. Moreover, it is shown that the area of the RS was increasing with the increasing final time. Furthermore, the initial angle,altitude and velocity of the interceptor is shown to influence the shape of the reachability boundary.

It should be noted that practical implementation of the proposed method can be facilitated by storing the computed reachability boundaries in a guidance computer with an embedded database. This allows for online applications and reduces the computational load for real-time scenarios. The method's computational feasibility depends on factors such as sampling time, prediction horizon, dimensions of matrices, and the number of constraints. Efforts to systematically identify and eliminate inactive constraints can further enhance computational efficiency.

In order to investigate the sensitivity of the results, a sensitivity analysis was conducted by introducing variations in the initial states and system parameters. To generate feasible input vectors, an input-constrained model predictive static programming approach was employed. Incremental changes were then applied to the system parameters. Specifically, the value of each parameter was modified by multiplication factors of 0.8, 0.9, 1.1, and 1.2. The aim was to observe the effect of these variations on the reachable set's area and shape. By comparing the reachable sets obtained for each modified parameter value, the sensitivity of the results is evaluated.The analysis revealed that the area of the reachable set was most sensitive to variations in the thrust and base drag coefficient. These parameters had a significant impact on the overall size and extent of the reachable set.

The proposed method offers advantages such as the elimination of cumbersome set operations and the avoidance of approximation errors. By directly incorporating input constraints within the optimization process, the method ensures realistic and accurate reachability boundary calculations. Additionally, the flexibility of the approach allows for the inclusion of control and state constraints, as well as boundary conditions, making it applicable to a wide range of systems.

Despite its effectiveness, the proposed method does not guarantee finding the globally optimal solution. Instead, it converges to local optimal solutions within the solution space. Moreover, while the proposed method is effective in finding nonconvex reachable sets along search directions, it does have a limitation in that it does not identify unreachable zones along those search directions. The search process is terminated once an unreachable point is observed, and further points along the search direction are not investigated for reachability. This limitation arises from the termination criterion employed in the method, which prioritizes identifying reachable points rather than exploring the entire search direction. As a result, the algorithm may not provide information about regions that are completely unreachable along a particular search direction.

In conclusion, the findings of this study have provided information about the performance and limitations of the developed algorithm. Through the examination of various scenarios, the results of the algorithm were demonstrated.

REFERENCES

[1] Shih-Ming Yang, "Analysis of optimal midcourse guidance law," in IEEE Transactions on Aerospace and Electronic Systems, vol. 32, no. 1, pp. 419-425, Jan. 1996, doi: 10.1109/7.481282.

[2] Lee, J.-I., Jeon, I.-S., and Tahk, M.-J., “Guidance law to control impact time and angle,” IEEE Transactions on Aerospace and Electronic Systems, vol. 43, 2007, pp. 301–310.

[3] Kirk, D. E., Optimal control theory: an introduction, Englewood Cliffs (New Jersey): Prentice-Hall, 1970.

[4] Dwivedi, P. N., Bhattacharya, A., and Padhi, R., “Suboptimal Midcourse Guidance of Interceptors for High-Speed Targets with Alignment Angle Constraint,” Journal of Guidance, Control, and Dynamics, vol. 34, 2011, pp. 860–877.

[5] Robb, M., White, B., Tsourdos, A., and Rulloda, D., “Reachability guidance: a novel concept to improve mid-course guidance,” Proceedings of the 2005, American Control Conference, 2005.

[6] Girard, A., Guernic, C. L., and Maler, O., “Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs,” Hybrid Systems: Computation and Control Lecture Notes in Computer Science, 2006, pp. 257–271.

[7] Kurzhanskiy, A. A., and Varaiya, P., “Ellipsoidal Techniques for Reachability Analysis of Discrete-Time Linear Systems,” IEEE Transactions on Automatic Control, vol. 52, 2007, pp. 26–38.

[8] Varaiya, P., “Reach Set Computation Using Optimal Control,” Verification of Digital and Hybrid Systems, 2000, pp. 323–331.

[9] Greenstreet, M. R., and Mitchell, I., “Reachability Analysis Using Polygonal Projections,” Hybrid Systems: Computation and Control Lecture Notes in Computer Science, 1999, pp. 103–116.

[10] Hildreth, C., “A quadratic programming procedure,” Naval Research Logistics Quarterly, vol. 4, 1957, pp. 79–85.

[11] Shin, H. S., Tsourdos, A., Ménec, S., Markham, K., and White, B., “Cooperative Mid Course Guidance for Area Air Defence,” AIAA Guidance, Navigation, and Control Conference, 2010.

[12] Shin, H.-S., Ménec, S. L., Tsourdos, A., Markham, K., White, B., and Zbikowski, R., “Cooperative Guidance for Naval Area Defence,” IFAC Proceedings Volumes, vol. 43, 2010, pp. 124–129.

[13] Murray, R. M., Li, Z., & Sastry, S. S. (1993). A Mathematical introduction to robotic manipulation. Crc Press.

[14] Wang, L. (2010). Model predictive control system design and implementation using Matlab. Springer.

[15] M. Chen and C. J. Tomlin, “Hamilton-jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management,” The Annual Review of Control, Robotics, and Autonomous Systems, vol. 1, pp. 333–358, 2018.

[16] Bayoğlu, T., & Akalin, G. (2022). Reachability analysis of an aerodynamic interceptor with input constraint. AIAA SCITECH 2022 Forum. https://doi.org/10.2514/6.2022-2039.

[17] Dionne, D., Michalska, H., & Rabbath, C. A. (2006). A predictive guidance law with uncertain information about the target state. 2006 American Control Conference. https://doi.org/10.1109/acc.2006.1656357

[18] Dionne, D., & Rabbath, C. (2006). Predictive guidance for pursuit-evasion engagements involving decoys. AIAA Guidance, Navigation, and Control Conference and Exhibit. https://doi.org/10.2514/6.2006-6214.

[19] Robb, M., White, B., & Tsourdos, A. (2005). Earliest intercept line guidance: A novel concept for improving mid-course guidance in area air defence. AIAA Guidance, Navigation, and Control Conference and Exhibit. https://doi.org/10.2514/6.2005-5971.

[20] Shin, H. S., Tsourdos, A., White, B. A., & Tahk, M. J. (2009). Earliest intercept geometry guidance to improve mid-course guidance in area air-defence. 2009 17th Mediterranean Conference on Control and Automation. https://doi.org/10.1109/med.2009.5164769.

[21] In-Soo Jeon, Jin-Ik Lee, & Min-Jea Tahk. (2006). Impact-time-control guidance law for anti-ship missiles. IEEE Transactions on Control Systems Technology, 14(2), 260–266. https://doi.org/10.1109/tcst.2005.863655.

[22] In-Soo Jeon, Jin-Ik Lee, & Tahk, M.-J. (n.d.). Guidance law to control impact time and Angle. 2005 International Conference on Control and Automation. https://doi.org/10.1109/icca.2005.1528241.

[23] Shaferman, V., & Shima, T. (2015). Cooperative optimal guidance laws for imposing a relative intercept angle. Journal of Guidance, Control, and Dynamics, 38(8), 1395–1408. https://doi.org/10.2514/1.g000568.

[24] Balhance, N., Weiss, M., & Shima, T. (2017). Cooperative guidance law for Intrasalvo tracking. Journal of Guidance, Control, and Dynamics, 40(6), 1441–1456. https://doi.org/10.2514/1.g002250.

[25] Zhai, C., He, F., Hong, Y., Wang, L., & Yao, Y. (2016). Coverage-based interception algorithm of multiple interceptors against the target involving decoys. Journal of Guidance, Control, and Dynamics, 39(7), 1647–1653. https://doi.org/10.2514/1.g001535.

[26] Wang, J., He, F., Wang, L., & Yao, Y. (2014). Cooperative guidance for multiple interceptors based on dynamic target coverage theory. Proceeding of the 11th World Congress on Intelligent Control and Automation. https://doi.org/10.1109/wcica.2014.7053406.

[27] Chung, C. F. (2008). Reachable Sets Analysis in the Cooperative Control of Pursuer Vehicles. Retrieved from https://core.ac.uk/display/108110739

[28] Longbiao Ma, Fenghua He, Long Wang, Silun Zhang, Yu Yao, & Xin Huo. (2015). Asynchronous cooperative guidance for multiple intercepting flight vehicles: A dynamic target coverage approach. 2015 10th Asian Control Conference (ASCC). https://doi.org/10.1109/ascc.2015.7244848

[29] Enjiao, Z., Chao, T., Wang, S., & Yang, M. (2016). Distributed cooperative guidance law for multiple flight vehicles of saturation attack. AIAA Atmospheric Flight Mechanics Conference. https://doi.org/10.2514/6.2016-3240.

[30] Zhao, J., & Zhou, R. (2016). Distributed three-dimensional cooperative guidance via receding Horizon Control. Chinese Journal of Aeronautics, 29(4), 972–983. https://doi.org/10.1016/j.cja.2016.06.011.

[31] Ghosh, S., Ghose, D., & Raha, S. (2013). Three dimensional retro-PN based impact time control for higher speed nonmaneuvering targets. 52nd IEEE Conference on Decision and Control. https://doi.org/10.1109/cdc.2013.6760652.

[32] Daughtery, E., & Qu, Z. (2014). Optimal design of cooperative guidance law for simultaneous strike. 53rd IEEE Conference on Decision and Control. https://doi.org/10.1109/cdc.2014.7039510.

[33] Zhou, J., & Yang, J. (2016). Distributed guidance law design for cooperative simultaneous attacks with multiple missiles. Journal of Guidance, Control, and Dynamics, 39(10), 2439–2447. https://doi.org/10.2514/1.g001609.

[34] Saleem, A., & Ratnoo, A. (2015). A nonlinear guidance law for Impact Time Control. 2015 American Control Conference (ACC). https://doi.org/10.1109/acc.2015.7170809.

[35] Wei, X., Wang, Y., Dong, S., & Liu, L. (2015). A three-dimensional cooperative guidance law of Multimissile System. International Journal of Aerospace Engineering, 2015, 1–8. https://doi.org/10.1155/2015/479427.

[36] Prof. Dr. M.K. Özgören, "ME 502 Advanced Dynamics – Lecture Notes", Ortadoğu Teknik Üniversitesi, Makine Mühendisliği, 2008

[37] Arslantas, Y. E. (2017). Development of a reachability analysis algorithm for Space Applications (thesis).

[38] Chung, C. F. (2022, March 21). *Reachable sets analysis in the cooperative control of pursuer vehicles*. UNSWorks. https://unsworks.unsw.edu.au/entities/publication/cf05d223-d0d8-46bf-b5b8-0519057d2693

[39] Delos, V., & Teissandier, D. (2015). Minkowski sum of polytopes defined by their vertices. Journal of Applied Mathematics and Physics, 03(01), 62–67. https://doi.org/10.4236/jamp.2015.31008

[40] Mitchell, I., & Tomlin, C. J. (2000). Level set methods for computation in Hybrid Systems. Hybrid Systems: Computation and Control, 310–323. https://doi.org/10.1007/3-540-46430-1_27

[41] Baier, R., & Gerdts, M. (2009). A computational method for non-convex reachable sets using optimal control. 2009 European Control Conference (ECC). https://doi.org/10.23919/ecc.2009.7074386

[42] Baier, R., Gerdts, M., & Xausa, I. (2013). Approximation of reachable sets using optimal control algorithms. Numerical Algebra, Control and Optimization, 3(3), 519–548. https://doi.org/10.3934/naco.2013.3.519

[43] Gillula, J. H., Hoffmann, G. M., Haomiao Huang, Vitus, M. P., & Tomlin, C. J. (2011). Applications of hybrid reachability analysis to robotic aerial

vehicles. The International Journal of Robotics Research, 30(3), 335–354. https://doi.org/10.1177/0278364910387173

[44] Prandini, M., & Hu, J. (2008). Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction. 2008 47th IEEE Conference on Decision and Control. https://doi.org/10.1109/cdc.2008.4739248

[45] Lygeros, J., Tomlin, C., & Sastry, S. (1999). Controllers for reachability specifications for Hybrid Systems. Automatica, 35(3), 349–370. https://doi.org/10.1016/s0005-1098(98)00193-9

[46] Kitsios, I., & Lygeros, J. (n.d.). Aerodynamic envelope computation for safe landing of the HL-20 personnel launch vehicle using Hybrid Control. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005. https://doi.org/10.1109/.2005.1467020

[47] Isaacs,R. (1965). Differential Games: A Mathematical Theory with Application to Warfare and Pursuit, Control and Optimization, Dover Publications.

[48] Salmon, D.M., & Heine, W. (1973). Reachable Set Analysis-An Efficient Technique for Performing Missile/Sensor Tradeoff Studies. AIAA Journal, vol. 11,no.7,July,pp.927-931

[49] Leitmann. The Calculus of Variations and Optimal Control. New York: Springer, 1981. 17

[50] D. G. Hull. Optimal Control Theory for Applications. New York: Springer-Verlag, 2003. 17

[51] H. B. Keller. Numerical Solution of Two Point BouBound Value Problems. SIAM, 1976. 17

[52] 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

CURRICULUM VITAE

Bayoğlu Akalın, Tuğba

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Aerospace Engineering	2016
Minor	METU Electrical and Electronics Engineering	2014
BS	METU Aerospace Engineering	2014

EMPLOYMENT HISTORY

Position	Company	Employment Dates
Lead Flight Control Engineer	PLANA Aero	12.2022-Present
Senior Drone Algorithms Engineer	Ampyx Power	01.2022-05.2022
Senior Flight Mechanics Engineer	ROKETSAN	07.2014-09.2021

CERTIFICATIONS

1- AIAA Aircraft and Rotorcraft System Identification: Engineering Methods and Hands on Training Using CIFER certificate

2- EX-EN MSC. Nastran certificate

FOREIGN LANGUAGES

Fluent English, Elementary German

PUBLICATIONS

1. Bayoğlu, T., Nalci, O., & Kutay, A. T. (2016). “Aerodynamic Parameter Estimation of a Supersonic Air to Air Missile with Rapid Speed Variation”.

AIAA Atmospheric Flight Mechanics Conference. American Institute of Aeronautics and Astronautics. http://doi.org/doi:10.2514/6.2016-3856

2. Bayoğlu, T (2016). “Aerodynamic Parameter Estimation of a Supersonic Missile with Rapid Speed Variation by Using Kalman Filtering”. The Graduate School Of Natural And Applied Sciences of Middle East Technical University. Aerospace Engineering Department
3. Bayoğlu, T., Akalin, G. (2022). “Reachability Analysis of an Aerodynamic Interceptor with Input Constraint”. AIAA SciTech Guidance,Navigation and Control Conference. American Institute of Aeronautics and Astronautics SciTech. https://doi.org/10.2514/6.2022-2039
4. Bayoğlu, T., Akalin, G. (2023). “Impact of Parameter Variations on Reachability Sets of Aerodynamic Interceptors”. International Science and Innovation Congress.
5. Bayoğlu, T., Akalin, G., Kutay, A. T. (2023). “Development of Reachability Analysis Algorithm for an Aerodynamic Interceptor”. Journal of Guidance, Control, and Dynamics. [Manuscript submitted for publication]

Voluntary Work for Animal Care, Fantasy Books, Trekking