공학박사 학위논문

Optimal Trajectory Shaping and Vector Field-based Guidance for Missile System

유도탄의 최적궤적 성형 및 벡터필드 기반 유도

2021 년 2 월

서울대학교 대학원

기계항공공학부

이 수 원

Optimal Trajectory Shaping and Vector Field-based Guidance for Missile System

유도탄의 최적궤적 성형 및 벡터필드 기반 유도

지도교수 김 유 단

이 논문을 공학박사 학위논문으로 제출함

2020 년 12 월

서울대학교 대학원

기계항공공학부

이 수 원

이수원의 공학박사 학위논문을 인준함

2020 년 12 월

위 원 장 김 현 진 (인)

부위원장 김 유 단 (인)

위 원 기 창 돈 (인)

위 원 박 찬 국 (인)

위 원 정 동 원 (인)

Optimal Trajectory Shaping and Vector Field-based Guidance for Missile System

by

Suwon Lee

Submitted to the Graduate School of
Seoul National University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical and Aerospace Engineering

Seoul National University

Advisor: Prof. Youdan Kim

February 2021

Optimal Trajectory Shaping and Vector Field-based Guidance for Missile System

by

Suwon Lee

Approved as to style and content by:

Hjin Kim

H. Jin Kim, Chair, Ph.D.

Youdan Kim

Youdan Kim, Vice-Chair, Ph.D.

Changdon Kee, Member, Ph.D.

ChanGook Park

Chan Gook Park, Member, Ph.D.

Jung Dongwon

Dongwon Jung, Member, Ph.D.

Abstract

Optimal Trajectory Shaping and Vector Field-based Guidance for Missile System

Suwon Lee
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

In this dissertation, a novel framework is proposed that combines the optimal trajectory shaping and a stable guidance algorithm for missile systems. First, three-dimensional missile dynamic equations of motion are analyzed to divide the time-scale into two. The separation of time-scale offers easiness in designing and analyzing the reference trajectory. The separated time-scale consists of the dual-loop system of the missile's three-dimensional kinematic system. In the inner-loop, which is the fast-scale system, a tracking controller is designed. The inner-loop dynamics is much faster than the outer-loop dynamics at the time-scale, and the error state of the inner-loop system converges to zero quickly. Therefore, the state variables of the inner-loop system can be considered as the control input variables for the outer-loop system. In the outer-loop, which is the slow-scale system, the reference trajectory is designed. The optimal reference trajectory is attained through the output shaping technique proposed in this dissertation, which implements a parameterized curve into a trajectory optimization problem.

The second part of the framework is about optimal output trajectory. The nonlinear single-input-single-output system is considered for the design of the output trajectory shaping. The shape of the trajectory is manipulated by the Bézier curve, a particular category of the polynomial curves. Because of the characteristics of the Bézier curve, the Bézier curve is useful for shaping the trajectory of the dynamic system. The last part of the framework is about vector field-based guidance. Vector field-based guidance laws are designed for two- and three-dimensional path-following guidance. The reference path can be represented in an implicit function form or a parameterized curve form. In this dissertation, both types of representations of the reference path are covered with the vector field-based guidance algorithm. The guidance algorithm mainly focuses on the convergence characteristics of the reference path. Vector field designs with three different time-varying gains are also addressed, and the convergent characteristics of each of the different vector fields are analyzed.

The effectiveness of the proposed framework is demonstrated through numerical simulations. It is possible to utilize the proposed methodologies, including optimal output trajectory shaping technique and the vector field-based guidance algorithms, not only for the missile systems but also for various nonlinear systems.

Keywords: Time-scale separation, Trajectory optimization, Vector field-based guidance, Nonlinear control, Asymptotic stability, Finite-time convergence

Student Number: 2015-20785

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Because of the growing utility and interests, three-dimensional path-planning, guidance, and control algorithms have attracted much attention from industrial, military, and academic fields. The enumerated technologies can be utilized for various flight vehicles including missiles, Unmanned Aerial Vehicles(UAVs), and drones. Especially, in the military field, missile system is one of the most critical weapon systems for modern warfare. However, UAVs and drones are expanding their area of mission activities including surveillance, reconnaissance, and strategic utilization.

For a practical utilization of the aforementioned strategic assets, a proper design of the mission is essential. Figure 1.1 shows an example of system architecture for UAVs [1]. In the aspect of the system architecture, every three components are necessary, which are organized hierarchically.

Among the three of them, the path-planning, or trajectory-planning algorithm, takes the highest hierarchy of the entire mission management. The objective of the trajectory-planning is to obtain inputs to a nonlinear dynamic system that moves the system from the initial state to a specified goal state. It is widely known that the shortest path connecting two points is the Dubins

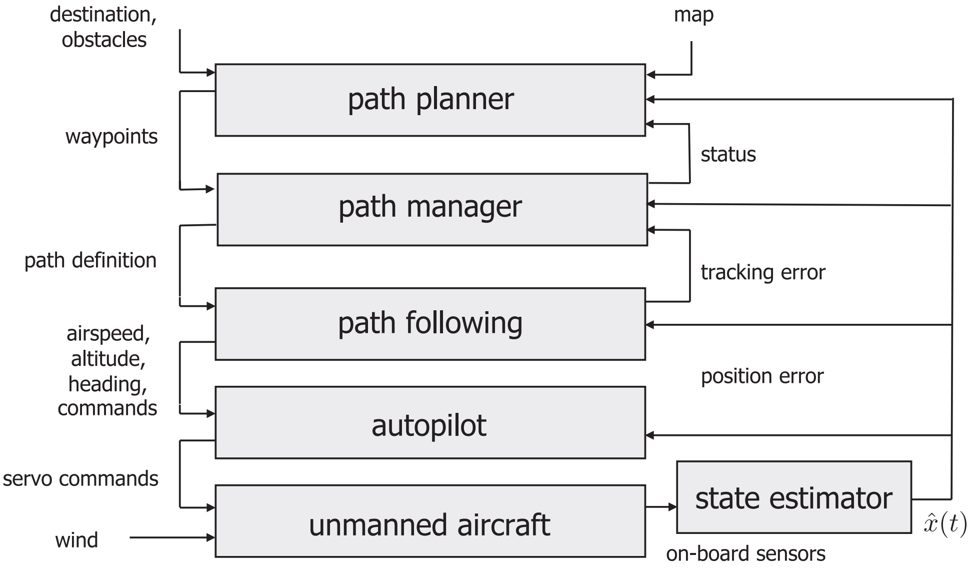


Figure 1.1: An example of system architecture for UAVs [1]

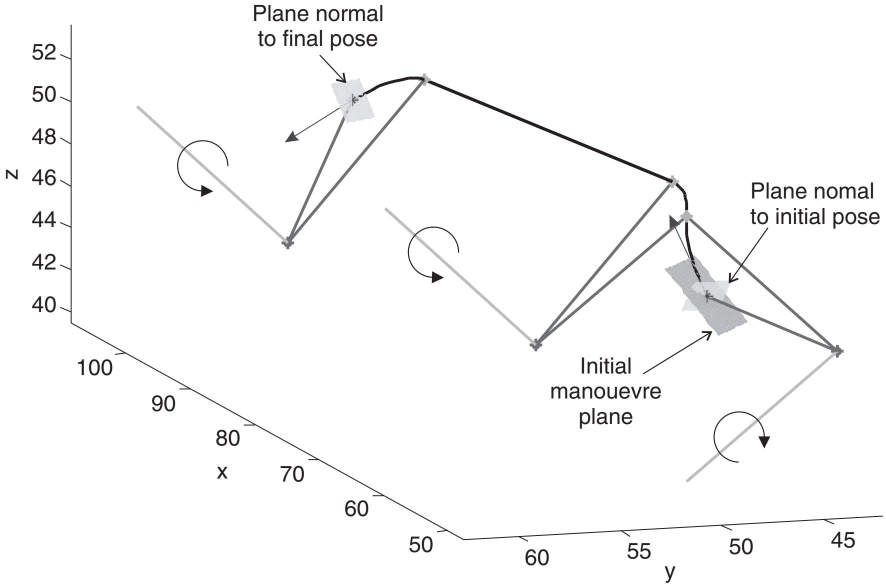


Figure 1.2: An example of path-planning using three-dimensional Dubins maneuver of a UAV [2]

path [3, 4]. Geometric approach is one of typical methods to define a reference trajectory planning. In [5], Pythagorean hodograph was used for the trajectory plannig of UAV group. Figure 1.2 shows an example of path-planning using three-dimensional Dubins maneuver of a UAV [2]. Recently, trajectory-planning considers additional aspects such as uncertainties, differential constraints, modeling errors, and optimality [6]. For example, in [7] and [8], path-planning for a glider vehicle was studied.

After the reference path is generated by the path-planner, the next hierarchy is the guidance algorithm. The guidance algorithm generates the control command that enables the vehicle to follow the reference path or to intercept a specified point in the space. Usually, the guidance algorithms are classified into the path-following and the homing guidance algorithms. The homing guidance algorithms have been extensively studied in the missile guidance field [9, 10]. The path-following guidance algorithms have been mainly studied for, but not limited to, UAVs [11–13].

The last part of the mission hierarchy is the control design including autopilot and stability augmentation system. The autopilot for aircraft [14] and the Thrust Vector Control (TVC) for missiles [15] are typical examples for the controller. The hierarchy should be systematically well-composed and designed to attain the mission's success.

The objective of this study is to develop a practical framework for the system architecture, including the path-planning and guidance algorithm. To this end, the nonlinear three-dimensional missile kinematic system is considered. The framework proposed in this study can be utilized for the mission planning of flight vehicles in consideration of the differential constraints and optimality.

1.2 Literature Review

The literature review presented in this section is categorized into three parts that is closely related to this study. The first category considers the preliminary for the system model: singular perturbation and timescale separation. The second category considers the trajectory tracking algorithms. And the third category is about the guidance algorithms.

1.2.1 Singular Perturbation and Timescale Separation

The dimension of the dynamic system can be reduced by singular perturbation method, and therefore, the singular perturbation method is useful in that the controllers can be desinged in the simplified low-dimensional subsystem model, other than in the original high-dimensional nonlinear system [16]. In the late 1970's, forced singular perturbation technique(FSPT) was propsed for the model reduction and decentralized controller for linear system [17, 18]. The designed controller based on FSPT shows better performance compared to the conventional modal trucation technique. In [19], the singular perturbation method was utilized for variational problems including optimal control problem of aircraft. Khalil studied a two-step method to stabilize multiparameter singularly perturbed system [20].

In 1980's and early 1990's, the singular perturbation method has been is widely utilized for the modeling of the dynamic system having multiple timescales. A forced singular perturbation model was obtained by artifically inserting the perturbation parameter into the dynamic constraints of the fast-timescale state variables [21, 22]. By taking the perturbation parameter as zero, zero-order approximation of the original system was obtained. However, the reduced model with the zero-order approximation does not satisfy the end condition imposed on the fast variables. Therefore, an additional boundary layer correction was introduced [23, 24]. There may exist multiple boundary layers if multiple timescales are introduced for the problem. Typically, the FSPT was applied to the missile guidance algorithm for the deveopment of the sub-optimal solution, which replaces the necessity of solving the time-consuming two-point boundary value problem(TPBVP) [25–27].

In late 1990's, the systematic means of identifying the time-scale structure in a nonlinear system was introduced by using regional Lyapunov exponents [28], which focused on the behavior of the solutions to the variational equations along the trajectories of a nonlinear systems to identify the timescale structure of the nonlinear system. The Lyapunov exponents and their associated direction vectors were used to identify the timescale structure of the nonlinear system. Timescale analysis for nonlinear dynamics systems were further conducted in early 2000's [29–31]. Singular perturbation theory has been utilized in various field and for various platforms. Shapira et al. studied singular perturbation analysis of glide vehicle [32]. In the field of missile guidance, Dhananjay et al. analyzed and compared various guidance laws based on timescale gap defined from the Lyapunov exponents [33], and nonlinear missile autopilot was designed using time-scale separation [34]. Surveys on singular perturbation theory in aerospace applications can be found in [35, 36]. Singular perturbations and time scales in control theory and applications can also be found in [37].

1.2.2 Trajectory Tracking Algorithms

There exist various methods to address the trajectory tracking algorithms. First, trajectory optimization can be performed to obtain the optimal control input history [38]. In the trajectory optimization, a trajectory minimizing (or maximizing) specified measure of performance is designed, which satisfies some constraints including the boundary conditions. The optimum trajectory can be obtained using a direct method or an indirect method [39]. The obtained solution is the optimal trajectory with respect to a given cost function, and therefore the optimal trajectory is often used as a baseline solution to compare with other approaches. However, the trajectory optimization techniques generally require a good initial guess and high computational efforts.

Another approach is to utilize trajectory tracking algorithms [1, 11, 40], which usually consist of two stages: i) constructing a reference trajectory satisfying the given constraints, and ii) designing a control algorithm that makes the vehicle track the reference trajectory. The trajectory tracking algorithm does not require iterative computation. However, it is not easy to impose all constraints on the reference trajectory. For example, position and velocity constraints are easily imposed on the reference trajectory in three-dimensional space, but it is difficult to consider attitude constraints in general. To consider the attitude constraint, the reference trajectory should be designed in six-dimensional space, which makes the process of constructing a reference trajectory much more difficult. The feasibility of the constructed trajectory also needs to be investigated, because not every trajectory is followable under a given dynamic constraint. Therefore, a well-tailored trajectory is required for the trajectory tracking algorithm, and additional optimization should be performed if the problem has a specific cost function to be minimized. One possible approach for constructing a feasible and reasonable reference trajectory is to utilize the solution of the trajectory optimization problem [15].

An approach based on guidance algorithms may also be used to solve the trajectory tracking problem, especially for some specific systems and constraints. For example, in missile guidance systems, the kinematic features of the missile-target engagement geometry have been widely utilized, and various guidance laws have been developed to intercept the target while satisfying some final conditions. For this specific system, the impact angle or impact time constraints are often considered in the target interception problems. The impact angle or impact time guidance laws have been designed via trajectory optimization [41,42] or trajectory tracking algorithms [10]. However, these approaches cannot be used in general systems because the guidance-algorithm-based approaches can be used only for specific systems such as missile systems.

On the other hand, parametric curves are often utilized to generate a reference path or trajectory for guidance or trajectory tracking. In [43] and [44], the Pythagorean-Hodograph (PH) curve and Bézier curve were utilized in path-planning for unmanned or autonomous vehicles. These curves can be modified to satisfy a given curvature bound. Faigl and Vana studied surveillance planning using Bézier curves [45]. Paths of different shape but identical arc length can also be obtained by using PH curves. On the other hand, Bernstein polynomials and Bézier curves can be used to approximate functions via numerical optimization techniques. In [46], Bernstein polynomials were used to approximate the optimal state/input histories as well as the costates using the indirect optimization method. In [47], the direct finite elements in time (DEFT) method was used for the direct transcription of optimal control problems with a Bernstein basis. In [48], the time interval was divided into subintervals, and trajectory and control functions in each subinterval were approximated by Bézier curves to solve the constrained quadratic optimal control of time-varying linear systems.

The Bézier function was also used to estimate spacecraft trajectories to fit a set of measured positions [49]. In these approaches, the Bernstein polynomials were used as basis functions for connecting specific nodes, dividing the entire trajectory. The Bernstein polynomials can be used as the basis for function approximation, which has merit in that the approximants and their derivatives converge uniformly to the corresponding functions and their derivatives. In [50], Pythagorean-Hodograph Bézier curves were used to generate trajectories for a team of cooperating vehicles with consideration of the spatial and temporal constraints. In [51], the Bernstein approximation was utilized in motion planning for differentially flat systems. However, these results are limited to the motion planning and do not consider the guidance and control algorithms.

1.2.3 Guidance Algorithms

The proportional navigation guidance (PNG) law is one of the widely used homing guidance laws for missile systems because of its simplicity in realization and its guaranteed performance [52–56]. Guidance laws based on line-of-sight (LOS) have been extensively studied [57]. Also, sliding mode guidance was developed considering the nonsingularity against maneuvering targets [58] and finite time-convergent characteristics [59]. However, these guidance laws have some shortcomings including lack of full use of predicted target trajectory and difficulty in analysis. On the other hand, differential geometry-based guidance laws have been developed [60]. In this approach, the target trajectory was modeled as a geometric curve, and consequently, a small miss distance could be achieved and the performance of the guidance law could be easily analyzed. However, the differential geometry-based guidance heavily depends on the target information. Many guidance laws have been developed exploiting the characteristics of geometric shapes. For example, circular impact-time guidance was designed [61], and the inscribed angle for a circle was used for the design of guidance laws with geometric constraints [62–64]. Elliptic guidance law is another example of geometry-inspired guidance laws [65].

In the 2000s, the vector field-based guidance law was developed for the path following problem of the unmanned aerial vehicles [66, 67] which was not for a target interception problem of the missile. The stand-off target tracking problem of unmanned aerial vehicle is a typical utilization of the vector field-based guiance algorithms [68–72]. Olavo et al. focused on developing a robustness analysis framework with the target circulation problem using the vector field based guidance [73]. Oh et al. utilized vector field guidance for the standoff tracking of moving target groups [74]. Goncalves et al. studied the vector field-based guidance in n-dimensions and time-varying vector fields [75, 76]. For the time-varying vector fields, additional time-varying component of the vector field was considered [77]. Path following technique using the vector field-based guidance law often combines the following for path segments including line and circles [67, 78, 79]. The vector field for linear path following and circular path following were separately designed and switched to follow the given path and avoid obstacles [69,80,81]. Also, dynamic constraints and input saturation were considered when designing the vector field [82–84].

Note that the vector field guidance is mainly used for the target tracking missions of UAVs. The advantages of the vector field guidance law are that it can be easily implemented for various types of problems including the precise path following problem [11] and the missile-target interception problem [85].

1.3 Contributions

The contributions of this dissertation are three-fold. First, missile's three-dimensional kinematic equations of motion is contemplated in the time-scale aspect. Second, optimal trajectory planning algorithm is delivered. Third, vector field-based guidance algorithm is proposed and convergence analysis to the reference path is performed.

The entire system architecture for the missile system is described in Fig. 1.3. In Fig. 1.3, each part shows the contribution of this dissertation. The timescale separation technique reduces the dynamic equations of the missile system. The benefit of the system reduction is that simple control design and analysis are possible. The OOTS plays the role of the path-planner and generates the reference path for the path-following guidance algorithm. The vector field-based guidance algorithm is a path-following guidance algorithm. The guidance algorithm calculates the command signal for the missile autopilot so that the missile can follow the desired reference path.

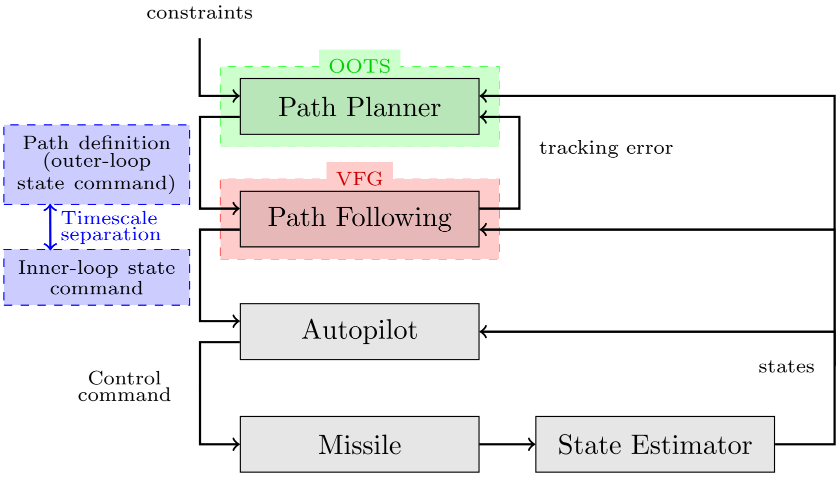


Figure 1.3: System architecture for the missile

1.3.1 Contribution 1: Time-scale separation of the missile system

Missile kinematics is divided into two and the divided systems organize the dual-loop system, which is often called as the time-scale separation. The time-scale separation divides the nonlinear dynamic system into the outer-loop (slow time-scale) and the inner-loop (fast time-scale) systems. The time-scale separation of the missile kinematics has several merits. First, the time-scale separation makes the planning and analysis of the dynamic system simple because the separation bears the effect of system order reduction. One can design control algorithms for each respective reduced-systems (inner- or outer-loop system) separately.

1.3.2 Contribution 2: Optimal output trajectory shaping algorithm

Optimal output trajectory shaping algorithm is developed to design the optimal trajectory of nonlinear systems. The nonlinear single-input-single-output system is considered. First, the output trajectory is shaped using parameterized curves, i.e., Bézier curves, and the input trajectory is shaped by manipulating the shape of the parameterized curves of the output trajectory. Next step is the parameterization of the state trajectory. The relative degree of the nonlinear system is essential for the parameterization of the state trajectory. The benefit of the proposed method is that it is easy to impose the boundary conditions and nonlinear constraints, because the trajectories of the system states and inputs can be manipulated by adequately designing the shape of the output trajectory. The shape of the trajectory can be optimized by parameter optimization algorithms including sequential quadratic programming. After the development of the parameterization and the optimization of the trajectories, a nonlinear feedback linearization controller is designed. The feedback linearization controller regulates the error between the designed trajectory and the system trajectory.

1.3.3 Contribution 3: Vector field-based guidance algorithm

Vector field-based guidance algorithm is designed for a two- and three-dimensional missile systems with constant speed. The vector field is given as the directional command for the missile in the specified space. Then, the missile is controlled to follow the vector field command, which guarantees convergence to a given reference curve. There exist two types of reference curves: 1) reference curve represented as an implicit function, and 2) reference curve represented as a parametric curve. In this study, both of the two types of reference curves are considered. The vector field command is designed and analyzed in the aspect of smooth and finite-time convergence to the reference curve.

1.4 Dissertation Overview

This dissertation is oranized as follows. In Chapter 2, the normal form and canonical form of a nonlinear system are addressed. The definition of the Bézier curve is also delivered. Finite-time stability theory and finite-time convergence guidance laws are addressed.

In Chapter 3, the time-scale separation of the missile's three-dimensional kinematics is mathematically explained.

In Chapter 4, the output trajectory shaping method for a nonlinear system is discussed. The trajectory shaping is attained using a parameterized curve, i.e., Bézier curve. The parameterizations of the state, input, and internal state are addressed. The trajectory optimization problem is solved for the parameterized trajectories of the system.

In Chapter 5, the VFG algorithm is designed for reference path represented as implicit function and parametric function. Three different types of gains for vector field design are proposed and analyzed. The characteristics of the convergence of the designed vector fields are discussed.

In Chapter 6, the performance of the proposed methodologies and design framework is demonstrated by numerical simulations.

In Chapter 7, the summary of the main results of this dissertation is summarized, and suggestions for future work are provided.

Chapter 2

Mathematical Preliminaries

2.1 Perturbation Theory

It is difficult to find a closed-form solution of a nonlinear ordinary differential equation, and therefore it is natrual to try to find a way to simplify the given equation. For this, the perturbation theory is often considered to compromise a mathematical method for finding an approximate solution to a problem, starting from the solution of a simpler problem. However, it is not always possble to obtain the regular perturbation model of the original problem, which can be uniformly approximated by the asymptotic expansions.

Definition 2.1. *Suppose $f(x,\varepsilon)$ and $\phi(x,\varepsilon)$ are continuous functions for $x \in I$ and $0 < \varepsilon < \varepsilon_1$. In this case, $\phi(x,\varepsilon)$ is a uniformly valid asymptotic approximation of $f(x,\varepsilon)$ for $x \in I$ if, given any positive constants δ, there is an ε_2 (independent of x and ε) such that,*

$$|f - \phi| \leq \delta|\phi| \quad \textit{for} \quad x \in I \quad \textit{and} \quad 0 < \varepsilon < \varepsilon_2 \tag{2.1}$$

where I designates an interval on the x-axis.

2.1.1 Singular Perturbation Model

The singular perturbation problem is a problem containing a small parameter that cannot be approximated by setting the parameter value to zero. For an introductory example, let us consider the following dynamical system, which is exploited from [86].

$$\varepsilon y'' + 2y' + 2y = 0, \quad \text{for } 0 < x < 1, \tag{2.2}$$

where

$$y(0) = 0 \ \text{ and } \ y(1) = 1. \tag{2.3}$$

Observe that if $\varepsilon = 0$, the problem is no longer second order, and the solution cannot be uniformly approximated by an asymptotic expansion. This leads to what is generally known as a singular perturbation problem. A singular perturbation generally, but not necessily, occurs when a problem's small parameter multiplies its highest operator.

In contrast to the regular perturbation problem, taking the parameter value to zero may changes the order of the system in the singular perturbation problem. This property of the singular perturbation model is exploted to derive an order-reduced system of dynamical systems. However, naively taking the parameter to zero changes the nature of the problem, which violates some boundary conditions. Therefore, a boundary layer matching process is necessary to derive the satisfactory solution of the problem. This method of approximating the solution is called the method of matched asymptotic expansions. More detailed explanations and examples can be found in [86, 87].

The singular perturbation model of the dynamic equations of motions is often utilized to analyze the timescale separation of the state variables, because the singularly perturbed problems are characterized by the dynamics operating on multiple scales.

2.1.2 Multiple Scales

Let us consider the following example.

$$y'' + \varepsilon y' + y = 0 \quad \text{for} \quad 0 < t \tag{2.4}$$

where

$$y(0) = 0 \quad \text{and} \quad y'(0) = 1 \tag{2.5}$$

The exact solution of the above problem is

$$y(t) = \frac{1}{\sqrt{1-\varepsilon^2/4}} e^{-\varepsilon t/2} \sin\left(t\sqrt{1-\varepsilon^2/4}\right). \tag{2.6}$$

The solution of the problem has an oscillatory component and a slow variation component. The regular power series expansion of the solution can be written as

$$\begin{aligned} y(t) &\sim y_0(t) + \varepsilon y_1(t) + \cdots \\ &\sim \sin(t) - \frac{1}{2}\varepsilon t \sin(t) \end{aligned} \tag{2.7}$$

The second term in the expansion is as significant as the first term when $\varepsilon t \simeq 1$ and grows unbounded, and therefore it can be argued that the regular approximation of the solution is valid only when $\varepsilon t \ll 1$. The multiple-scale approximation of the solution can alleviate this weakness.

Let us introduce the variable $t_1 = t$ and $t_2 = \varepsilon^q t$. These two time scales,

t_1 and t_2, will be treated as independent variables. Then, the original time derivative can be transformed as follows,

$$\frac{d}{dt} \rightarrow \frac{dt_1}{dt}\frac{\partial}{\partial t_1} + \frac{dt_2}{dt}\frac{\partial}{\partial t_2} = \frac{\partial}{\partial t_1} + \varepsilon^q \frac{\partial}{\partial t_2} \tag{2.8}$$

Substituting (2.8) into Eqs. (2.4) and (2.5) yields

$$(\partial_1^2 + 2\varepsilon^q \partial_1 + \varepsilon^{2q}\partial_2^2)y + \varepsilon(\partial_1 + \varepsilon^q \partial_2)y + y = 0 \tag{2.9}$$

where

$$y = 0 \quad \text{and} \quad (\partial_{t_1} + \varepsilon^q \partial_{t_2})y = 1 \quad \text{for} \quad t_1 = t_2 = 0. \tag{2.10}$$

To simplify the notation, let us use the symbols ∂_1 and ∂_2 in place of $\frac{\partial}{\partial t_1}$ and $\frac{\partial}{\partial t_2}$. Observe that the solution of Eqs. (2.9) and (2.10) is not unique and require additional conditions for the uniqueness of the solution.

Consider a power series expansion of the solution with respect to ε as follows,

$$y \sim y_0(t_1, t_2) + \varepsilon y_1(t_1, t_2) + \cdots . \tag{2.11}$$

Substituting the power series expansion into Eq. (2.9), we have

$$\begin{aligned} &(\partial_1^2 + 2\varepsilon^q \partial_1 \partial_2 + \varepsilon^{2q}\partial_2^2)(y_0 + \varepsilon y_1 + \cdots) \\ &\quad + \varepsilon(\partial_1 + \varepsilon^q \partial_2)(y_0 + \varepsilon y_1 + \cdots) + y_0 + \varepsilon y_1 + \cdots = 0. \end{aligned} \tag{2.12}$$

By equating the coefficients, it can be observed that $q = 1$. Finally, following problems are obtained.

Problem 1, $O(1)$:

$$(\partial_1^2 + 1)y_0 = 0, \quad y_0 = 0,\ \partial_1 y_0 = 1 \tag{2.13}$$

$$\text{when} \ \ t_1 = t_2 = 0 \tag{2.14}$$

Problem 2, $O(\varepsilon)$:

$$(\partial_1^2 + 1)y_1 = -2\partial_1\partial_2 y_0 - \partial_1 y_0, \quad y_1 = 0,\ \partial_1 y_1 = -\partial_2 y_0 \tag{2.15}$$

$$\text{when} \quad t_1 = t_2 = 0. \tag{2.16}$$

First, the general solution of the Problem 1 can be found as

$$y_0 = a_0(t_2)\sin(t_1) + b_0(t_2)\cos(t_1) \tag{2.17}$$

where

$$a_0(0) = 1 \quad \text{and} \quad b_0(0) = 0. \tag{2.18}$$

From the general solution of the Problem 1, the differential equation for y_1 is obtains as

$$(\partial_1^2 + 1)y_1 = (2b_0' + b_0)\sin(t_1) - (2a_0' + a_0)\cos(t_1) \tag{2.19}$$

Now, the general solution of the Problem 2 can be found as

$$\begin{aligned} y_1 = a_1(t_2)\sin(t_1) &+ b_1(t_2)\cos(t_1) \\ &- \frac{1}{2}(2b_0' + b_0)t_1\cos(t_1) - \frac{1}{2}(2a_0' + a_0)t_1\sin(t_1) \end{aligned} \tag{2.20}$$

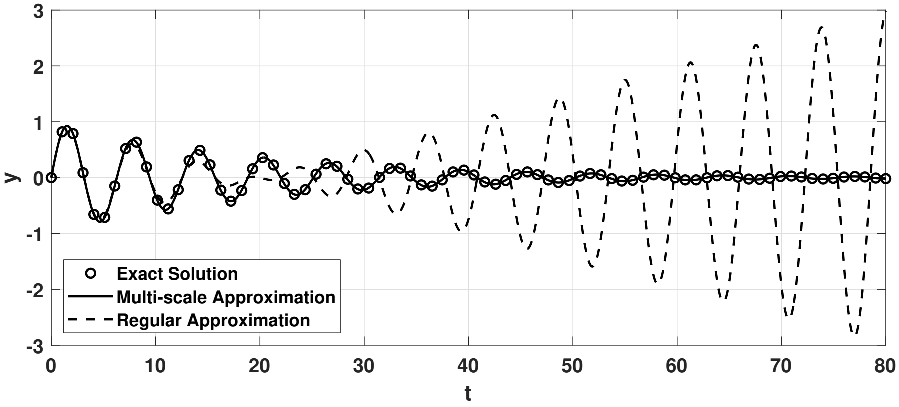


Figure 2.1: Comparison between the exact solution, the multi-scale approximation, and the regular approximation

where

$$a_1(0) = a_0'(0) \quad \text{and} \quad b_1(0) = 0. \tag{2.21}$$

The solution, Eq. (2.20), has the secular terms that grow without bound. However, $a_0(t_2)$ and $b_0(t_2)$ can be selected to satisfy the following constraints to remove the secular terms.

$$\begin{aligned} 2a_0' + a_0 &= 0 \\ 2b_0' + b_0 &= 0 \end{aligned} \tag{2.22}$$

From Eqs. (2.18) and (2.22), the coefficients $a_0(t_2)$ and $b_0(t_2)$ are found to be

$$a_0(t_2) = e^{-t_2/2} \tag{2.23}$$

$$b_0(t_2) = 0 \tag{2.24}$$

Finally, the first-term approximation of the solution is obtained as follows,

$$y \sim e^{-\varepsilon t/2} \sin(t) \tag{2.25}$$

This approximation is valid up to at least $\varepsilon t = O(1)$. The comparison between the exact solution, the multi-scale approximation, and the regular approximation are shown in Fig. 2.1. This approximation is a uniformly valid asymptotic approximation for $0 \leq t \leq O(\frac{1}{\varepsilon})$.

2.2 Normal Form and Canonical Form

The normal form of a nonlinear system will be used in this study to obtain a canonical form and design a tracking controller. In this section, the normal form of the nonlinear system and transformation to the canonical form are breifly introduced [88], [89].

2.2.1 Nonlinear SISO systems

Consider a following n-dimensional, single-input-single-output (SISO) system.

$$\dot{x} = f(x) + g(x)u, \quad y = h(x) \tag{2.26}$$

where f, g and h are suffciently smooth in a domain $D \subset R^n$. Using the Lie Derivative of h with respect to f, the derivative $\dot{y}$ can be represented as

$$\dot{y} = \frac{\partial h}{\partial x}[f(x) + g(x)u] \triangleq L_f h(x) + L_g h(x)u \tag{2.27}$$

Assume that $h(x)$ satisfies the following equation for an integer ρ (relative degree).

$$L_g L_f^{i-1} h(x) = 0, \quad \text{for } i = 1, 2, \cdots, \rho - 1, \text{ and } L_g L_f^{\rho-1} h(x) \neq 0 \tag{2.28}$$

Then, the control input u appears in the equation of $y^{(\rho)}$ with a nonzero coefficient as follows,

$$y^{(\rho)} = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u \tag{2.29}$$

If a system has relative degree ρ, then its input-output map can be converted into a chain of integrators $y^{(\rho)} = v$ by the following state feedback control input.

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} \Big(- L_f^{\rho} h(x) + v \Big) \tag{2.30}$$

In order to find a transformation $z = T(x)$ such that the new state z can be partitioned into a ρ-dimensional vector ξ and $(n - \rho)$-dimensional vector η, let us define ξ as follows,

$$\xi \triangleq \begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(\rho-1)} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{\rho-1} h(x) \end{bmatrix}. \tag{2.31}$$

If $\rho = n$, the change of variables is given by

$$z = T(x) = \operatorname{col}\Big(h(x), L_f h(x), \cdots, L_f^{n-1} h(x) \Big) \tag{2.32}$$

On the other hand, if $\rho < n$, the change of variables can be taken as

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho}(x) \\ --- \\ h(x) \\ \vdots \\ L_f^{\rho-1}h(x) \end{bmatrix} \triangleq \begin{bmatrix} \phi(x) \\ --- \\ \mathcal{H}(x) \end{bmatrix} \triangleq \begin{bmatrix} \eta \\ --- \\ \xi \end{bmatrix} \tag{2.33}$$

where ϕ_1 to $\phi_{n-\rho}$ are chosen such that $T(x)$ is a diffeomorphism and

$$\frac{\partial \phi_i}{\partial x} g(x) = 0, \quad \text{for } 1 \leq i \leq n - \rho. \tag{2.34}$$

It can be stated that the transformation $T(x)$ is diffeomorphism for any ρ by Theorem 2.1 [88].

Theorem 2.1 (Existence of a Diffeomorphism [88])**.** *If $\rho = n$, a neighborhood N of x_0 exists such that the map $T(x)$ of Eq. (2.32), is a diffeomorphism on N. If $\rho < n$, then a neighborhood N of x_0 and continuously differentiable functions $\phi_1(x), \cdots, \phi_{n-\rho}(x)$ exist such that Eq. (2.34) is satisfied for all $x \in N$ and the map $T(x)$ of Eq. (2.33) is a diffeomorphism on N.*

When $\rho < n$, the change of variables in Eq. (2.33) transforms Eq. (2.26) into

$$\begin{aligned} \dot{\eta} &= f_0(\eta, \xi) \\ \dot{\xi} &= A_c\xi + B_c\Big(L_f^{\rho}h(x) + L_g L_f^{\rho-1}h(x)u\Big) \\ y &= C_c\xi \end{aligned} \tag{2.35}$$

where $\xi \in \mathbb{R}^{\rho}$, $\eta \in \mathbb{R}^{n-\rho}$, (A_c, B_c, C_c) is a canonical form representation of a chain of ρ integrators,

$$A_c = \begin{bmatrix} 0_{\rho-1,1} & I_{\rho} \\ 0 & 0_{1,\rho-1} \end{bmatrix}, \quad B_c = \begin{bmatrix} 0_{\rho-1,1} \\ 1 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0_{1,\rho-1} \end{bmatrix} \tag{2.36}$$

and

$$f_0(\eta, \xi) = \frac{\partial \phi}{\partial x} f(x)\Big|_{x=T^{-1}(z)} \tag{2.37}$$

The vector η includes the internal state variables, and the corresponding dynamics is called the internal dynamics of the system.

In case $\rho = n$, the normal form reduces to

$$\dot{z} = A_c z + B_c\Big(L_f^n h(x) + L_g L_f^{n-1} h(x) u\Big), \quad y = C_c z \tag{2.38}$$

2.2.2 Multi-input-multi-output system

Consider a following multi-input-multi-output (MIMO) system.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}, \quad \mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{2.39}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the control input vector of the components u_i, $\mathbf{y} \in \mathbb{R}^m$ is the system output vector of the components y_i, $\mathbf{f}$ and $\mathbf{h}$ are smooth vector fields, and $\mathbf{G} \in \mathbb{R}^{n\times m}$ is a matrix whose columns are smooth vector field $\mathbf{g}_i$. Input-output linearization of MIMO systems is obtained similarly to the SISO case. Assume that ρ_i is the smallest integer that at least one of the inputs appears in $y_i^{(r_i)}$, then

$$y_i^{(r_i)} = L_{\mathbf{f}}^{r_i} h_i + \sum_{j=1}^{m} L_{\mathbf{g}_j} L_{\mathbf{f}}^{r_i - 1} h_i u_j \tag{2.40}$$

with $L_{\mathbf{g}_j} L_{\mathbf{f}}^{r_i - 1} h_i(\mathbf{x}) \neq 0$ for at least one j. For each output y_i, we have

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_{\mathbf{f}}^{r_1} h_1(\mathbf{x}) \\ \vdots \\ L_{\mathbf{f}}^{r_m} h_m(\mathbf{x}) \end{bmatrix} + \mathbf{E}(\mathbf{x})\mathbf{u} \tag{2.41}$$

The input-output linearization can be achieved only when the decoupling matrix $\mathbf{E}$ is invertible. The following input transformation provides m equations of the form $y_i^{(r_i)} = v_i$.

$$\mathbf{u} = \mathbf{E}^{-1} \begin{bmatrix} v_1 - L_{\mathbf{f}}^{r_1} h_1 \\ \vdots \\ v_m - L_{\mathbf{f}}^{r_m} h_m \end{bmatrix} \tag{2.42}$$

2.3 Bézier Curve

The output trajectory profile can be generated by any paramterized curves. In this study, for the simplicity and easiness in implementation, Bézier curve is adopted to generate the ouput trajectory.

The Bézier curve is a part of Bernstein polynomials restricted to the paramter interval $[0, 1]$. The Bernstein polynomials have a property that a continuous function on the interval $[0, 1]$ can be approximated uniformly [90]. The general representation of N-th order Bézier curve with cruve parameter τ is given as follows,

$$\mathbf{B}_N(\tau) = \sum_{v=0}^{N} b_{v,N}(\tau)\mathbf{P}_v, \quad \tau \in [0, 1] \tag{2.43}$$

where

$$b_{v,N}(\tau) = \binom{N}{v}\tau^v(1-\tau)^{N-v} \tag{2.44}$$

The polynomials $b_{v,N}$ and constant coefficients $\mathbf{P}_v$ determine the shape of the curve. Note that the polynomials $b_{v,N}(\tau)$ are the Bernstein polynomials of Nth degree, which form a complete basis over $[0, 1]$. The binomial coefficient, $\binom{N}{v}$, is defined as

$$\binom{N}{v} = \frac{N!}{v!(N-v)!} \tag{2.45}$$

The derivative of the N-th degree Bernstein polynomials are the polynomials of degree $N-1$, which is given by,

$$\frac{d}{d\tau}b_{v,N}(\tau) = N(b_{v-1,N-1}(\tau) - b_{v,N-1}(\tau)) \tag{2.46}$$

The i-th derivative of $b_{v,N}(\tau)$ is reiresented as follows,

$$\left(\frac{d}{d\tau}\right)^i b_{v,N}(\tau) = \frac{n!}{(n-i)!} \sum_{k=\max(0,v+i-N)}^{\min(v,i)} (-1)^{k+i} \binom{i}{k} b_{v-k,N-i}(\tau) \tag{2.47}$$

More detailed explanations on the derivatives of Bernstein polynoimals can be found in [91]. The coefficients $\mathbf{P}_v, v = 0, \cdots, N$, are called the *control points* of the Bézier curve. Note that the first coefficient, $\mathbf{P}_0$, and the last coefficient, $\mathbf{P}_N$, determine the initial ($\tau = 0$) point and final ($\tau = 1$) point of the curve, respectively. The polygon formed by connecting the control points with lines is called the Bézier polygon, and the convex hull of the Bézier polygon contains the Bézier curve.

2.4 Finite-time Stability Theory

Continuous finite-time-stabilizing feedback controllers have been developed in the 1990s. The continuous finite-time stabilization of the double integrator system was studied [92]. Bhat et.al. pointed out that the finite-time convergence implies the nonuniqueness of solutions, which is not possible in the presence of Lipschitz-continuous dynamics. A rigorous foundation for the finite-time stability theory for continuous autonomous systems was studied in [93]. In the early 2000s, finite-time control for nonlinear systems with specific structures has been studied. Finite-time stabilization and stabilizability of the small-time local controllable nonlinear system have been studied in [94]. To alleviate the limitations of the finite-time controllers which require the full state information, output feedback control law was developed for the finite-time control in the 2010s [95–97]. In [98], impact angle constraind guidance law with finite-time convergence was proposed using sliding mode technique. In [99], prescribed finite time stabilization technique was studied using time-varying feedback control. This technique can attain a uniformly prespecifiable convergence time, but may require large time-varying gain.

In this section, the mathematical background for the finite-time stability of continuous autonomous systems is introduced, which is mainly extracted from [92]. The finite-time stability theory will be utilized for the development of the guidance algorithm in Section V. Certainly, there exists some research works on guidance laws based on the finite-time control. The finite-time convergent guidance laws are also discussed in this section.

2.4.1 Finite-time Stability of Continuous Autonomous Systems

Let us consider the following nonlinear system.

$$\dot{y}(t) = f(y(t)) \tag{2.48}$$

where $f : \mathcal{D} \to \mathbb{R}^n$ is continous on an open neighborhood $\mathcal{D} \subseteq \mathbb{R}^n$of the origin $f(0) = 0$. The finite-time stability of a system is defined as follows.

Definition 2.2. *The origin is said to be a finite-time stable equilibrium of (*2.48*), if there exists an open neighborhood $\mathcal{N} \subseteq \mathcal{D}$ of the origin and a function $T_s : \mathcal{N}\backslash\{0\} \to (0, \infty)$, called the settling-time function, such that the following statements hold:*

- *Finite-time convergence : For every $x \in \mathcal{N}\backslash\{0\}$, the solution of the system ψ^x is defined on $[0, T_s(x))$, $\psi^x(t) \in \mathcal{N}\backslash\{0\}$ for all $t \in [0, T_s(x))$, and $\lim_{t\to T_s} \psi^x(t) = 0$.*

- *Lyapunov stability : For every open neighborhood $\mathcal{U}_\varepsilon$ of 0, there exists an open subset $\mathcal{U}_\delta$ of $\mathcal{N}$ containing 0 such that, for every $x \in \mathcal{U}_\delta\backslash\{0\}$, $\psi^x(t) \in \mathcal{U}_\varepsilon$ for all $t \in [0, T_s(x))$.*

The origin is said to be a globally finite-time stable equilibrium if it is a finite-time stable equilibrium with $\mathcal{D} = \mathcal{N} = \mathbb{R}^n$.

Suppose that the origin is a finite-time stable equilibrium of Eq. (2.48). Then, it has a unique solution for every initial condition in an open neighborhood of 0, including 0 itself.

Now, let us consider a motivating example for the derivation of the settling time function.

A Motivating Example

The above example is extracted from [93].

$$\dot{y}(t) = -\alpha \text{sign}(y(t))|y(t)|^{\lambda} \tag{2.49}$$

where $\text{sign}(0) = 0, \alpha > 0$, and $\lambda \in (0, 1)$. This system is continuous everywhere and locally Lipschitz everywhere exept the origin. The solution for Eq. (2.49) is obtained as follows,

$$y_{sol}(t,x) = \begin{cases} \text{sign}(x)[|x|^{1-\lambda} - \alpha(1-\lambda)t]^{\frac{1}{1-\lambda}}, & t < \dfrac{1}{\alpha(1-\lambda)}|x|^{1-\lambda}, \quad x \neq 0 \\ 0, & t \geq \dfrac{1}{\alpha(1-\lambda)}|x|^{1-\lambda}, \quad x \neq 0 \\ 0, & t \geq 0, \quad x = 0 \end{cases} \tag{2.50}$$

It can be readily seen that the settling-time function $T_s : \mathbb{R} \rightarrow \mathbb{R}_+$ satisfies the Definition 2.2.

$$T_s(x) = \frac{1}{\alpha(1-\lambda)}|x|^{1-\lambda} \tag{2.51}$$

Also, the Lyapunov stability of the system can be obtained by considering a following Lyapunov function.

$$V(x) = x^2 \tag{2.52}$$

Therefore, the origin of the system (2.49) is a globally finite-time stable equilibrium. Note that the settling-time function T_s is not Lipschitz continuous at the origin. Now, the finite-time stability theorem is given as follows.

Theorem 2.2 (Finite-time Stability [93]). *Suppose there exists a continuous function $V : \mathcal{D} \to \mathbb{R}$ such that the following conditions hold.*

- *V is positive definite.*
- *There exist real numbers $\alpha > 0$ and $\lambda \in (0, 1)$ and an open neighborhood $\mathcal{V} \subseteq \mathcal{D}$ of the origin such that,*

$$\dot{V}(x) + \alpha V(x)^{\lambda} \leq 0, \quad x \in \mathcal{V}\backslash\{0\}. \tag{2.53}$$

*Then, the origin is a finite-time stable equilibrum of (*2.48*). Also, if $\mathcal{N}$ is as in Definition 2.2 and T_s is the settling-time function, then*

$$T_s(x) \leq \frac{1}{\alpha(1-\lambda)} V(x)^{1-\lambda}, \quad x \in \mathcal{N} \tag{2.54}$$

and T_s is continuous on $\mathcal{N}$.

The proof of Theorem 2.2 can be found in [93]. The converse to this theorem is as follows.

Theorem 2.3 (Converse to the Finite-time Stability [93]). *Suppose the origin is a finite-time stable equilibrium of (*2.48*) and the settling-time function T_s is continuous at 0. Let $\mathcal{N}$ be as in Definition 2.2 and let $\alpha \in (0, 1)$. Then, there exists a continuous function $V : \mathcal{N} \to \mathbb{R}$ such that the following conditions are satisfied.*

- *V is positive definite.*
- *$\dot{V}$ is real valued and continuous on $\mathcal{N}$ and there exists $\lambda > 0$ such that*

$$\dot{V}(x) + \lambda(V(x))^{\alpha} \leq 0, \quad x \in \mathcal{N} \tag{2.55}$$

2.4.2 Finite-time Convergence Guidance Laws

As stated in the previous section, there exist various studies on the guidance laws with finite-time convergence. The finit-controll has robustness and the finite-time convergent property, which are advantageous aspects, compared to the conventional robust guidance laws including the H_∞ guidance, L_2 gain guidance, and Lyapunov-based nonlinear guidance laws. Practically, the homing guidance problem requires the interception against the target in a very short range of time to neutralize. However, the conventional nonlinear robust guidance algorithms without finite-time convergence will converge to zero or a small neighborhood of zero as the time approaches infinity. Therefore, it can be stated that the theoretical findings are inconsistent with practical observations.

The guidance algorithms based on the finite-time stability theorem have drawn attention from the late 2000s. A finite-time convergent guidance law was proposed, in which the line-of-sight (LOS) angular rate converges to zero in finite time [100, 101]. The structure of the finite-time convergent guidance law is similar to that of the conventional sliding-mode guidance laws. Sliding surface was designed for an impact-angle control guidance with the finite-time convergent property [102]. The settling-time was computed in the converging phase and the sliding phase, respectively, to obtain proper settling-time. The finite-time convergent guidance algorithms were further developed considering additional constraints, including the input saturation and three-dimensional geometry [59, 103].

Chapter 3

Time-scale Separation and Dual-loop System

3.1 Three-dimensional Kinematics

In this study, the 3-dimensional equations of motion (EOM) for a missile are considered as follows,

$$\dot{x}(t) = V_m \cos\gamma \cos\chi \tag{3.1}$$

$$\dot{y}(t) = V_m \cos\gamma \sin\chi \tag{3.2}$$

$$\dot{z}(t) = -V_m \sin\gamma \tag{3.3}$$

$$\dot{\gamma}(t) = \frac{a_v}{V_m} \tag{3.4}$$

$$\dot{\chi}(t) = \frac{a_h}{V_m \cos\gamma} \tag{3.5}$$

where x, y, z are the NED (North-East-Down) coordinate elements of the glider vehicle with the origin located at the final target point, and V_m, γ, χ are the speed, flight path angle, heading angle, respectively.

The coordinate system is described in Fig. 3.1. The missile is controlled by a_v and a_h, which are the accelerations towards the vertical and horizontal directions, respectively.

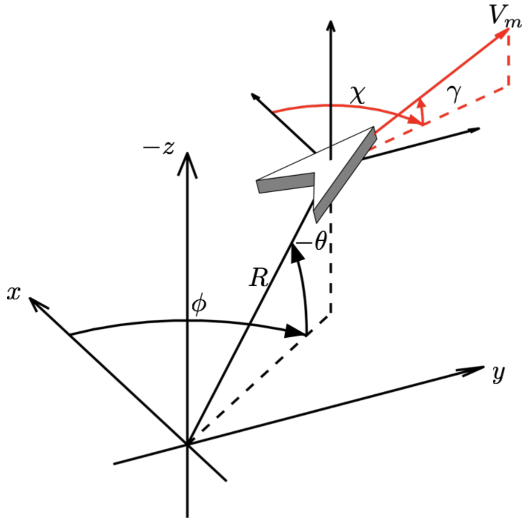


Figure 3.1: Coordinate system

The lateral accelerations are designed as follows,

$$a_v = -\frac{V_m}{\varepsilon}(\gamma - \gamma_d) \tag{3.6}$$

$$a_h = -\frac{V_m \cos\gamma}{\varepsilon}(\chi - \chi_d) \tag{3.7}$$

where γ_d and χ_d are the desired values of the flight path angle and heading angle, respectively, and $0 < \varepsilon \ll 1$. These values are given from the outer-loop controller. Then, the kinematics for the flight path angle and heading angle can be rewritten as follows,

$$\dot{\gamma}(t) = -\frac{1}{\varepsilon}(\gamma - \gamma_d) \tag{3.8}$$

$$\dot{\chi}(t) = -\frac{1}{\varepsilon}(\chi - \chi_d) \tag{3.9}$$

In this study, the aerodynamic model is neglected for simplicity.

3.2 Time-scale Separation

To separately design the inner-loop and outer-loop controllers, the time-scale separation between the two loops is considered using the singular perturbation method [86, 104]. Let us define a new time variable $\sigma = \varepsilon t$. Note that σ can be understood as a slow-scale time relative to time t because $\sigma \ll t$. The time differentiation with respect to t can be replaced by the differentiation with respect to σ using the relation $\frac{d}{d\sigma} = \varepsilon \frac{d}{dt}$. Then, the kinematic EOM, (3.1)-(3.3),(3.8), and (3.9) can be rewritten as follows,

$$\frac{d}{d\sigma}(x) = \varepsilon V_m \cos\gamma \cos\chi \tag{3.10}$$

$$\frac{d}{d\sigma}(y) = \varepsilon V_m \cos\gamma \sin\chi \tag{3.11}$$

$$\frac{d}{d\sigma}(z) = -\varepsilon V_m \sin\gamma \tag{3.12}$$

$$\frac{d}{d\sigma}(\gamma) = -(\gamma - \gamma_d) \tag{3.13}$$

$$\frac{d}{d\sigma}(\chi) = -(\chi - \chi_d) \tag{3.14}$$

The state variables can be classified into the outer-loop variables (x, y, z) and inner-loop state variables (γ, χ). The inner-loop variables can be understood as control variables in the outer-loop EOM.

The state history from the EOM has fast variations that occur on a time scale, that is $O(1)$, and it also has slow variations on a time scale $O(\frac{1}{\varepsilon})$ [86,105]. To incorporate two time scales into the problem, let us introduce two time variables as

$$t_1 = \sigma, \qquad t_2 = \varepsilon\sigma \tag{3.15}$$

These two time variables will be treated as independent variables. The time derivative with respect to σ can be transformed as follows,

$$\frac{d}{d\sigma} = \frac{dt_1}{d\sigma}\frac{\partial}{\partial t_1} + \frac{dt_2}{d\sigma}\frac{\partial}{\partial t_2} = \frac{\partial}{\partial t_1} + \varepsilon\frac{\partial}{\partial t_2} \tag{3.16}$$

To simplify the notation, let us use the symbols ∂_1 and ∂_2 in place of $\frac{\partial}{\partial t_1}$ and $\frac{\partial}{\partial t_2}$.

Note that the reciprocal of the parameter ε resembles a control gain that regulates the errors of γ and χ. In other words, a smaller ε regulates the error faster, and the difference in time scale becomes larger.

3.3 Outer-Loop State Variables

Substituting Eq. (3.16) into (3.10) yields

$$(\partial_1 + \varepsilon\partial_2)x = \varepsilon V_m \cos\gamma\cos\chi \tag{3.17}$$

The original ordinary differential equation (ODE) has been turned into a partial differential equation (PDE). Now, let us consider a power series expansion of the form,

$$x \sim x_0(t_1, t_2) + \varepsilon x_1(t_1, t_2) + \cdots \tag{3.18}$$

where $(\cdot)_0$ denotes a first-term approximation, and $(\cdot)_1$ denotes a second-term approximation, etc. Applying Eq. (3.18) into Eq. (3.17) yields

$$(\partial_1 + \varepsilon\partial_2)(x_0 + \varepsilon x_1 + \cdots) = \varepsilon V_m \cos\gamma\cos\chi \tag{3.19}$$

From Eq. (3.19), the kinematics in the fast time scale $O(1)$ and slow time scale $O(\frac{1}{\varepsilon})$ are obtained as

$$O(1): \qquad \partial_1 x_0 = 0 \tag{3.20}$$

$$O(1/\varepsilon): \qquad \partial_1 x_1 + \partial_2 x_0 = V_m \cos\gamma \cos\chi \tag{3.21}$$

Note from Eq. (3.20) that the first-term approximation of the state variable x, i.e., x_0, is constant, which is valid up to at least $\varepsilon t = O(1)$ [86]. Similarly, for the other outer-loop variables y and z, we have

$$\partial_1 y_0 = \partial_1 z_0 = 0 \tag{3.22}$$

In other words, the outer-loop state variables may be considered as constants in the fast time scale.

3.4 Inner-Loop State Variables

Now, let us consider the inner-loop variable γ. The power series expansion for γ can be written as follows,

$$\gamma = \gamma_0(t_1, t_2) + \varepsilon\gamma_1(t_1, t_2) + \cdots \tag{3.23}$$

Applying the power series expansion to Eq. (3.13), we have

$$(\partial_1 + \varepsilon\partial_2)(\gamma_0 + \varepsilon\gamma_1 + \cdots) = -(\gamma_0 + \varepsilon\gamma_1 + \cdots - \gamma_d) \tag{3.24}$$

From Eq. (3.24), the kinematics in the fast time scale $O(1)$ and slow time scale $O(\frac{1}{\varepsilon})$ are obtained as

$$O(1): \qquad \partial_1\gamma_0 = -(\gamma_0 - \gamma_d) \tag{3.25}$$

$$O(1/\varepsilon): \qquad \partial_1\gamma_1 + \partial_2\gamma_0 = -\gamma_1 \tag{3.26}$$

with the initial conditions of

$$\gamma_0(0,0) = \gamma_i, \quad \gamma_1(0,0) = 0 \tag{3.27}$$

where γ_i is the initial flight path angle of the vehicle. The kinematics in Eq. (3.25) and the initial condition in Eq. (3.27) are exactly identical to the original EOM in Eq (3.13). Therefore, there is only fast-scale dynamics in $O(1)$ and no slow-scale dynamics in Eq. (3.13).

Let us assume that γ_d is a constant. Then, the general solution of the problem in $O(1)$ can be represented as

$$\gamma_0 = a_0(t_2)e^{-t_1} + \gamma_d \tag{3.28}$$

where

$$a_0(0) = \gamma_i - \gamma_d \tag{3.29}$$

From Eq. (3.28), coefficient $a_0(t_2)$ is an arbitrary function of t_2, but it is required to satisfy the initial condition in Eq. (3.29). Differentiating Eq. (3.28) with respect to t_2 and substituting the resulting equation into Eq. (3.26), we obtain the PDE in the slow time scale as follows,

$$\partial_1 \gamma_1 = -a_0' e^{-t_1} - \gamma_1 = 0 \tag{3.30}$$

where $a_0' = da_0/dt_2$. There is no slow-scale dynamic equation, and therefore Eq. (3.30) is zero. Consequently, $a_0' = 0$, and the general solution for $\gamma = \gamma_0$ can be represented as follows,

$$\begin{aligned}\gamma &= (\gamma_i - \gamma_d)e^{-t_1} + \gamma_d \\ &= (\gamma_i - \gamma_d)e^{-t_2/\varepsilon} + \gamma_d \end{aligned} \tag{3.31}$$

Similarly, we have

$$\begin{aligned}\chi &= (\chi_i - \chi_d)e^{-t_1} + \chi_d \\ &= (\chi_i - \chi_d)e^{-t_2/\varepsilon} + \chi_d \end{aligned} \tag{3.32}$$

Equations (3.31) and (3.32) indicate that the inner-loop state variables exponentially converge to the desired values in the fast-time scale.

If variable ε becomes smaller, the converging rate becomes faster. Smaller ε can be understood as a larger control gain for the accelerations, which can also be understood as the wider difference between the fast-scale dynamics and the slow-scale dynamics. Therefore, in the slow-scale dynamics $O(1/\varepsilon)$, the inner-loop state variables γ and χ can be considered as the control variables because the inner-loop state variables converge to the desired values in the fast-scale dynamics.

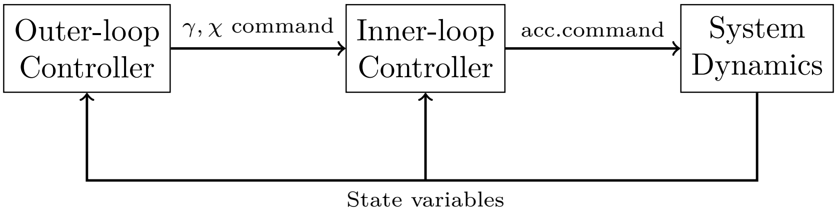


Figure 3.2: Block diagram of the dual-loop system

3.5 The Dual-Loop System

The given dynamic equations of motion is separated into two time-scales, i.e., the slow-scale system and the fast-scale system. In the slow-scale system, state variables are x, y, and z, and the inputs are γ and χ. And in the fast-scale system, state variables are γ and χ, and the inputs are a_v and a_h.

Figure 3.2 show the block diagram of the dual-loop system discussed in this section. In the slow-scale system, the γ and χ are considered to be input variables. Therefore, the role of the outer-loop controller is to generate proper input command for the slow-scale system, i.e., γ_d and χ_d. Meanwhile in the fast-scale system, the γ and χ are considered to be state variables and lateral accelerations are used as input variables. The inner-loop controller generates acceleration command for the fast-scale system to properly track the γ_d and χ_d.

3.6 A Motivating Example for Timescale Separation

In this section, the merit of utilizing the timescale separation method in designing a controller is discussed with a demonstration. Sliding model controller(SMC) is adopted to design a controller for a missile. First, a control design example using SMC without the timescale separation technique is presented.

3.6.1 Controller Design without Timescale Separation

Let us consider the dynamic equations of motion given in Eqs. (3.1)-(3.5). To design a SMC, the sliding surface $\underline{\sigma}$ is defined as follows,

$$\underline{\sigma} = [x - x_d, y - y_d, z - z_d, \lambda_1(\dot{x} - \dot{x}_d), \lambda_2(\dot{y} - \dot{y}_d), \lambda_3(\dot{z} - \dot{z}_d)]^T \tag{3.33}$$

where $(\cdot)_d$ is the desired values. It is assumed that the desired trajectory $[x_d(t), y_d(t), z_d(t)]$ and thetime derivatives are given as the reference trajectory. Let us consider the following Lyapunov function.

$$V = \frac{1}{2}\underline{\sigma}^T\underline{\sigma} \tag{3.34}$$

The time derivative of V is represented as follows,

$$\begin{aligned}
\dot{V} &= (x - x_d)(\dot{x} - \dot{x}_d) + (y - y_d)(\dot{y} - \dot{y}_d) + (z - z_d)(\dot{z} - \dot{z}_d) \\
&\quad + \lambda_1(\dot{x} - \dot{x}_d)(\ddot{x} - \ddot{x}_d) + \lambda_2(\dot{y} - \dot{y}_d)(\ddot{y} - \ddot{y}_d) + \lambda_3(\dot{z} - \dot{z}_d)(\ddot{z} - \ddot{z}_d) \\
&= V_m\Big((x - x_d)\cos\gamma\cos\chi + (y - y_d)\cos\gamma\sin\chi - (z - z_d)\sin\gamma\Big) \\
&\quad - \Big((x - x_d)\dot{x}_d + (y - y_d)\dot{y} + (z - z_d)\dot{z}\Big) \\
&\quad + \lambda_1(V_m\cos\gamma\cos\chi - \dot{x}_d)(-\sin\gamma\cos\chi a_v - \sin\chi a_h - \ddot{x}_d) \\
&\quad + \lambda_2(V_m\cos\gamma\sin\chi - \dot{y}_d)(-\sin\gamma\sin\chi a_v + \cos\chi a_h - \ddot{y}_d) \\
&\quad + \lambda_3(-V_m\sin\gamma - \dot{z}_d)(-\cos\gamma a_v - \ddot{z}_d) \\
&\triangleq F(x, y, z, \gamma, \chi) + G(\gamma, \chi)\underline{a}
\end{aligned} \tag{3.35}$$

where

$$
\begin{aligned}
F = V_m\Big((x-x_d)\cos\gamma\cos\chi + (y-y_d)\cos\gamma\sin\chi - (z-z_d)\sin\gamma\Big) \\
- \Big((x-x_d)\dot{x}_d + (y-y_d)\dot{y} + (z-z_d)\dot{z}\Big) \\
+ V_m\Big(\lambda_1(\cos\gamma\cos\chi - \dot{x}_d)(-\ddot{x}_d) \\
+ \lambda_2(\cos\gamma\sin\chi - \dot{y}_d)(-\ddot{y}_d) + \lambda_3(-\sin\gamma - \dot{z}_d)(-\ddot{z}_d)\Big)
\end{aligned}
\tag{3.36}
$$

$$
G = [G_1, G_2] \tag{3.37}
$$

$$
\begin{aligned}
G_1 = \lambda_1(V_m\cos\gamma\cos\chi - \dot{x}_d)(-\sin\gamma\cos\chi) \\
+ \lambda_2(V_m\cos\gamma\sin\chi - \dot{y}_d)(-\sin\gamma\sin\chi) \\
+ \lambda_3(-V_m\sin\gamma - \dot{z}_d)(-\cos\gamma)
\end{aligned}
\tag{3.38}
$$

$$
\begin{aligned}
G_2 = \lambda_1(V_m\cos\gamma\cos\chi - \dot{x}_d)(-\sin\chi) \\
+ \lambda_2(V_m\cos\gamma\sin\chi - \dot{y}_d)(\cos\chi)
\end{aligned}
\tag{3.39}
$$

$$
\underline{a} = [a_v, a_h]^T \tag{3.40}
$$

Now, the control input is designed as follows,

$$
\underline{a} = RG(G^T RG)^{-1}\Big(- F - \underline{\sigma}^T K \text{sign}(\underline{\sigma})\Big) \tag{3.41}
$$

where R is a positive definite weight matrix, and $K = \text{diag}(k_1, \cdots, k_6),\ k_i > 0,\ i \in \{1, \cdots, 6\}$. Then, the time derivative of the Lyapunov function becomes

$$
\dot{V} = -\underline{\sigma}^T K \text{sign}(\underline{\sigma}) \prec 0 \tag{3.42}
$$

The negative definite $\dot{V}$ shows the Lyapunov stability of the designed controller.

Note that the design procedure of the SMC without the timescale separation technique is quite complicated. It requires the second derivatives of the reference trajectory, and there exists a lot of tuning parameters including $\lambda_i (i \in \{1, 2, 3\}), R$, and K. Also, it is difficult to understand the physical implications of each term in the controller, which also leads to difficulties in tuning the design parameters.

3.6.2 Controller Design with Timescale Separation

In this section, the controller is designed utilzing timescale separation technique. Likewise the previous section, it is assumed that the reference trajectory $[x_d(t), y_d(t), z_d(t)]$ and its time derivatives are given.

Outer-loop Controller Design

In the outer-loop system, the state variables are $\mathbf{x}_{out} = [x, y, z]^T$, and the control inputs are $\mathbf{u}_{out} = [\gamma, \chi]^T$. Let us define the sliding surface $\underline{\sigma}$ as follows,

$$\underline{\sigma} \triangleq [x - x_d, y - y_d, z - z_d]^T \tag{3.43}$$

The Lyapunov function is defined as follows.

$$V \triangleq \frac{1}{2}\underline{\sigma}^T \underline{\sigma} \tag{3.44}$$

The time derivative of the Lyapunov function is represented as follows,

$$\begin{aligned} \dot{V} &= (x - x_d)(\dot{x} - \dot{x}_d) + (y - y_d)(\dot{y} - \dot{y}_d) + (z - z_d)(\dot{z} - \dot{z}_d) \\ &= V_m\big((x - x_d)\cos\gamma\cos\chi + (y - y_d)\cos\gamma\sin\chi - (z - z_d)\sin\gamma\big) \\ &\quad - \big((x - x_d)\dot{x}_d + (y - y_d)\dot{y}_d + (z - z_d)\dot{z}_d\big) \end{aligned} \tag{3.45}$$

Now, let us design the control inputs to satisfy the following equations.

$$\sin\gamma \triangleq \frac{1}{V_m}\Big(k_3\text{sign}(z-z_d)-\dot{z}_d\Big), \quad \gamma\in\Big[-\frac{\pi}{2},\frac{\pi}{2}\Big] \tag{3.46}$$

$$\cos\chi \triangleq -\frac{1}{V_m\cos\gamma}\Big(k_1\text{sign}(x-x_d)-\dot{x}_d\Big) \tag{3.47}$$

$$\sin\chi \triangleq -\frac{1}{V_m\cos\gamma}\Big(k_2\text{sign}(y-y_d)-\dot{y}_d\Big), \quad \chi\in[-\pi,\pi] \tag{3.48}$$

That is,

$$\gamma \triangleq \arcsin\left(\frac{1}{V_m}\Big(k_3\text{sign}(z-z_d)-\dot{z}_d\Big)\right)\in[-\pi/2,\pi/2] \tag{3.49}$$

$$\chi \triangleq \text{atan2}\left(\Big(k_2\text{sign}(y-y_d)-\dot{y}_d\Big),\Big(k_1\text{sign}(x-x_d)-\dot{x}_d\Big)\right)\in[-\pi,\pi] \tag{3.50}$$

where $k_i > 0$ are the control gains which should be selected not to violate the limit of the domain of the trigonometric functions. Then, the time derivative of the Lyapunov function can be written as follows,

$$\begin{aligned}\dot{V} &= -k_1(x-x_d)\text{sign}(x-x_d)-k_2(y-y_d)\text{sign}(y-y_d)-k_3(z-z_d)\text{sign}(z-z_d)\\ &= -k_1|\sigma_1|-k_2|\sigma_2|-k_3|\sigma_3| \prec 0\end{aligned} \tag{3.51}$$

Therefore, the outer-loop system is Lyapunov stable with the designed outer-loop controller.

Inner-loop Controller Design

The outer-loop control inputs are given as desired values for the inner-loop state variables, γ_d and χ_d. The inner-loop controller is designed as follows,

$$a_v = -\frac{V_m}{\varepsilon}(\gamma - \gamma_d) \tag{3.52}$$
$$a_h = -\frac{V_m \cos\gamma}{\varepsilon}(\chi - \chi_d) \tag{3.53}$$

In the inner-loop system, the outer-loop variables are considered as constant. Therefore, the desired values γ_d and χ_d from the outer-loop system are considered as constant. The resulting inner-loop system dyamics can be represented as follows,

$$\dot{\gamma} = -\frac{1}{\varepsilon}(\gamma - \gamma_d) \tag{3.54}$$
$$\dot{\chi} = -\frac{1}{\varepsilon}(\chi - \chi_d) \tag{3.55}$$

From this example of designing SMC, the usefulness of the timescale separation technique is demonstrated. In this particular example of designing SMC for the reference trajectory following, it can be seen that the timescale separation technique significantly reduces the complexity in designing controller. Especially, the SMC design in the outer-loop system is much simpler because of the reduced system dimension by the timescale separation technique. Another benefit of designing controller utilizing timescale separation technique is that the physical meaning of the control command is intuitive. Therefore, it is easy to select the design parameters when tuning the controllers.

3.7 Summary

In this chapter, the time-scale separation and dual-loop system of the missile's dynamic equation of motion are discussed. In the slow-scale system, the fast-scale state variables are considered to be the control inputs. In the fast-scale system, the slow-scale state variables are considered to be constants. Therefore, the missile dynamic system can be understood as dual-loop system. In the following Chapter, the optimal output trajectory shaping (OOTS) algorithm is proposed. The OOTS plays the role of the outer-loop controller and generates the commands γ_{cmd} and χ_{cmd}. Moreover, the OOTS can provide reference trajectory in the state space. Therefore, path following guiance algorithms can be adopted to follow the reference path obtained from OOTS.

Chapter 4

Optimal Output Trajectory Shaping

In this chapter, the optimal output trajectory shaping (OOTS) algorithm is proposed. With OOTS, the trajectory of the system output is shaped and optimized. It is shown that the state trajectory can be manipulated to satisfy constraints by properly optimizing the shape of the output trajectory. The shaping is attained by utilizing parameterized curves. That is, the shape of the output trajectory is manipulated to fit to a parameterized curve, and the trajectory is finally parameterized. In this dissertation, Bézier curve is adopted for the parameterization of the output trajectory. When the output trajectory is parameterized, it can be shown that the state trajectories can also be parameterized when there exists a diffeomorphism for the nonlinear system and the system output. In this chapter, the condition of possibility for the state parameterization of nonlinear system is discussed. Finally, the optimal trajectory for the nonlinear missile system is obtained by OOTS.

4.1 Time Parameterization

The relation between the curve parameter τ and the time t needs to be formulated to specify the output trajectory with given parameterized curve, e.g., the Bézier curve. The time parameterization specifies the relation between the curve paramter and the time with a smooth function. In this study, the following time paramterizing function is considered.

$$t = \frac{\tau}{\mu}, \quad t \in [0, t_f] \tag{4.1}$$

where $\mu = 1/t_f$, and t_f is the final time. The time parameterization is used for a Bézier polynomial [50]. One can utilize various shape of functions when designing the parameterization function, not only the linear function.

Because of the linear relationship, the curve parameter τ increases uniformly in time. Note that the arc-length of the curve may not increase uniformly if the parametric speed of the curve is not constant. The parametric speed $v_p(\tau) = |\mathbf{c}'(\tau)|$ of a curve $\mathbf{c}(\tau)$ is the derivative $\mathrm{d}s/\mathrm{d}\tau$ of the arc length s with respect to the parameter τ. If the parametric speed of a curve is constant, then the points on the curve are distributed uniformly. However, the parametric speed of rational curves, other than a straight line, is not constant [106].

Constant parametric speed is important for the trajectory tracking problem, because a significant change in parametric speed may require an agile acceleration command for the system to follow the trajectory accurately. Of course, it does not matter in the path-following problem which does not specify a position on the curve at a certain time. The vector field-based guidance algorithm considered in this dissertation can be classified as a path-following guidance algorithm. In contrast, the optimal output trajectory tracking problem considers the trajectory tracking problem, and therefore the parametric speed is important.

The time parameterization process considered in this section can also be understood as a re-parameterization of a geometric curve by rational functions. One may easily imagine an arc-length parameterization of a curve which results in the uniform parameterization over the curve with a unit parametric speed. As stated before, it is impossible to exactly parameterize curves by rational functions of the arc length. However, it is natural to ask the designer to approximate the given curve as closely as possible. Let us consider the parameter transformation (or, re-parameterization) $\tau_1 \in [0,1] \to \tau_2 \in [0,1]$ as follows,

$$\tau_1 = \frac{(1-\alpha_r)\tau_2}{\alpha_r(1-\tau_2)+(1-\alpha_r)\tau_2} \tag{4.2}$$

The above transformation provides a rational transformation of the same degree with a single degree of freedom, α_r. Now, the closeness to arc-length parameterization can be measured with the following cost function.

$$J_{closeness} = \int_0^1 (|\mathbf{c}'(\tau_2)| - 1)^2 \mathrm{d}\tau_2 \tag{4.3}$$

The unique solution minimizing $J_{closeness}$ can be found [107,108]. An example of applying this method to a quadratic Bezier curve is shown in Fig. 4.1 [109].

The parameterization of the geometric curve can be attained with various approaches including optimization techniques. The further inquisition of this problem is beyond the scope of this dissertataion. In this study, a simple linear transformation is utilized.

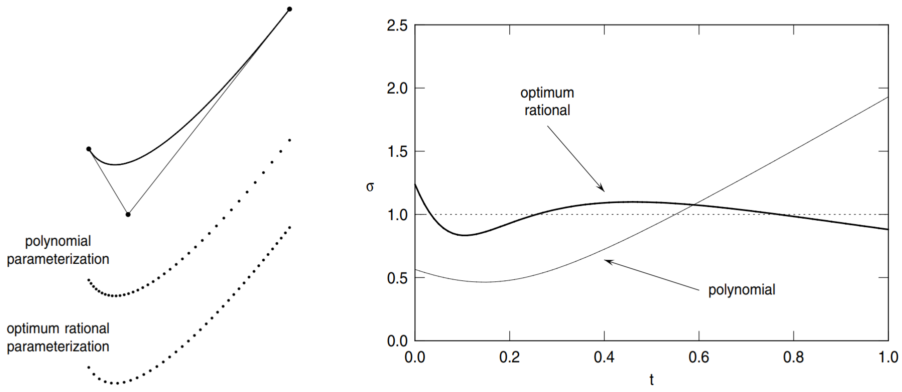


Figure 4.1: Left: the parameter flow along the curve. Right: comparison of parametric speed variations. [109]

4.2 Error dynamics

Assume that desired output trajectory $p(\tau)$ is given with the time parameterization of Eq. (4.1). Let us define the error $z_1 = \tilde{y} = y - p(\tau)$ for the nonlinear SISO system in Eq. (2.26). Differentiating z_1 with respect to time gives,

$$\begin{aligned}
\dot{z}_1 &= \dot{\tilde{y}} = L_f h(x) - \mu p' \\
\dot{z}_2 &= \ddot{\tilde{y}} = L_f^2 h(x) - \mu^2 p'' \\
&\;\;\vdots \\
\dot{z}_\rho &= \tilde{y}^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x) u - \mu^\rho p^{(\rho)}
\end{aligned} \tag{4.4}$$

where $(\cdot)'$ stands for the differentiation with respect to the curve parameter τ. The system can be partially linearized by designing the control algorithm u as follows,

$$u = [L_g L_f^{\rho-1} h(x)]^{-1} \Big(- L_f^\rho h(x) + \mu^\rho p^{(\rho)} + v \Big) \tag{4.5}$$

Substituting Eq. (4.5) into Eq. (4.4) yields

$$\begin{aligned}\dot{z} &= A_c z + B_c v \\ \tilde{y} &= C_c z\end{aligned} \tag{4.6}$$

where (A_c, B_c, C_c) has the canonical form. The error vector $z = [z_1, \cdots, z_\rho]^T$ can be regulated with proper contorl gain vector k as follows,

$$\begin{aligned}v &= -kz \\ k &= [k_1, \cdots, k_\rho]\end{aligned} \tag{4.7}$$

The resulting error dynamic equation can be obtained as follows,

$$\dot{z} = (A_c - B_c k)z = \begin{bmatrix} \mathbf{0}_{\rho-1\times 1} & \mathbf{I}_{\rho-1} \\ -k_1 & \cdots - k_\rho \end{bmatrix} z \tag{4.8}$$

Therefore, the error vector z can be regulated to zero asymptotically.

4.3 State Parameterization

If the relative degree of the system is same with the dimension n of the system, i.e., $\rho = n$, then the states can be represented as the parameterized equations.

Let us consider the case $\rho = n$ and the following diffeomorphism.

$$\xi = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{\rho-1} h(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} \tag{4.9}$$

Now, consider the following assumption.

Assumption 4.1. *The system output precisely follows the given parameterized trajectory $p(\tau)$ using the control input from Eq. (4.5).*

Assumption 4.1 is reasonable because the error dynamic equation, Eq. (4.8), is asymptotically stable. Assumption 4.1 comprehends that $\tilde{y} = \dot{\tilde{y}} = \cdots = \tilde{y}^{(n)} = 0$. Because the transformation $T(x)$ is a diffeomorphism, $T(x)^{-1}$ exists and the state vector can be represented as $x = T(\xi)^{-1}$.

From Assumption 4.1 with the time parameterization in Eq. (4.1), we have

$$\bar{\xi}(\tau) = \begin{bmatrix} p \\ \mu p' \\ \vdots \\ \mu^{\rho-1} p^{(\rho-1)} \end{bmatrix} \tag{4.10}$$

and

$$\bar{x}(\tau) = T^{-1}(\bar{\xi}) = T^{-1}\left(\begin{bmatrix} p \\ \vdots \\ \mu^{\rho-1}p^{(\rho-1)} \end{bmatrix}\right) \tag{4.11}$$

In this study, the parameterized states are denoted as $\bar{\xi}(\tau)$ and $\bar{x}(\tau)$ to avoid confusion. Note that the control input, Eq. (4.5), can also be represented as a parameterized equation using Eq. (4.11).

Remark 4.1. *Final states $x|_{\tau=1}$ can be specified by properly designing $p, \cdots, p^{(n-1)}$. In this study, $p(\tau)$ is designed using the Bézier curve in Eq. (2.43), which can be constructed by the selection of the coefficients $\mathbf{P}_v$.*

Remark 4.2. *There exists a minimum order of Bézier curve to impose the initial and final boundary conditions of the state variables. For a system with n number of boundary conditions, the minimum order of Bézier curve is $N = n-1$ to attain the curve satisfying the boundary conditions.*

4.4 Internal States

Consider a special case of Eq. (2.35) with the following dynamic equation.

$$\dot{\eta} = f_0(\xi) \in \mathbb{R}^{n-\rho} \tag{4.12}$$

That is, the relative degree $\rho < n$, and the internal dynamics is determined by the output $h(x)$ and its derivatives, i.e., ξ.

Under Assumption 4.1, the differential equation can be represented with respect to the curve paramter τ as

$$\mu \bar{\eta}' = f_0(\bar{\xi}(\tau)) \tag{4.13}$$

A definite integral of the equation can be represented for the interval $[0, \tau_1], \tau_1 \in (0, 1]$, as follows,

$$\mu \int_0^{\tau_1} \bar{\eta}' d\tau = \mu \Big(\bar{\eta}|_{\tau=\tau_1} - \bar{\eta}|_{\tau=0} \Big) = \int_0^{\tau_1} f_0(\bar{\xi}(\tau)) d\tau \tag{4.14}$$

Consider the change of variables in Eq. (4.11) and note that $p(\tau)$ is a Bézier curve. Then, the definite integral in Eq. (4.14) becomes a function of the Bézier coefficients, $\mathbf{P}_v$. Therefore, Eq. (4.14) comprehends that the value of the internal states η at given $\tau = \tau_1$ is determined by the Bézier coefficients. This property is used to make the internal state have a desired value at specified time.

Remark 4.3. *The minimum order of Bézier curve in Remark 4.2 is modified considering the constraints on the internal state variables. The minimum order of Bézier curve is $N = n + m - 1$ in order to consider m number of constraints on the internal state variables.*

4.5 Differentially Flat System

When the nonlinear system is diffrentially flat, the full state parameterization and control input parameterization become available. Flatness can be understood as an extension of the notion of controllability from linear systems to nonlinear systems. When the nonlinear system is a flat system with a flat output, all the states and inputs can be explicitly expressed in terms of the flat output and its finite number of derivatives [110].

Definition 4.1. *Let us consider a nonlinear system of the following form.*

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{4.15}$$

where

$$\mathbf{u}(t) \in \mathbb{R}^m, \quad \mathbf{x}(t) \in \mathbb{R}^n \ \ \textit{and} \ \ \mathrm{Rank}\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = m \tag{4.16}$$

This system is flat if there exists an output $\mathbf{y}(t)$ represented as follows,

$$\mathbf{y}(t) = (y_1(t), \cdots, y_m(t)) \tag{4.17}$$

and the following conditions are satisfied.

- *The signals $y_i,\ i = 1, \cdots, m$ can be represented as functions of the states $x_i,\ i = 1, \cdots, n$, inputs $u_i,\ i = 1, \cdots, m$, and a finite number of derivatives with respect to time $\frac{d^k}{dt^k}u_i,\ k = 1, \cdots, \alpha_i$, i.e., $\mathbf{y} = \mathbf{\Phi}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \cdots, \mathbf{u}^{\alpha})$.*

- *The states $x_i,\ i = 1, \cdots, n$, and inputs $u_i,\ i = 1, \cdots, m$ can be represented as functions of the outputs $y_i,\ i = 1, \cdots, m$, and of its derivatives with respect to time $\frac{d^k}{dt^k},\ i = 1, \cdots, m$.*

- *The components of* $\mathbf{y}$ *are differentially independent.*

The output satisfying these conditions is flat output and the system is a flat system.

The flatness property is often utilized for solving the trajectory planning problem. Because the states and inputs of the flat system are representable as functions of the outputs and their derivatives, the OOTS technique can readily be adopted for the flat system. An example of OOTS for a flat system will be presented lated.

4.6 Optimal Output Trajectory Shaping using Bézier Curve

In this section, OOTS algorithm using Bézier curve is introduced. Note that optimality is attained by properly selecting the free coefficients of the Bézier curve, which are not specified by the boundary conditions. The optimization problem of the Bézier curve coefficients can be solved by any constrained nonlinear optimization algorithms including sequencial quadratic programming (SQP) method.

OOTS has several merits. First, boundary conditions are automatically satisfied by specifying the coefficients of the Bézier curve. Second, the difficulty in the implementation is relatively low because the algorithm can easily find the solution of the OOTS compared to indirect optimization approach solving a two-point boundary value problems (TPBVP). Third, OOTS requires relatively low computational cost to solve the problem compared to the conventional trajectory optimization problem. Besides, if the parameterizations of the state and input variables are possible, it is easy to impose specific constraints on the trajectory or manipulate the trajectory of variables by changing the coefficients.

Remark 4.4. *Because it is not possible for a Bézier curve with specific degree N to represent an arbitrary function perfectly, the solution trajectory $p(\tau)$ for the OOTS can be understood as a sub-optimal solution for the trajectory optimization problem. If the degree of the Bézier curve increases, i.e., $N \to \infty$, then the OOTS solution approaches the optimal solution.*

4.6.1 Higher Order Curves and Degree Increase

In Remarks 4.2 and 4.3, the minimum order of the Bézier curve ($N = n + m - 1$) is considered, where the number of boundary conditions for state variables is n and the number of constraints on the internal state variables is m. If the degree of Bézier curve is higher than the minimum order, $N = n+m-1$, then the remaining coefficients can be used to optimize the trajectory.

Note that any Bézier curve can be represented with another Bézier curve with higher order. Therefore, the higher order Bézier curve includes the lower order Bézier curve. In other words, a curve represented by a Bézier curve can always be represented as a Bézier curve with higher order.

Remark 4.5. *A Bézier curve with degree N can be represented by a Bézier curve with degree M for $M > N$. A degree N Bézier curve with control points $P_0, \cdots, P_N$ is equivalent to the degree $N+1$ Bézier curve. One particular parameterization of the control points are: $P'_0, \cdots, P'_{N+1}$ where $P'_k = \frac{k}{N+1} P_{k-1} + (1 - \frac{k}{N+1}) P_k$.*

Because a Bézier curve with degree N can be represented with another Bézier curve with degree $N+1$, the optimal Bézier curve with degree N can be utilized as an initial guess for obtaining the optimal Bézier curve with degree

$N+1$, after the degree increases. An example of utilizing the optimal solution of the lower degree as the initial guess for the higher degree is presented at the end of this chapter. As N increases, the Bézier curve has more degree of freedom and the optimality would increase, i.e., the optimal cost function value decreases. However, in the practical aspect, it is worth to know the proper value of N. In practice, thresholding the increment of the optimal cost function value is a typical strategy to determine the proper degree N. That is, the strategy is to stop increasing N when the increment of the optimal cost function value is below a predefined threshold value.

4.6.2 Optimization Problem Formulation

The optimization problem in output trajectory shaping using the Bézier curve can be formulated to determine the remaining free coefficients of the Bézier curve as follows,

$$J(P_0, \cdots, P_N, \mu) = \phi(\bar{\xi}(P_0, \cdots, P_N), \mu)|_{\tau=1} + \int_0^1 L(\bar{\xi}(\tau, P_0, \cdots, P_N), \tau) d\tau \tag{4.18}$$

where $P_i, i \in (1, \cdots, N)$ are the coefficients of the Bézier curve.

The boundary conditions for the output and states are already imposed on the some coefficients of the Bézier curve before solving the optimization problem, and therefore the number of coefficients to be optimized is reduced. Note that the final time can also be considered by including μ as one of the optimization variables.

When there exist boundary conditions for output $y = h(x) = \xi_1$ and the initial/final conditions are y_i and y_f, then $\bar{\xi}_1(0) = p(0) = P_0 = y_i$ and $\bar{\xi}_1(1) = p(1) = P_N = y_f$. The first and final coefficients are determined by the output boundary conditions. Also, when there exist boundary conditions for state $x_k = T^{-1}(\xi_k)$ and the initial/final conditions are $x_{k,i}$ and $x_{k,f}$, then $\bar{\xi}_k(0) = \mu^{k-1}p^{(k-1)}(0) = T(x_{k,i})$ and $\bar{\xi}_k(1) = \mu^{k-1}p^{(k-1)}(1) = T(x_{k,f})$. Because $p^{(k-1)}(\tau)$ is linear function of $P_0, \cdots, P_N$, one of any $P_i, i \in (1, \cdots, N)$ can be eliminated from the cost function, which reduces the number of coefficients to be optimized. Therefore, Bézier curve with degree $N > n + m - 1$ should be used for the OOTS.

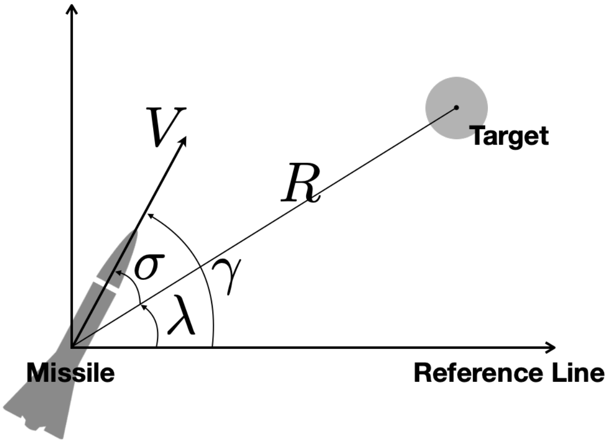


Figure 4.2: Missile engagement geometry

4.7 A Motivating Example for OOTS - I

A basic example of OOTS is considered. The algorithm proposed in this Chapter is applied to the nonlinear model of planar missile engagement dynamics against a stationary target. The system equation can be represented as follows,

$$\begin{aligned} \dot{R}(t) &= -V\cos\sigma \\ \dot{\sigma}(t) &= \frac{V}{R}\sin\sigma + \dot{\gamma} \end{aligned} \tag{4.19}$$

where R is the distance to the target from the missile, V is a speed, σ is the look angle, and γ is the flight path angle. It is assumed that the speed of the missile is constant. Note that the nomenclature here are only valid in the examples of this chapter. The missile engagement geometry is shown in Fig. 4.2.

The state variables are $x = [x_1, x_2]^T = [R, \sigma]^T$, and the control input is $\dot{\gamma}$.

The nonlinear system can be rewritten in the form of Eq. (2.26) as follows,

$$\begin{aligned}\dot{x}_1 &= -V\cos x_2\\ \dot{x}_2 &= \frac{V}{x_1}\sin x_2 + u\\ y &= h(x) = x_1\end{aligned} \tag{4.20}$$

The output $y = x_1$ is selected for the missile to intercept the target. Therefore, the boundary conditions for the output are $y(0) = R_0$ and $y(t_f) = 0$.

Differentiating the output with respect to time gives

$$\begin{aligned}\frac{dh(x)}{dt} &= L_f h(x) = -V\cos x_2\\ \frac{d^2h(x)}{dt^2} &= L_f^2 h(x) + L_g L_f h(x)u = \frac{V^2}{x_1}(\sin x_2)^2 + V\sin x_2 u\end{aligned} \tag{4.21}$$

Because the relative degree is $\rho = n = 2$, the state parameterization is possible, and the change in variables is defined as the following diffeomorphism.

$$\xi = T(x) = \Big[h(x) \quad L_f h(x)\Big]^T = \Big[x_1 \quad -V\cos x_2\Big]^T \tag{4.22}$$

Control algorithm and state parameterization

The control input is designed with positive gains k_1 and k_2 as follows,

$$u = L_g L_f h(x)^{-1}\Big(-L_f^2 h(x) + \mu^2 p'' + v\Big) \tag{4.23}$$

$$v = -[k_1 \quad k_2]z \tag{4.24}$$

$$z = [z_1 \quad z_2]^T = \begin{bmatrix} h(x) - p(\tau) \\ L_f h(x) - \mu p' \end{bmatrix} \tag{4.25}$$

where $p(\tau)$ is the desired output trajectory with the time parameterization in Eq. (4.1). The boundary conditions for the states are given as follows,

$$\begin{aligned} x_1(0) = R(0) = R_0, &\qquad x_1(t_f) = R(t_f) = 0 \\ x_2(0) = \sigma(0) = \sigma_0, &\qquad x_2(t_f) = \sigma(t_f) = 0 \end{aligned} \tag{4.26}$$

The parameterized states can be obtained using Eq. (4.11) as follows,

$$\bar{x}(\tau) = \begin{bmatrix} \bar{x}_1(\tau) & \bar{x}_2(\tau) \end{bmatrix}^T = T^{-1}(\bar{\xi}) = \begin{bmatrix} p(\tau) & \cos^{-1}(-\mu p'(\tau)/V) \end{bmatrix}^T \tag{4.27}$$

Trajectory generation

Let us consider the output trajectory $p(\tau)$ represented by the Bézier curve $B_n(\tau)$. The cubic Bézier curve and its derivative are given as follows,

$$\begin{aligned} B_3(\tau) &= (1-\tau)^3 P_0 + 3(1-\tau)^2 \tau P_1 + 3(1-\tau)\tau^2 P_2 + \tau^3 P_3 \\ B_3'(\tau) &= 3(1-\tau)^2(P_1 - P_0) + 6(1-\tau)\tau(P_2 - P_1) + 3\tau^2(P_3 - P_2) \end{aligned} \tag{4.28}$$

where $P_i, i = 1, \cdots, 3$, are scalar control points. The control points can be determined to satisfy the boundary conditions, Eq. (4.26), as

$$\begin{aligned} P_0 = R_0, &\qquad P_3 = 0 \\ \cos^{-1}(-\mu B_3'(0)/V) = \sigma_0, &\qquad \cos^{-1}(-\mu B_3'(1)/V) = 0 \end{aligned} \tag{4.29}$$

Using Eq. (4.28) in Eq. (4.29), we have

$$\begin{aligned} P_1 &= -\frac{V}{3\mu}\cos\sigma_0 + R_0 \\ P_2 &= \frac{V}{3\mu} \end{aligned} \tag{4.30}$$

Desired impact time, s	Impact time error, s	Bézier coefficients
30	0.0002	[2000, 1250, 750, 750, 0]
60	0.0026	[2000, 500, 1500, 1500, 0]
75	0.0006	[2000, -250, 2250, 2250, 0]
90	0.0035	[2000, -1000, 3000, 3000, 0]
120	0.0017	[2000, -1750, 3750, 3750, 0]
150	0.0034	[2000, 125, 1875, 1875, 0]

Table 4.1: Guidance error and Bézier coefficients for impact time control guidance (Example 1)

In this particular example, all the coefficients of the Bézier curve are determined by the boundary conditions. Therefore, there does not exist any design parameters to be optimized.

Simulation results

Numerical simulation is conducted for various final times, i.e., $t_f = 30$, 60, 90, 120, and 150 sec. The boundary conditions are chosen as $R_0 = 2,000$ m, $\sigma_0 = 0$ deg, and speed of the missile $V = 100$ m/s. An initial range error of 300 m is considered for one simulation case, i.e., $t_f = 75$ sec. The numerical reslts of the simulation are shown in Fig. 4.3. As expected, the states $x(t)$ of all cases follow the trajectory $\bar{x}(\tau)$ well. The dashed line shows a case in which there is an initial range error, but the system follows the prescribed trajectory well by the proposed output trajectory tracking controller. The guidance error and Bézier coefficients are summarized in Table 4.1.

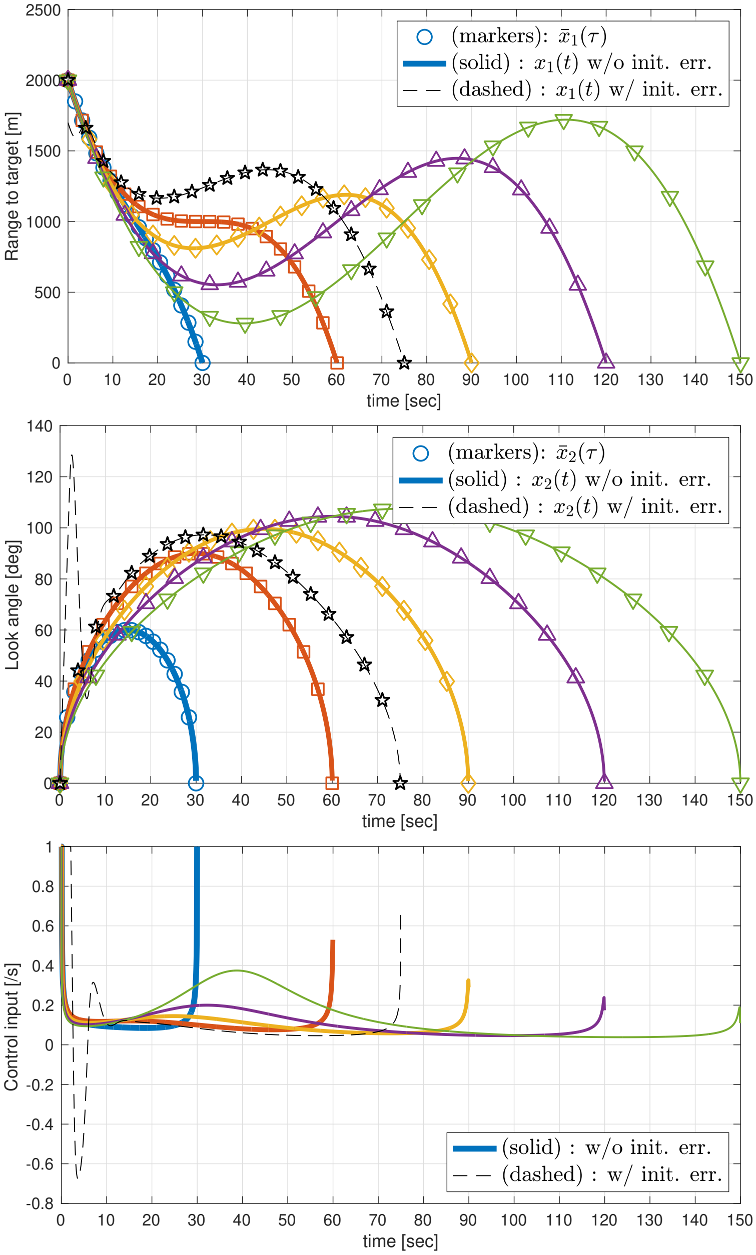


Figure 4.3: Simulation result for ITCG

4.8 A Motivating Example for OOTS - II

A nonlinear system with internal state is considered. In this example, impact angle and time control guidance (IATCG) [111] is considered and the following additional internal dynamics is considered in addition to Eq. (4.19).

$$\dot{\lambda}(t) = -\frac{V}{R} \sin \sigma \tag{4.31}$$

where λ is the line-of-sight angle. Note that the internal dynamics is in the form of Eq. (4.12), that is, the internal state value at a certain moment can be manipulated by shaping trajectory $\xi(\tau)$ in Eq. (4.14). Based on this action, the desired impact angle can be attained by manipulating the final value of the internal state $\lambda(t_f)$. To apply an additional condition for the internal state λ, a Bézier curve with $n = 4$ is used, i.e., $p(\tau) = B_4(\tau)$,

$$\begin{aligned} B_4(\tau) &= (1-\tau)^4 P_0 + 4(1-\tau)^3 \tau P_1 + 6(1-\tau)^2 \tau^2 P_2 + 4(1-\tau)\tau^3 P_3 + \tau^4 P_4 \\ B_4'(\tau) &= 4(1-\tau)^3(P_1 - P_0) + 12(1-\tau)^2\tau(P_2 - P_1) + 12(1-\tau)\tau^2(P_3 - P_2) \\ &\quad + 4\tau^3(P_4 - P_3) \end{aligned} \tag{4.32}$$

By applying the boundary conditions of Eq. (4.26), we have

$$\begin{aligned} &P_0 = R_0 \\ &P_4 = 0 \\ &\cos^{-1}(-\mu B_4'(0)/V) = \sigma_0 \\ &\cos^{-1}(-\mu B_4'(1)/V) = 0 \end{aligned} \tag{4.33}$$

Using Eqs. (4.32) and (4.33), we have

$$\begin{aligned} P_1 &= -\frac{V}{4\mu}\cos\sigma_0 + R_0 \\ P_3 &= \frac{V}{4\mu} \end{aligned} \tag{4.34}$$

The remaining coefficient P_2 is used to determine the internal state value at the final time.

Internal state parameterization

The internal dynamics can be rewritten as

$$\dot{\eta} = -\frac{V}{x_1}\sin x_2 \tag{4.35}$$

The above equation can be represented with respect to τ as

$$\mu\bar{\eta}'(\tau) = \frac{V}{\bar{x}_1(\tau)}\sin\bar{x}_2(\tau) = \frac{V}{p(\tau)}\sin\Big(\cos^{-1}\Big(-\frac{\mu}{V}p'(\tau)\Big)\Big) \tag{4.36}$$

The definite integral representation can be obtained as follows,

$$\mu\Big(\bar{\eta}(\tau_1) - \bar{\eta}(0)\Big) = \int_0^{\tau_1}\frac{V}{p(\tau)}\sin\Big(\cos^{-1}\Big(-\frac{\mu}{V}p'(\tau)\Big)\Big)d\tau \tag{4.37}$$

where $\tau_1 = 1$, $\bar{\eta}(\tau_1) = \eta_f = \lambda(t_f)$ and $\bar{\eta}(0) = 0 = \lambda(t_0)$. It is difficult to analytically express the definite integral of the polynomials in the transcendental function. However, there exists only one parameter P_2 to be determined, and therefore it is easy to find P_2 satisfying Eq. (4.37) using numerical root-finding algorithm.

Desired impact angle, deg	Impact angle error, deg	Bézier coefficients
60	0.3429	[2000, 2000, 1213.49, 625, 0]
90	0.5023	[2000, 2000, 1133.25, 625, 0]
120	0.6368	[2000, 2000, 998.68, 625, 0]
150	0.7020	[2000, 2000, 811.90, 625, 0]

Table 4.2: Guidance error and Bézier coefficients for impact time and impact angle control guidance (Example 2)

Remark 4.6. *Note that the existence of the solution to Eq. (4.37) is not guaranteed in general. The existence of the solution depends on the given nonlinear system and its boundary conditions. For the impact time and angle control problem, there is limited range of solutions, and not every η_f can be attained for the given boundary conditions and t_f.*

Simulation results

Numerical simulation is conducted for various desired impact angles, i.e., the final LOS angle. The final values of the internal state are chosen as $\eta_f = -60$, -90, -120, and -150 deg. The boundary conditions are chosen as $R_0 = 2,000$ m, $\sigma_0 = 90$ deg, and the speed of the missile $V = 100$ m/s. The final time is $t_f = 25$ sec. The simulation results are shown in Figs. 4.4 and 4.5. The states follow the desired trajectory well, and all the boundary conditions are satisfied. Furthermore, the final value of the internal state $\eta(t_f)$ is satisfied. The guidance error and Bézier coefficients are summarized in Table 4.2.

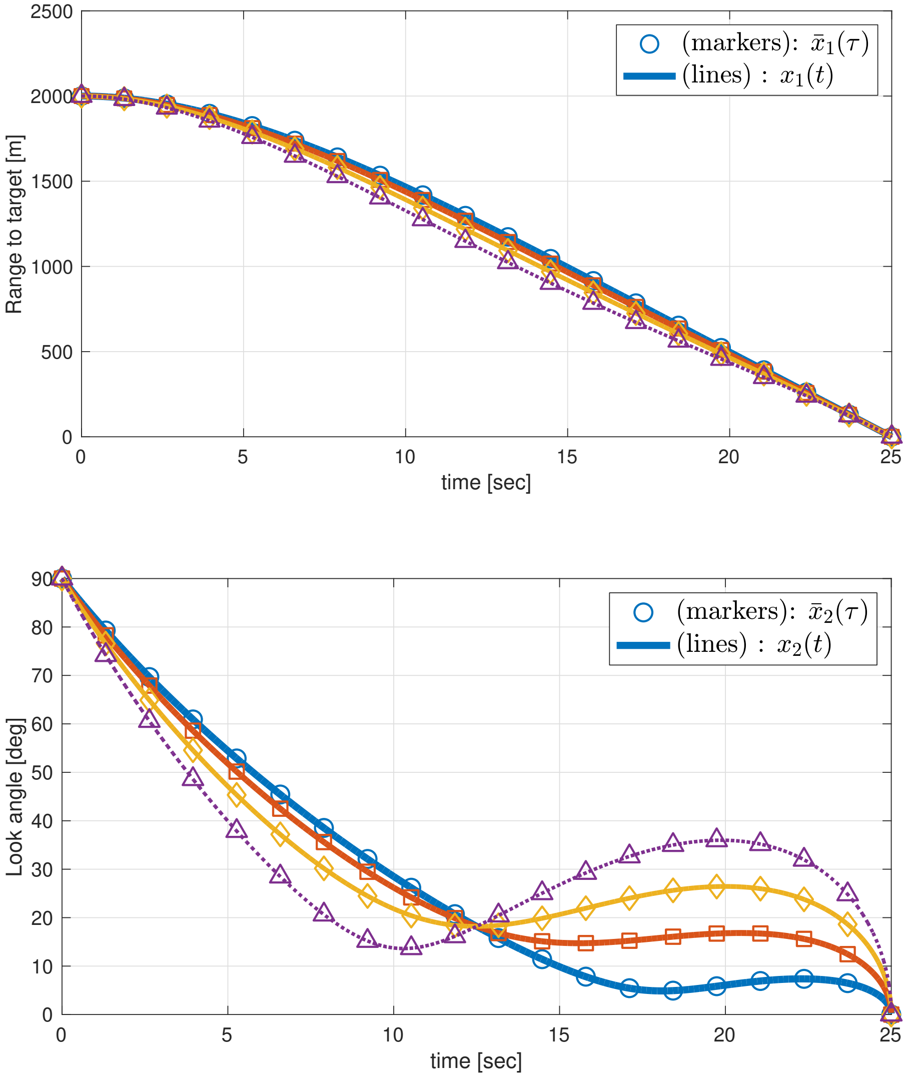


Figure 4.4: Simulation result for IATCG (range, lead angle)

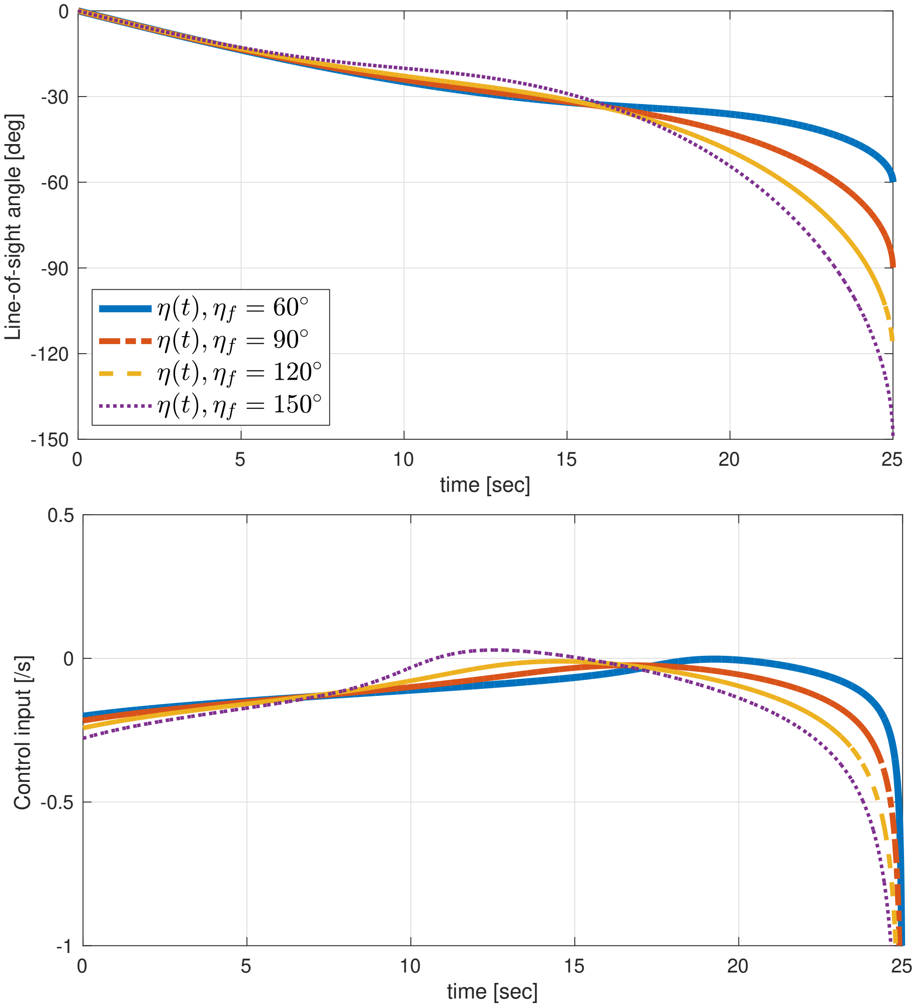


Figure 4.5: Simulation result for IATCG (LOS angle, inputs)

4.9 A Motivating Example for OOTS - III

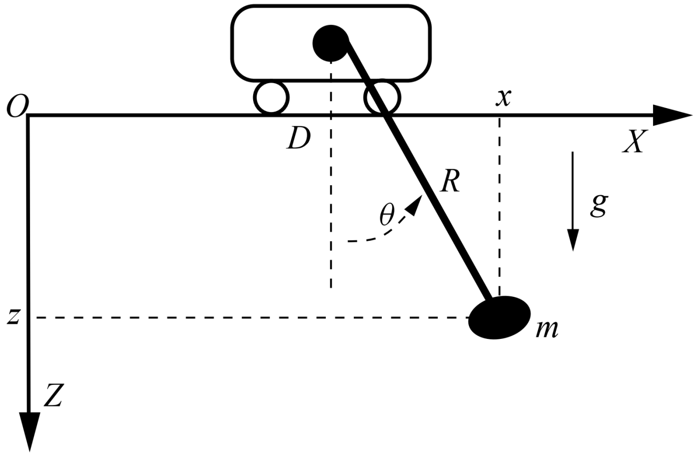


Figure 4.6: Description of the 2D crane

OOTS for the 2-D crane, which is a flat system is considered. This example is exploited in [110], which is a classical and fundamental object of in control theory. A trolly travels and rolling up and down the rope. The 2D crane is described in Fig. 4.6. The traveling position D and the hoisting rope length R are the control variables, and the coordinate of the payload m on the plane (x, z) are the system output. The rope dynamics are neglected in this example. Then, the dynamic model of this system is described as follows,

$$\begin{aligned} m\ddot{x}(t) &= -T(t)\sin\theta(t) \\ m\ddot{z}(t) &= -T(t)\cos\theta(t) + mg \\ x(t) &= R(t)\sin\theta(t) + D(t) \\ z(t) &= R(t)\cos\theta(t) \end{aligned} \tag{4.38}$$

where T denotes the tension of the rope, and θ denotes the angle between the rope and the vertical axis OZ. From the dynamic model of the 2D crane system, it can be seen that $D(t)$, $R(t)$, $T(t)$, and $\sin\theta(t)$ can be represented as algebraic functions of the outputs $x(t)$ and $z(t)$.

$$\begin{aligned} D &= x - \frac{\ddot{x}z}{\ddot{z}-g} \\ R^2 &= z^2 + \left(\frac{\ddot{x}z}{\ddot{z}-g}\right)^2 \\ T &= \frac{mR(g-\ddot{z})}{z} \\ \sin\theta &= \frac{x-D}{R} \end{aligned} \tag{4.39}$$

Therefore, the 2D crane's dynamic system is flat with (x, z) as flat output.

The objective of this problem is to carry the payload from the steady-state $R = R_1 > 0$ and $D = D_1$ at time t_1, to the steady-state $R = R_2 > 0$ and $D = D_2$ at time $t_2 > t_1$. To this end, any oscillations should be attenuated at the end of the trajectory. Therefore, the following boundary conditions for the outputs should be satisfied. For all $i \in \{1, 2\}$ and $r \in \{1, 2, 3, 4\}$,

$$\begin{aligned} (x, z)(t_i) &= (D_i, R_i) \\ \frac{d^r}{dt^r}(x, z)(t_i) &= 0 \end{aligned} \tag{4.40}$$

Let us consider the output trajectories as Bézier curves as follows,

$$\begin{aligned} \bar{x} &= p_x(\tau) \\ \bar{z} &= p_z(\tau) \end{aligned} \tag{4.41}$$

To satisfy the boundary conditions in Eq. (4.40), the minimum degree of the Bézier curve is 9. The parameterizations are given as follows,

$$
\begin{aligned}
\bar{D}(\tau) &= p_x - \frac{\mu^2 p_x'' p_z}{\mu^2 p_z'' - g} \\
\bar{R}(\tau) &= \sqrt{p_z^2 + \left(\frac{\mu^2 p_x'' p_z}{\mu^2 p_z'' - g} \right)^2}
\end{aligned}
\tag{4.42}
$$

$$
\begin{aligned}
\bar{T}(\tau) &= \frac{m\bar{R}(g - \mu^2 p_z'')}{p_z} \\
\sin\bar{\theta}(\tau) &= \frac{p_x - \bar{D}}{\bar{R}}
\end{aligned}
\tag{4.43}
$$

The boundary conditions are represented as follows. For $i \in \{x, z\}$,

$$
\begin{aligned}
\left. \frac{d^k}{d\tau^k} p_i(\tau) \right|_{\tau=0} &= \frac{N!}{(N-k)!} \sum_{j=0}^{k} \binom{k}{j} (-1)^{j+k} P_j^i \\
\left. \frac{d^k}{d\tau^k} p_i(\tau) \right|_{\tau=1} &= \frac{N!}{(N-k)!} \sum_{j=0}^{k} \binom{k}{j} (-1)^{j+k} P_{N-j}^i
\end{aligned}
\tag{4.44}
$$

where $P_j^i,\ i \in \{x, z\}$ is the j-th control point of the curve $p_i(\tau)$, and N is the degree of the Bézier curve. The output trajectories $p_x(\tau)$ and $p_z(\tau)$ satisfying the dynamic model of 2D crane and the boundary conditions are computed using Bézier curve of degree $N = 9$. The parameters are summarized in Table. 4.3. The fixed initial and final times (t_1, t_2) implies $\mu = 1/(t_2 - t_1)$.

The trajectory planning result is shown in Figs. 4.7 and 4.8. Note that a feasible trajectory satisfying the boundary conditions are obtained. In this planning with $N = 9$ and fixed μ, no optimization process is required because all of the control points and μ are defined by the boundary conditions. If the designer consider Bézier curves of $N > 9$ or final-time free problem, a feasible solution of the trajectory planning problem can be obtained by optimizing the free control points and μ.

Parameter	m, kg	R_1, m	R_2, m	D_1, m	D_2, m	t_1, sec	t_2, sec
Value	10	0.4	0.4	0	1	0	1

Table 4.3: Parameter values for trajectory planning of 2D crane

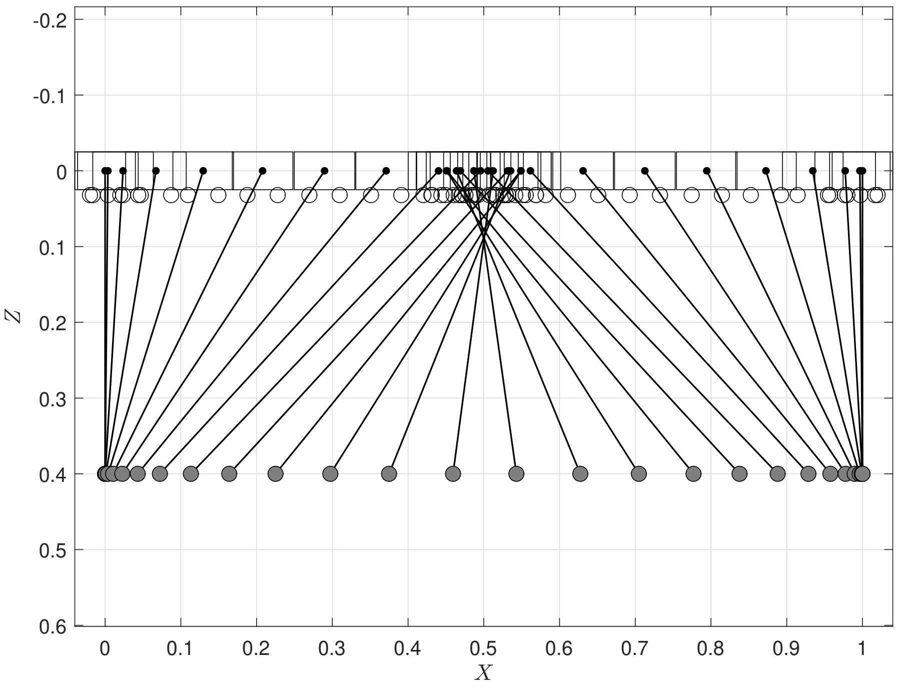
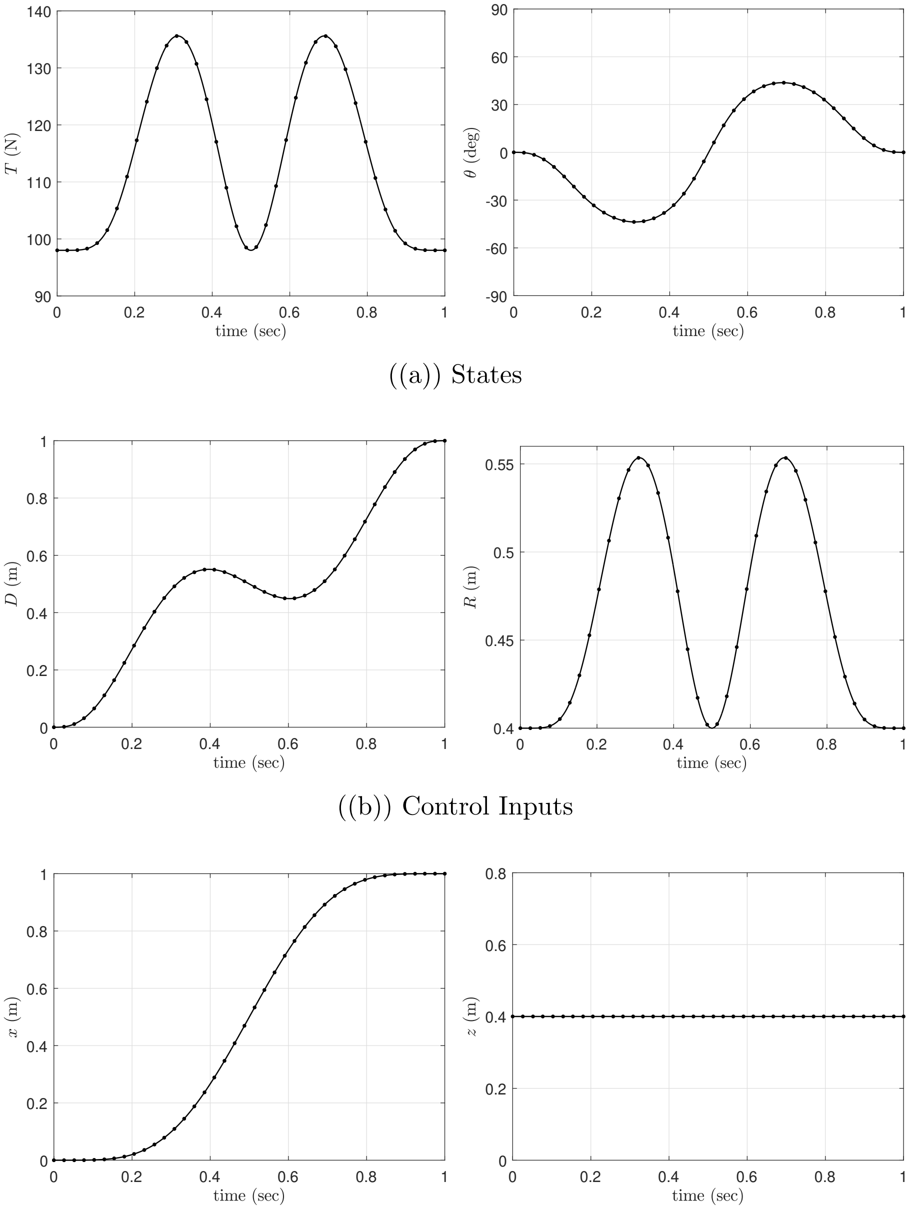


Figure 4.7: The successive motions of the trolley and the payload



((a)) States

((b)) Control Inputs

((c)) Outputs

Figure 4.8: The trajectory planning for 2D crane using OOTS

4.10 A Motivating Example for OOTS - IV

In this section, the comparison between the proposed method and the conventional trajectory optimization technique using direct collocation method is conducted. For the comparison, a scalar linear system, for which analytic solution can be calculated, is considered. For the OOTS, the degree of the Bézier curve is increased using the method given in Remark 4.5. The optimality and the computation times of the solutions are compared.

Let us consider the following scalar linear system.

$$\dot{x}(t) = ax(t) + bu(t) \tag{4.45}$$

$$y(t) = cx(t) \tag{4.46}$$

The boundary conditions are given as the initial and final condition for the output and represented as follows,

$$y(0) = y_0, \quad y(T) = y_f \tag{4.47}$$

where T is the final time. Let us consider the objective function which minimizes the energy as follows,

$$\min_u J = \frac{1}{2}\int_0^T u(t)^2 dt \tag{4.48}$$

The proposed OOTS method is used to obtain the optimal trajectory with respect to the cost function in Eq. (4.48). Then, the optimality of the OOTS solution is validated. To this end, the analytic solution of the optimization problem is obtained.

4.10.1 Analytic Solution

The state and costate equations are represented as follows,

$$\begin{aligned} \dot{x} &= ax - b^2\lambda \\ \dot{\lambda} &= -a\lambda \end{aligned} \tag{4.49}$$

The analytic solutions are obtained as

$$\begin{aligned} \lambda(t) &= e^{a(T-t)}\lambda(T) \\ x(t) &= x(0)e^{at} - \frac{b^2}{a}\lambda(T)e^{aT}\sinh(-at) \\ u(t) &= -b\lambda(t) \end{aligned} \tag{4.50}$$

When the final state $x(T)$ is given, the solution can be obtained as follows,

$$\lambda(T) = \Big(x(T) - x(0)e^{aT}\Big)\Big(\frac{b^2}{a}e^{aT}\sinh(-aT)\Big)^{-1} \tag{4.51}$$

Substitutiting Eq. (4.51) to Eq. (4.50), the analytic solution for the optimal state and the optimal control input can be obtained.

4.10.2 Direct Collocation Solution

The trajectory optimization using direct collocation method is a process of computing an open-loop solution to an optimal control problem. In the direct collocation method, the state and control trajectories are approximated using polynomial splines. A trapezoidal method, Hermite-Simpson method, and Runge-Kutta method are utilized to obtain the optimal solution.

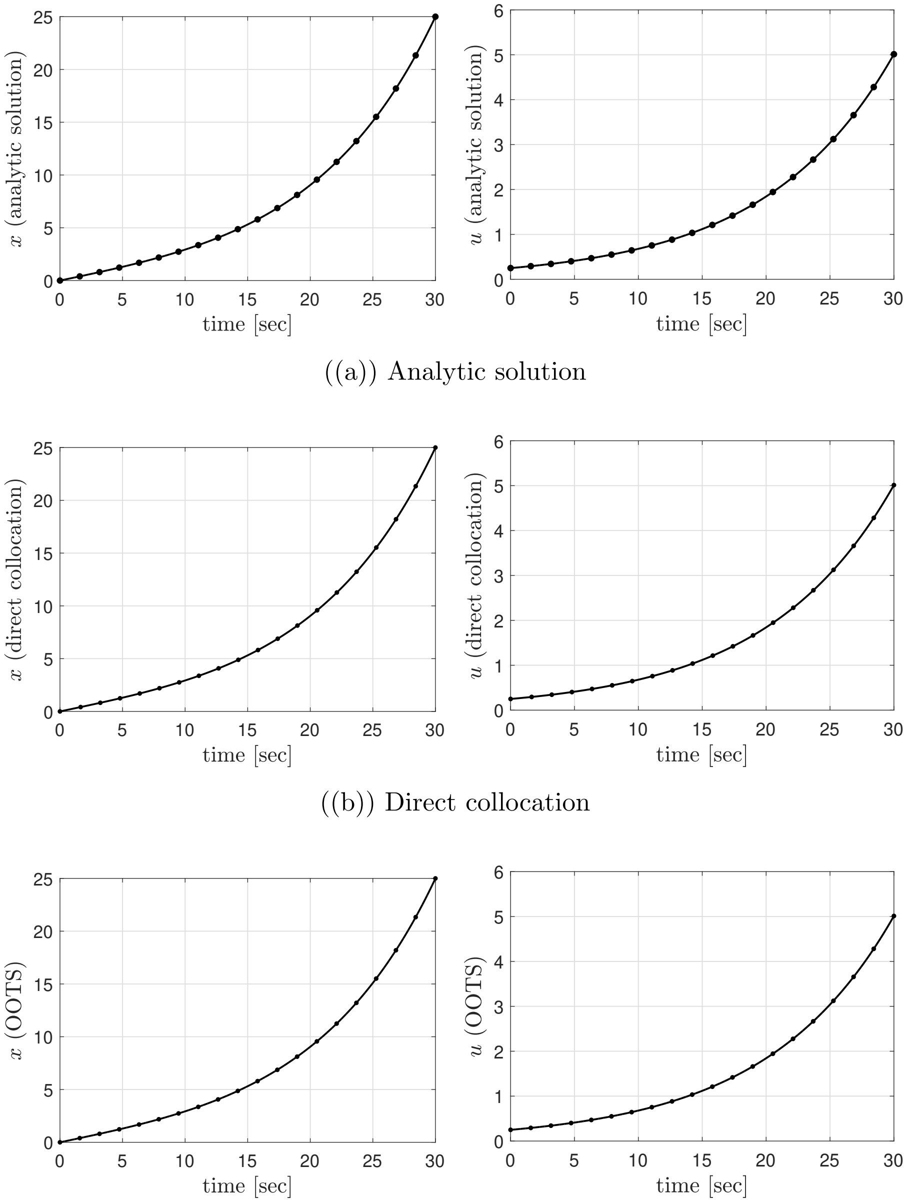
4.10.3 OOTS Solution

The objective of this example is twofold. The first objective is to compare the optimality (cost function value) and computation time of the OOTS solution.

The merit of the OOTS lies in that the required computation time for obtaining the (sub-)optimal solution is smaller, not sacrificing the optimality much. The second objective is to verify that the optimality of the OOTS solution increases as the degree of the Bézier curve, N, increases. To this end, the degree increase technique stated in Remark 4.5 is utilized.

Figure 4.9 shows the trajectories of the analytic solution (4.9(a)), the direct collocation solution (4.9(b)), and the OOTS solution (4.9(c)). The direct collocation solution trajectory in Fig. 4.9(b) is obtaied using Runge-Kutta method with 100 segements. That is, the trajectory is approximated using 100 number of polynoimal splines. The OOTS solution trajectory in Fig. 4.9(c) is obtained by Bézier curve of degree $N = 7$ of 1,000 segments. That is, 1,000 number of uniformly sampled points on the curve are evaluted to evalute the curve during the numerical optimization process. Note that the obtained trajectories are almost same.

Figure 4.10 shows the comparison between methods. The cost function value and optimization error are plotted in Fig. 4.10(a). The optimization error is computed based on the cost function value of the analytic solution. The numbers of segments for OOTS solutions are 1,000, and the numbers of segments for direct collocation solutions are 100. The number of segments for OOTS and the direct collocation method affects the evaluations of the curve. Methods using larger number of segments provide more accurate result, and therefore increasing the number of segments has an affirmative effect on the quality of the optimal solution. However, in both of the methods, the computation time increases when the number of segments increases, because it increases the number of evaluations. Especially for the direct collocation method, increasing the

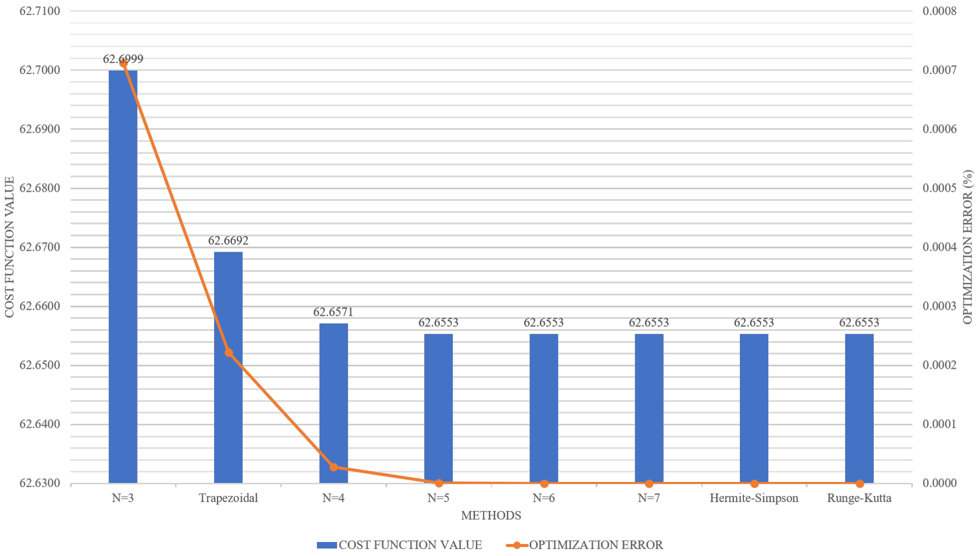


((a)) Analytic solution

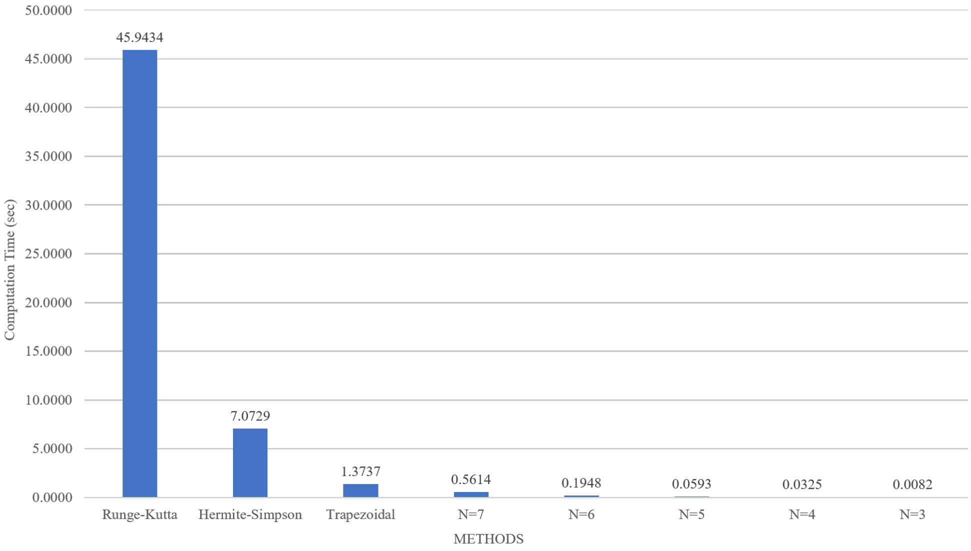
((b)) Direct collocation

((c)) OOTS

Figure 4.9: Example III - Optimization results



((a)) Cost function value and optimization error



((b)) Computation time

Figure 4.10: Example III - Comparison between methods

number of segments results in an increase in the number of parameters to be optimized. Consequently, there exists a risk that the numerical solver may not find a feasible solution when the number of segments is too large in the direct collocation method. In contrast, the number of segments has nothing to do with the number of parameters to be optimized for the OOTS.

Note that the OTTS solution of $N = 3$ shows a relatively high optimization error. The optimization error of the direct collocation solution with the trapezoidal method is the second biggest. As N increases, the optimality of the OTTS solution gets better and the optimization error decreases. The direct collocation solutions with the Hermite-Simpson method and Runge-Kutta method shows similar optimality when $N = 7$ in this particular problem. From the results shown in Fig. 4.10(a), it can be stated that increasing N has an affirmative effect on the optimality because the curve of a higher degree has more degree of freedom.

Figure 4.10(b) compares the computation time. The simulation was conducted in AMD Ryzen 9 3900x Processor (3.80Ghz), using MATLAB R2020a. Note that much more computation time is required for the direct collocation solution than the OOTS solutions, although the number of segments is smaller in the direct collocation method. Especially, the computation time of the Runge-Kutta method is about 80 times longer than that of OOTS of $N = 7$, which shows similar optimality. The computation time of OOTS increases as N increases, because the number of parameters to be optimized increases. In the process of solving the optimization problem, one can stop increasing N when enough optimality is obtained. However, in practice, the optimization error cannot be computed for the problems that the analytic solution is not available.

Therefore, thresholding the decrement of the optimal cost function value can be a possible stopping strategy.

4.11 Summary

In this chapter, it is shown that the output trajectory of a nonlinear SISO system can be shaped and optimized using a parameterized curve including Bézier curve. Also, not only the output trajectory but also the state trajectory can be represented by parameterized curves. Consequently, the optimal state trajectory can be easily obtaind by optimizing the shape of the parametric curve. In OOTS, tracking to the reference trajectory for the system is attained by the feedback linearization control algorithm. However, tracking to the reference trajectory can also be attained by other guidance and control techniques. In the following Chapter, vector field-based guidance (VFG) law is designed for the missile system. The missile can be guided to track the optimal reference trajectory using VFG or feedback linearization control. The pros and cons of the two different tracking techniques are compared by numerical simulation.

Chapter 5

Vector Field-based Guidance Law Design

In this chapter, the VFG algorithm for missile system is addressed. In the VFG algorith, a vector field is defined in the position space. The missile is controlled to follow the direction of the vector field and converge to a reference trajectory, which is designed in the previous chapter. Therefore, designing VFG can be understood as the matter of designing the direction of the vector field in the position space. The reference curve can be a target's predicted trajectory that has to be intercepted by the missile, or an optimal trajectory that is generated prior to the launch of the missile for the mission. In this chapter, the VFG in two-dimensional space is designed for the reference trajectory which is represented as implicit function. Then, VFG in three-dimensional space is designed for the reference trajectory which is represented as parametric function. The asymptotic stability and the finite-time stability of the VFG are analyzed using the Lyapunov-like approach for both two- and three-dimensional guidance algorithms with implicit and parameterized reference curves, respectively.

5.1 Vector Field Design in Two-Dimensional Space

Vector field is constructed as a linear combination of convergence vector and orthogonal vector of the gradient vector [75]. To find the vector components, predicted target trajectory should first be defined. The predicted trajectory of the target is given as an implicit function $f(x, y) = 0$ in two-dimensional space.

5.1.1 Convergence Vector and Traverse Vector

A convergence vector $\mathbf{w}_{\text{conv}}$ is defined by the gradient vector of the trajectory function $f(x, y)$ as

$$\mathbf{w}_{\text{conv}} = -\text{sign}(f)\frac{\nabla f}{\|\nabla f\|} \tag{5.1}$$

Because the gradient vector directs to the steepest ascent direction of the function, the convergence vector is also directed toward the target trajectory.

A traverse vector $\mathbf{w}_{\text{trav}}$ is defined as an orthogonal vector of the gradient vector of the trajectory function, ∇f. In a two-dimensional plane, the traverse vector can be obtained as

$$\mathbf{w}_{\text{trav}} = -\text{sign}(g)\frac{\nabla g}{\|\nabla g\|} \tag{5.2}$$

$$\nabla f^T \nabla g = 0 \tag{5.3}$$

where the orthogonal function $g(x, y)$ can be calculated by analytically integrating the orthogonal vector $\nabla g(x, y)$. For example, for a trajectory function $f(x, y) = -0.05(x^2 - 9x - 10) - y$, which is a second-order polynomial function, the corresponding orthogonal function is $g(x, y) = 10 \log(2x - 9) - y$. The trajectory function f, the orthogonal function g, and their gradient vectors are visualized in Fig. 5.1.

Now, the vector field vector $\mathbf{w}(x, y)$ is designed as a linear combination of the convergence vector, $\mathbf{w}_{\text{conv}}$, and the traverse vector, $\mathbf{w}_{\text{trav}}$, as

$$\mathbf{w} = k_f \mathbf{w}_{\text{conv}} + k_g \mathbf{w}_{\text{trav}} \tag{5.4}$$

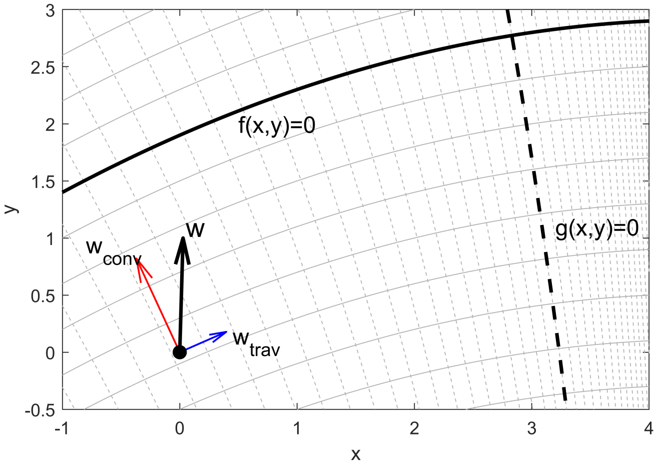


Figure 5.1: Visualization of $f_1(x, y)$, $g_1(x, y)$, and gradient vectors

where k_f and k_g are positive coefficients satisfying the condition $k_f^2 + k_g^2 = 1$, and therefore $\mathbf{w}$ is a unit vector.

5.2 Stability Analysis for Vector Field Represented as Implicit Function

5.2.1 Stability of the Vector Field with Constant Coefficients

Let us analyze that the vector field asymptotically converges to the specified reference trajectory represented as the implicit functions $f(x, y)$ and $g(x, y)$.

Let us consider a Lyapunov candidate function.

$$V(x, y) = \frac{1}{2}\left(f^2(x, y) + g^2(x, y)\right) \tag{5.5}$$

Assuming that the missile can instantly follow the field vector, the velocity vector can be written as follows.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = V_M \mathbf{w}(x, y) \tag{5.6}$$

Differentiating $V(x, y)$ with respect to time and substituting Eqs. (5.1) - (5.4) and (5.6) into the resulting equation yields

$$\begin{aligned}\dot{V}(x, y) &= f\dot{f} + g\dot{g} \\ &= f\nabla f^T \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + g\nabla g^T \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \\ &= -V_M k_f |f| \|\nabla f\| - V_M k_g |g| \|\nabla g\| < 0 \end{aligned} \tag{5.7}$$

The Lyapunov function $V(x, y)$ and the time derivative of $V(x, y)$ are zero only when both of the functions $f(x, y)$ and $g(x, y)$ are zero, and therefore the system converges to the point (x^*, y^*), where $f(x^*, y^*) = 0$ and $g(x^*, y^*) = 0$.

There exists only one point, where $f(x, y)$ and $g(x, y)$ are equal to zero. Finally, by the Lyapunov stability theorem, the vector field is asymptotically stable to the point (x^*, y^*), where $f(x^*, y^*) = g(x^*, y^*) = 0$ are satisfied. However, in actual engagement situation, the missile may not accurately and instantly follow the vector field direction, especially when the speed of the missile is fast and the direction of the vector field changes sharply. Therefore, time-varying coefficients k_f and k_g would be better than constant values, which will make the missile approach the reference trajectory smoothly.

In this section, the stability analysis for a given reference curve $f(x, y) = 0$ is performed. The vector field smoothly converges to the reference curve because $f(x, y)$ goes to zero, and $\dot{f}(x, y)$ also goes to zero by means of the varying coefficients k_f and k_g.

Let us consider a Lyapunov candidate function.

$$V = \frac{1}{2}\mathbf{s}^T \Psi \mathbf{s} \tag{5.8}$$

where

$$\mathbf{s} = [f \quad \dot{f}]^T \tag{5.9}$$

$$\Psi = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{5.10}$$

Note that P is a constant positive definite matrix, i.e., $p_{11} > 0$ and $p_{11}p_{22}-p_{12}^2 > 0$. An additional condition, $\text{sign}(p_{12}/p_{22}) < 0$, is required for the finite-time stability, which will be discussed in a later section. Differentiating the Lyapunov candidate function V with respect to time obtains

$$\dot{V} = p_{11}f\dot{f} + p_{12}\dot{f}^2 + \ddot{f}\left(p_{12}f + p_{22}\dot{f}\right) \tag{5.11}$$

Assume that the missile can ideally follow the vector field $\mathbf{w}(x, y)$ direction at the current position with a constant V_M. Then, the velocity components of the missile in XY-coordinates can be represented as follows,

$$\dot{X} = V_M k_g \tag{5.12}$$

$$\dot{Y} = V_M k_f \tag{5.13}$$

where the XY coordinate is a coordinate of which the X axis and the Y axis direct to the $\mathbf{w}_{\text{trav}}$ and $\mathbf{w}_{\text{conv}}$ directions, respectively, as shown in Fig. 5.2. The velocity of the missile aligned to the direction of $\mathbf{w}$ is described in Fig. 5.3.

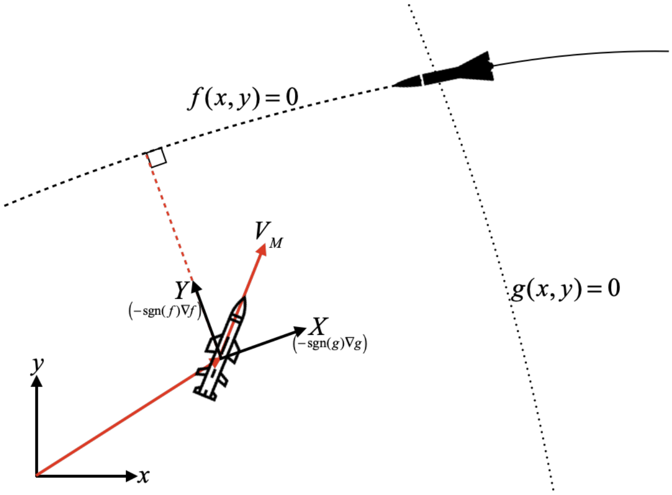


Figure 5.2: Coordinates in two dimensional engagement

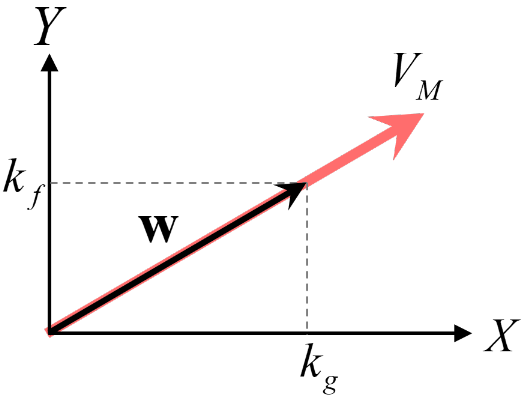


Figure 5.3: The velocity components in XY-coordinate

Now, the time derivative of the function $f(x, y)$ can be represented as follows,

$$\begin{aligned} \dot{f} &= \begin{bmatrix} f_X & f_Y \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = V_M \begin{bmatrix} f_X & f_Y \end{bmatrix} \begin{bmatrix} k_g \\ k_f \end{bmatrix} \\ &= V_M f_Y k_f \end{aligned} \tag{5.14}$$

where f_X and f_Y are the partial derivatives of f in XY-coordinates. Note that the X-axis is parallel to the tangent direction of the curve $f(x, y) = 0$ at any point where $f_X = 0$. The calculation procedure of $f_X, f_Y, f_{XX}, f_{XY}, f_{YX}$ and f_{YY} can be found in Appendix. Differentiating Eq. (5.14) with respect to time and using Eq. (5.13) in the resulting equation yields

$$\begin{aligned} \ddot{f} = \frac{d}{dt}(\dot{f}) = \frac{d}{dt}\left(V_M f_Y k_f\right) &= V_M(f_{YY}\dot{Y})k_f + V_M f_Y \dot{k}_f \\ &= V_M^2 f_{YY} k_f^2 + V_M f_Y \dot{k}_f \end{aligned} \tag{5.15}$$

Substituting Eq. (5.15) into Eq. (5.11) gives

$$\dot{V} = p_{11} f \dot{f} + p_{12} \dot{f}^2 + \left(p_{12} f + p_{22} \dot{f}\right)\left(V_M^2 f_{YY} k_f^2 + V_M f_Y \dot{k}_f\right) \tag{5.16}$$

By designing a proper k_f and $\dot{k}_f$ for Eq. (5.16), the asymptotic stable or finite-time stable vector field can be obtained.

5.2.2 Asymptotic Stable Vector Field Design

The time update rule for k_f can be obtained from Eq. (5.16) for $\dot{V}$ to have a specific form. For the asymptotic stability, let us propose the update rule of

k_f as follows,

$$\dot{k_f} = \frac{1}{V_M f_Y}\Big\{-V_M^2 f_{YY} k_f^2 + (p_{12} f + p_{22}\dot{f})^{-1}(-p_{11} f\dot{f} - p_{12}\dot{f}^2 - \rho_1 f^2 - \rho_2 \dot{f}^2)\Big\} \tag{5.17}$$

where ρ_1 and ρ_2 are positive scalars. Then, the time derivative of the Lyapunov function can be written as

$$\dot{V} = -\rho_1 f^2 - \rho_2 \dot{f}^2 \tag{5.18}$$

Because $\dot{V}$ is negative definite, the system is asymptotically stable for the point $(f, \dot{f}) = (0, 0)$. Note from Eq. (A.3) in Appendix that $f_Y \neq 0$ in general because it is easy to choose a proper $f(x, y)$ with nonzero gradient on the curve. In the definition of $\dot{k_f}$ in Eq. (5.17), $\dot{k_f}$ is not defined only when $(f, \dot{f}) = (0, 0)$, which is the situation when the missile is on the reference trajectory and maintaining that trajectory. By choosing $\dot{k_f} = 0$ on the point $(f, \dot{f}) = (0, 0)$, $\dot{k_f}$ can be shown as a continuous function near zero as follows,

$$\begin{aligned}
\lim_{f,\dot{f}\to 0} \dot{k_f} &= \lim_{f,\dot{f}\to 0} \frac{1}{V_M f_Y}\left(-V_M^2 f_{YY} k_f^2 + \frac{-p_{11} f\dot{f} - p_{12}\dot{f}^2 - \rho_1 f^2 - \rho_2 \dot{f}^2}{p_{12} f + p_{22}\dot{f}}\right) \\
&= \lim_{f,\dot{f}\to 0} \frac{1}{V_M f_Y}\left\{-\frac{\rho_1}{p_{12}} f - \frac{p_{12}+\rho_2}{p_{22}}\dot{f} + \Big(-p_{11} + \frac{\rho_1 p_{22}}{p_{12}} + \frac{p_{12}(p_{12}+\rho_2)}{p_{22}}\Big)\frac{f\dot{f}}{p_{12} f + p_{22}\dot{f}}\right\} \\
&= \lim_{f,\dot{f}\to 0} \frac{1}{V_M f_Y}\left\{-p_{11} + \frac{\rho_1 p_{22}}{p_{12}} + \frac{p_{12}(p_{12}+\rho_2)}{p_{22}}\right\}\frac{1}{p_{12}/\dot{f} + p_{22}/f} \\
&= 0
\end{aligned} \tag{5.19}$$

Note that $\lim_{\dot{f}\to 0}(k_f) = 0$ from Eq. (5.14). Therefore, by choosing $k_f = \dot{k_f} = 0$ on the reference trajectory, the missile will maintain on the reference curve.

5.2.3 Finite-time Convergent Vector Field Design

Let us analyze the finite-time convergence of the system. Now, let us propose the following k_f update rule instead of Eq. (5.17),

$$\dot{k_f} = \frac{1}{V_M f_Y}\Big(- V_M^2 f_{YY} k_f^2 + (p_{12} f + p_{22}\dot{f})^{-1}(-p_{11} f \dot{f} - p_{12}\dot{f}^2 - \alpha V^{\lambda})\Big) \quad (5.20)$$

where $f \neq 0$ and $\dot{f} \neq 0$. Substituting Eq. (5.20) into (5.16) yields

$$\dot{V} = -\alpha V^{\lambda} \quad (5.21)$$

The condition for the finite-time convergence is satisfied with the proposed update rule in Eq. (5.20). Note that there is a singular case when $(p_{12}f + p_{22}\dot{f}) = 0$. To deal with the singular case, similar to Eq. (5.19), the update rule $\dot{k_f}$ approaching $(f, \dot{f}) \to (0, 0)$ can be calculated as follows:

$$\begin{aligned} \lim_{f,\dot{f}\to 0} \dot{k_f} &= \lim_{f,\dot{f}\to 0} \frac{1}{V_M f_Y} \frac{-\alpha V^{\lambda}}{p_{12} f + p_{22}\dot{f}} \\ &= \lim_{f,\dot{f}\to 0} \frac{1}{V_M f_Y} \frac{-\alpha (p_{11} f^2 + 2p_{12} f\dot{f} + p_{22}\dot{f}^2)^{\lambda}}{2^{\lambda}(p_{12} f + p_{22}\dot{f})} \end{aligned} \quad (5.22)$$

Note that as $f, \dot{f} \to 0$, $(p_{11} f^2 + 2p_{12} f\dot{f} + p_{22}\dot{f}^2)$ in Eq. (5.22) approaches zero and smaller than 1. Therefore, the following inequality holds.

$$\left[\lim_{f,\dot{f}\to 0} |\dot{k_f}|\right]_{\lambda=1} < \lim_{f,\dot{f}\to 0} |\dot{k_f}| \leq \left[\lim_{f,\dot{f}\to 0} |\dot{k_f}|\right]_{\lambda=\underline{\lambda}} \quad (5.23)$$

where $\underline{\lambda} \leq \lambda < 1$.

First, let us consider the $\lambda = 1$ case. In this case, as like in Eq. (5.19), it can be readily shown that the limit goes to zero. It can be shown that the limit of the median term in Eq. (5.23) is zero by showing that the limit of the right term in Eq. (5.23) for certain $\underline{\lambda}$ is zero.

Now, let us consider the polar coordinate variables r, θ for $f, \dot{f}$ as follows,

$$f = r\cos\theta, \quad \dot{f} = r\sin\theta \tag{5.24}$$

That is, the value of $(f, \dot{f})$ at an arbitrary (x, y) point can be represented by using r and θ. In the $(f, \dot{f})$-plane, when the missile approaches the curve $f(x, y) = 0$ smoothly, the system approaches the origin along a certain trajectory satisfying the dynamic relationship between f and $\dot{f}$. Note that any trajectories approaching the origin, i.e., $f, \dot{f} \to 0$, can be represented as $r \to 0$. Therefore, Eq. (5.22) can be rewritten as follows,

$$\begin{aligned}\lim_{f,\dot{f}\to 0} \dot{k}_f &= \lim_{r\to 0} \frac{1}{V_M f_Y 2^\lambda} \frac{-\alpha r^{2\lambda}(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^\lambda}{r(p_{12}\cos\theta + p_{22}\sin\theta)} \\ &= \lim_{r\to 0} \frac{-\alpha}{V_M f_Y 2^\lambda} \left(\frac{r^{2\lambda_0}(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^{\lambda_0}}{\Big(r(p_{12}\cos\theta + p_{22}\sin\theta)\Big)^{\lambda_0/\lambda}} \right)^{\lambda/\lambda_0}\end{aligned} \tag{5.25}$$

where λ_0 is an arbitrary scalar. Let us consider a special case $\lambda = \frac{2}{3}$ and $\lambda_0 = 2$.

$$\begin{aligned}\lim_{f,\dot{f}\to 0} \dot{k}_f \Big|_{\lambda=2/3,\lambda_0=3} &= \lim_{r\to 0} \frac{-\alpha}{V_M f_Y 2^\lambda} \left(\frac{r^4(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^2}{r^3(p_{12}\cos\theta + p_{22}\sin\theta)^3} \right)^{\frac{1}{3}} \\ &= \lim_{r\to 0} \frac{-\alpha}{V_M f_Y 2^\lambda} \left(\frac{r(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^2}{(p_{12}\cos\theta + p_{22}\sin\theta)^3} \right)^{\frac{1}{3}} \\ &= 0\end{aligned} \tag{5.26}$$

Note that the equation still has an indeterminate form, that is, the denominator is zero when the following condition holds as $r \to 0$.

$$p_{12}\cos\theta + p_{22}\sin\theta = 0 \tag{5.27}$$

However, it can be shown that this case cannot happen with a proper selection of p_{12} and p_{22}. Using Eqs. (5.24) and (5.27), we have

$$\frac{\dot{f}}{f} = \tan\theta = -\frac{p_{12}}{p_{22}} \tag{5.28}$$

Meanwhile, f and $\dot{f}$ always have different signs because the convergent vector is designed in the direction of $f(x,y) = 0$, which is the negative gradient direction, or $\text{sign}(\dot{f}/f) = -1$. Therefore, if the sign of $\frac{p_{12}}{p_{22}}$ is designed to be negative, then the condition (5.27) is not satisfied, and consequently the limit of $\dot{k}_f$ for $\lambda_0 = 2/3$ is zero.

Finally, using Eq. (5.23) and Eq. (5.26), the limit of k_f for $\frac{2}{3} \leq \lambda < 1$ is zero as $f, \dot{f} \to 0$. By defining $\dot{k}_f = 0$ at $(f, \dot{f}) = (0,0)$, $\dot{k}_f$ can be shown as a continuous function near the origin. That is, by choosing $k_f = \dot{k}_f = 0$ at the point $(f, \dot{f}) = (0,0)$, the missile will stay on the reference curve and finally intercept the target. Note that the range of λ can be widened analytically when the pair of integers $(\lambda_0, \lambda_0/\lambda)$ is properly chosen. In this study, for the sake of brevity, only the simple case $(\lambda_0, \lambda_0/\lambda) = (2,3)$ is considered.

Theorem 5.1. *Consider $\dot{k}_f$ defined in Eq. (5.22) and its limit in Eq. (5.25). When $\lambda = \frac{\lambda_0}{2\lambda_0 - 1}$ for any positive odd interger $\lambda_0 \geq 3$, then* $\lim_{f,\dot{f}\to 0} \dot{k}_f = 0$.

Proof. Substituting $\lambda = \frac{\lambda_0}{2\lambda_0 - 1}$ into Eq. (5.25) gives,

$$
\begin{aligned}
\lim_{f,\dot{f}\to 0} \dot{k_f} &= \lim_{r\to 0} \frac{-\alpha}{V_M f_Y 2^\lambda} \left(\frac{r^{2\lambda_0}(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^{\lambda_0}}{r^{2\lambda_0 - 1}(p_{12}\cos\theta + p_{22}\sin\theta)^{2\lambda_0 - 1}} \right)^{\frac{1}{2\lambda_0 - 1}} \\
&= \lim_{r\to 0} \frac{-\alpha}{V_M f_Y 2^\lambda} \left(\frac{r(p_{11}\cos^2\theta + 2p_{12}\cos\theta\sin\theta + p_{22}\sin^2\theta)^{\lambda_0}}{(p_{12}\cos\theta + p_{22}\sin\theta)^{2\lambda_0 - 1}} \right)^{\frac{1}{2\lambda_0 - 1}} \\
&= 0
\end{aligned}
\tag{5.29}
$$

□

Figure 5.4 shows the value of the numerator and the denominator in Eq. (5.25) near the origin in $(f, \dot{f})$-space for specific values of $p_{11} = p_{22} = 1$, $p_{12} = -0.5$, and $\lambda = 2/3$. Note that the limit of $\dot{k_f}$ from Eq. (5.25) is not zero only when the trajectory in $(f, \dot{f})$ approaches the origin along the dashed line. However, that is not the case because the signs of f and $\dot{f}$ are different. The guaranteed settling time for the system is $T_s \leq V(0)^{1-\lambda}/\alpha(1-\lambda)$. The block diagram of the system summarizing the update law of the coefficient k_f is shown in Fig. 5.5.

Parameter Design without Time-update Law

The Lyapunov candidate function V in Eq. (5.8), which contains f and $\dot{f}$, is considered to smoothly converge to the curve $f(x, y) = 0$. Note that the time-derivative of $\dot{f}$ appears when differentiating V with respect to time, and the dynamic update rule for $\dot{k_f}$ is derived using the resulting equation. However, there may exist some trajectories that make the update rule indeterminate, and additional analysis for these cases were performed to address it in the previous section.

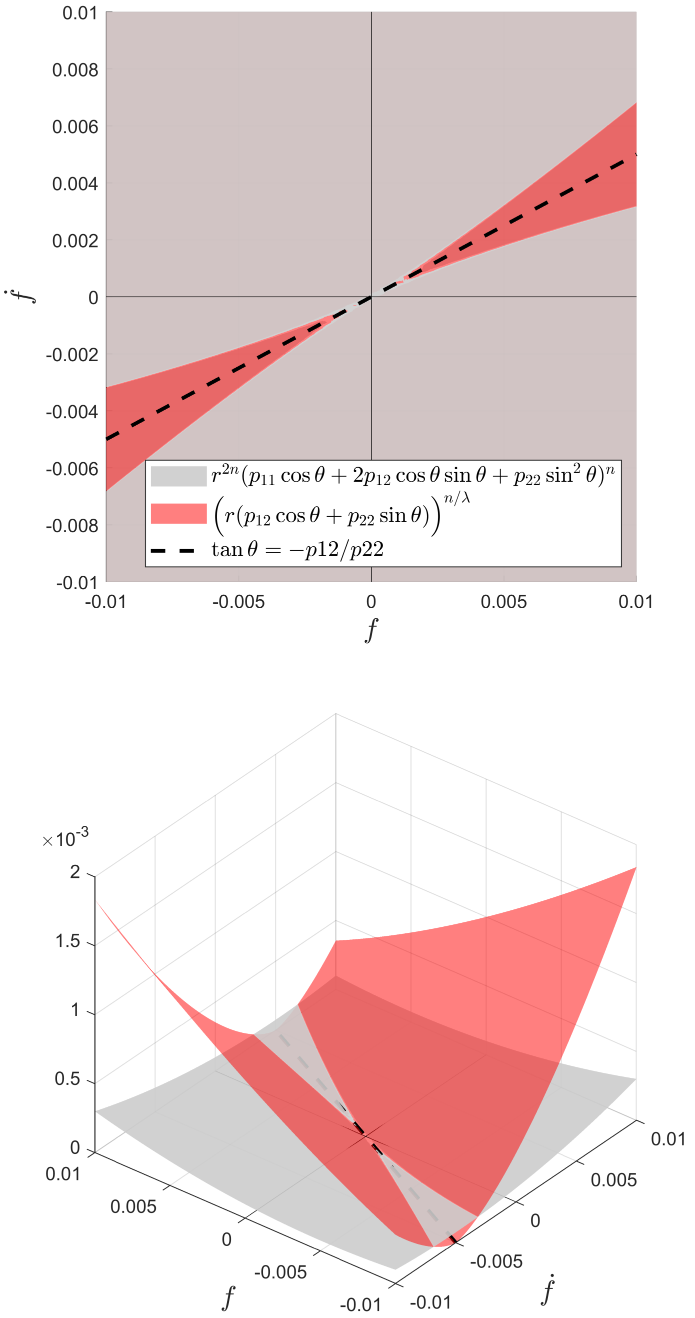


Figure 5.4: Visualization of the limit value near the origin

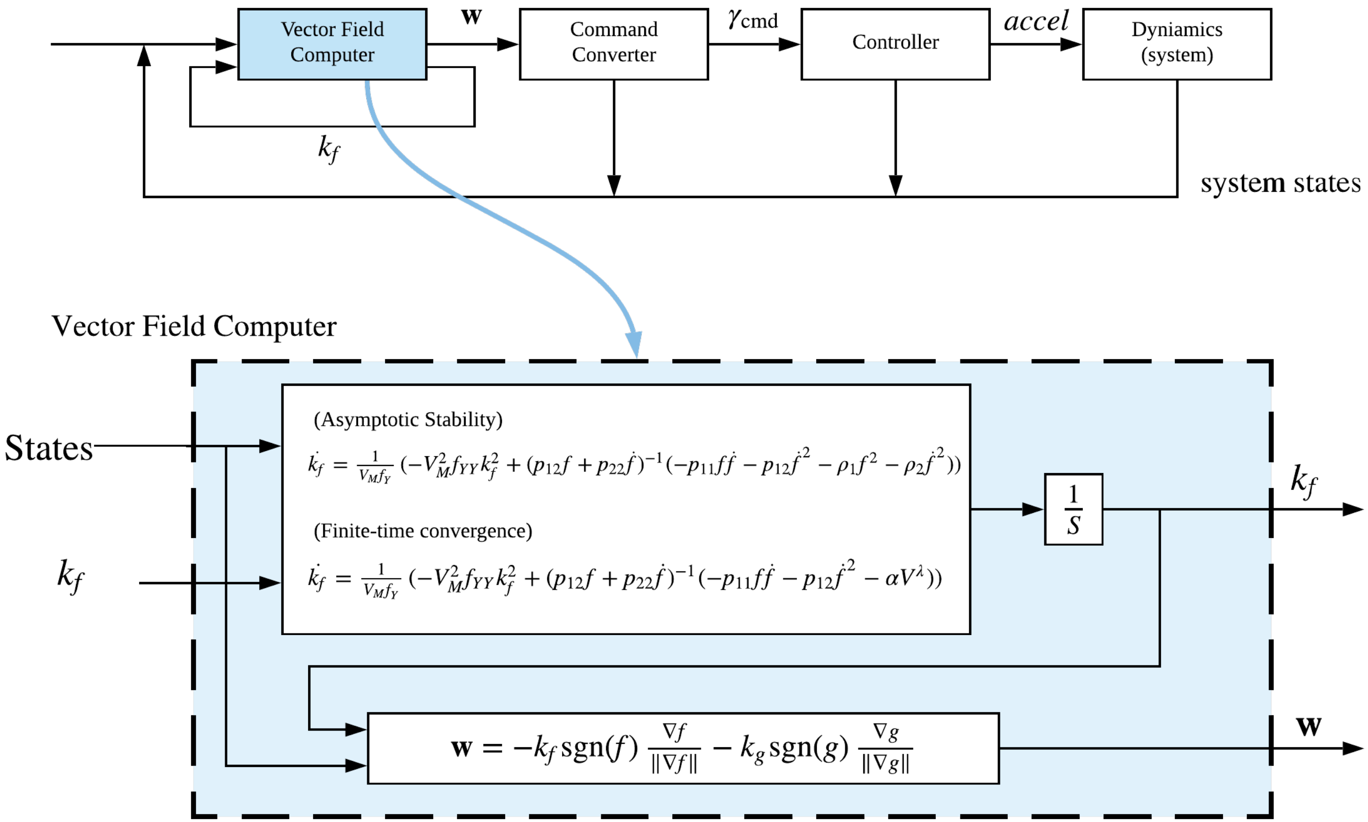

Vector Field Computer
w
Command Converter
γ_{cmd}
Controller
$accel$
Dyniamics (system)
k_f
system states
Vector Field Computer
States
k_f
(Asymptotic Stability)
$\dot{k}_f = \frac{1}{V_M f_Y}(-V_M^2 f_{YY} k_f^2 + (p_{12} f + p_{22} \dot{f})^{-1}(-p_{11} f \dot{f} - p_{12} \dot{f}^2 - \rho_1 f^2 - \rho_2 \dot{f}^2))$
(Finite-time convergence)
$\dot{k}_f = \frac{1}{V_M f_Y}(-V_M^2 f_{YY} k_f^2 + (p_{12} f + p_{22} \dot{f})^{-1}(-p_{11} f \dot{f} - p_{12} \dot{f}^2 - \alpha V^{\lambda}))$
$\frac{1}{S}$
k_f
$\mathbf{w} = -k_f \,\text{sgn}(f) \frac{\nabla f}{\|\nabla f\|} - k_g \,\text{sgn}(g) \frac{\nabla g}{\|\nabla g\|}$
w

Figure 5.5: Block diagram of the VFG system

In this section, a finite-time convergent gain, k_f, without the dynamic update rule is proposed. The gain k_f can be designed without the inverse term, and therefore no indeterminate case analysis is required. Additionally, a feasible region for the parameter pair (α, λ) could be obtained with this approach, which provides a guideline for selecting proper parameters α and λ. Consider the following Lyapunov candidate function, V_1.

$$V_1 = f^2 \tag{5.30}$$

Similar to Eq. (5.7), we have

$$\begin{aligned}\dot{V}_1 = 2f\dot{f} &= 2f\nabla f^T \left(-V_M k_f \text{sign}(f)\frac{\nabla f}{\|\nabla f\|}\right) \\ &= -2V_M k_f |f| \|\nabla f\| \end{aligned} \tag{5.31}$$

It may be thought that the vector field does not smoothly converge to the curve because the Lyapunov candidate function does not depend on $\dot{f}$. However, it can be shown that f and $\dot{f}$ go to zero as the interceptor missile approaches the trajectory by selecting a proper value of k_f.

For the finite-time convergence, the following inequality should be satisfied.

$$\dot{V}_1 + \alpha V_1^\lambda = -2V_M k_f |f| \|\nabla f\| + \alpha f^{2\lambda} \le 0 \tag{5.32}$$

This inequality condition can be rewritten with respect to the field vector gain, k_f, as follows,

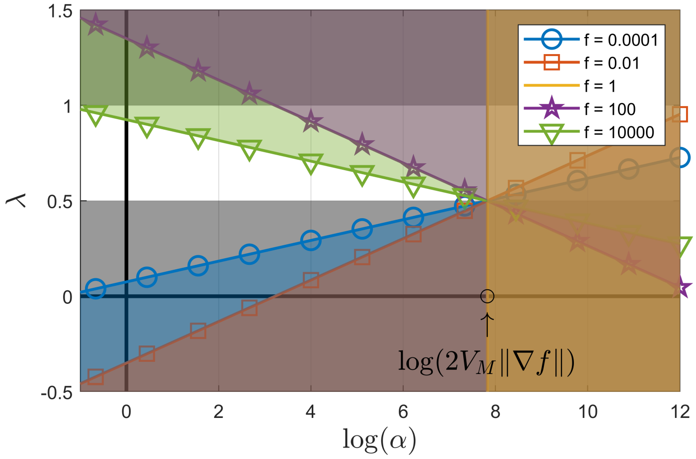
$$k_{f,\text{lower}} = \frac{\alpha |f|^{2\lambda-1}}{2V_M \|\nabla f\|} \le k_f \le 1 \tag{5.33}$$

The vector field coefficients k_f can be determined to satisfy Eq. (5.33). First, let us consider the case that $k_f = 1$. In this case, the field vector $\mathbf{w}$ directs towards the curve, which is the fastest strategy to arrive the reference curve. However, in this strategy, the vector field may not approach the curve smoothly, which may result in chattering phenomena of the lateral acceleration command near the curve. Second, let us consider $k_f = k_{f,\text{lower}}$ as an alternative. In this case, when $\lambda > 0.5$, as $f \to 0$, $k_f \to 0$, and consequently, $\dot{f} \to 0$. Finally, the missile converges to the curve smoothly before a specific settling time, which is determined by the parameters α and λ. On the other hand, if $k_{f,\text{lower}}$ is larger than 1, it is not possible to converge in the specific settling time determined by the parameters. Therefore, a proper pair of parameters should be chosen for the missile to approach the trajectory smoothly in the specified settling time in Eq. (2.54).

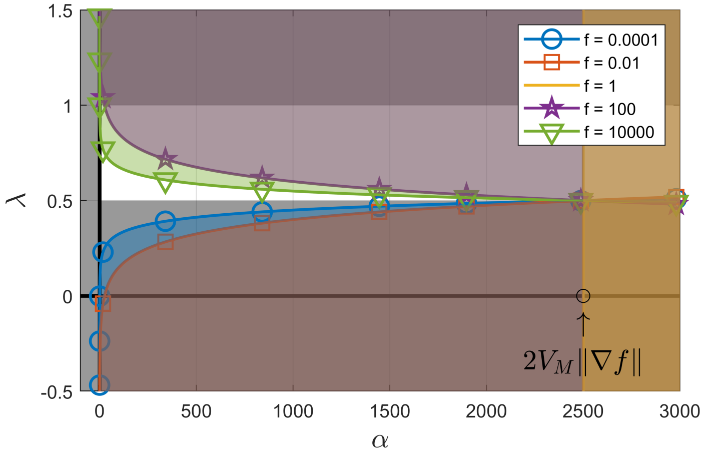
Using Eq. (5.33) and the condition $0.5 < \lambda < 1$ of Theorem 2.2, the feasible region for the parameters α and λ can be obtained. By taking the log function to Eq. (5.33), the feasible area for the parameters α and λ can be obtained as

$$\begin{cases} \lambda \leq \frac{1}{2}\Big\{ -\frac{1}{\log|f|}\log\alpha + \frac{\log(2V_M\|\nabla f\|)}{\log|f|} + 1 \Big\} & \text{when } |f| > 1 \\ \lambda \geq \frac{1}{2}\Big\{ -\frac{1}{\log|f|}\log\alpha + \frac{\log(2V_M\|\nabla f\|)}{\log|f|} + 1 \Big\} & \text{when } |f| < 1 \\ \alpha \leq 2V_M\|\nabla f\| & \text{when } |f| = 1 \end{cases} \tag{5.34}$$

The feasible parameter regions for various f values are depicted in Fig. 5.6. The points, $\log(2V_M\|\nabla f\|)$ and $2V_M\|\nabla f\|$, are also shown, which will be demonstrated in the numerical simulation, Chapter 6. The gray regions are from the conditions $0.5 < \lambda < 1$ and $\alpha > 0$, and the colored regions are from Eq. (5.34)



((a))



((b))

Figure 5.6: Feasible area for parameters α and λ

with specific f values. The remaining white region represents the feasible parameter region.

Because $f(t)$ decreases as the missile approaches the reference curve, the feasible region for the parameters (noncolored region) changes. Let us assume that $\log(2V_M\|\nabla f\|)$ is constant and that only the $|f|$ value is changing from $|f| \gg 1$ to zero. If the missile is far from the curve and $|f| > 1$, then the inclination is negative as shown in Fig. 5.6(a). As $|f|$ decreases, the inclination approaches to negative infinity until $|f| \to 1$. When $|f| = 1$, the dividing line becomes vertical. As $|f|$ continues to decrease from 1 to zero, the inclination decreases from infinity to zero. That is, the dividing line rotates clockwise as $|f| \to 0$. Figure 5.6(b) shows the same feasible region of Fig. 5.6(a), except that the horizontal axis is α. Note that as the missile is close to the curve, the dividing line rotates clockwise and the feasible area of the parameters is widened, as shown in Fig. 5.6(a). Especially, when $|f| < 1$, the colored area is almost submerged in the gray area ($\lambda \leq 0.5$), as shown in Fig. 5.6(b). It can be also seen that when the speed of the missile V_M is faster, the feasible region is enlarged.

5.3 Vector Field Design in Three-Dimensional Space

In this section, the vector field design in the previous section is expanded to three-dimensional space. The basic concept of the construction of the vector field in 3D space is the same with the construction of the vector field in 2D space. Another different thing is that the reference curve is not given as implicit functions, but given as parameterized curves.

Note that a practical reference curve in 2D space can be represented with a single implicit function without any singular point, but it requires two implicit functions for the representation of a curve in 3D space. That is, a curve in 3D space is represented as the intersection between two surfaces that are defined by the implicit functions. Therefore, it is convenient to consider the reference curve in 3D space as a parameterized curve. However, as in 2D space, the vector field for VFG in 3D space is defined as a linear combination of the convergence vector and the traverse vector. Note that it is also possible to represent a curve in 2D space as a parametric function, and therefore any of both representations is possible to design in 2D or 3D space.

Because most of the methodologies for the analysis is the same with that of the previous section, detailed explanations are omitted and briefly summarized in this section.

5.3.1 Convergence Vector and Traverse Vector

Assumption 5.1. *For an arbitrary missile position* $\mathbf{P}(t) = [x(t), y(t), z(t)]^T$ *at time* t*, the nearest point* $\mathbf{c}(\tau^*)$ *on the reference curve* $\mathbf{c}(\tau)$ *is known.*

Assumption 5.2. *The missile can instantly follow the field vector. That is, the direction of the missile's velocity vector is the same with the direction of the vector field* $\mathbf{w}$.

Assumption 5.3. *The speed of the missile V_m is constant.*

The three-dimensional geometry of the missile position $\mathbf{P}(t)$ and the reference curve represented as parametric curve $\mathbf{c}(\tau)$ is shown in Fig. 5.7. The vector field $\mathbf{w}$ is defined as the linear combination of the two vector field components, $\mathbf{w}_c$ and $\mathbf{w}_t$, as

$$\mathbf{w} = k_f \mathbf{w}_c + k_g \mathbf{w}_t \tag{5.35}$$

The convergence vector $\mathbf{w}_c$ is a unit vector which has the same direction with the range vector $\mathbf{r} = \mathbf{c}(\tau^*) - \mathbf{P}(t)$. The traverse vector $\mathbf{w}_t$ is a unit vector which is tangent to the curve at the point $\mathbf{c}(\tau^*)$. From the two possible choices of the tangent direction, the direction that τ increases is chosen as the traverse vector's direction. The normal vector $\mathbf{w}_n = \mathbf{w}_c \times \mathbf{w}_t$ is defined by the cross product of the convergence vector and the traverse vector. Assumptions 5.1-5.3 are used for the VFG with respect to parameterized curve.

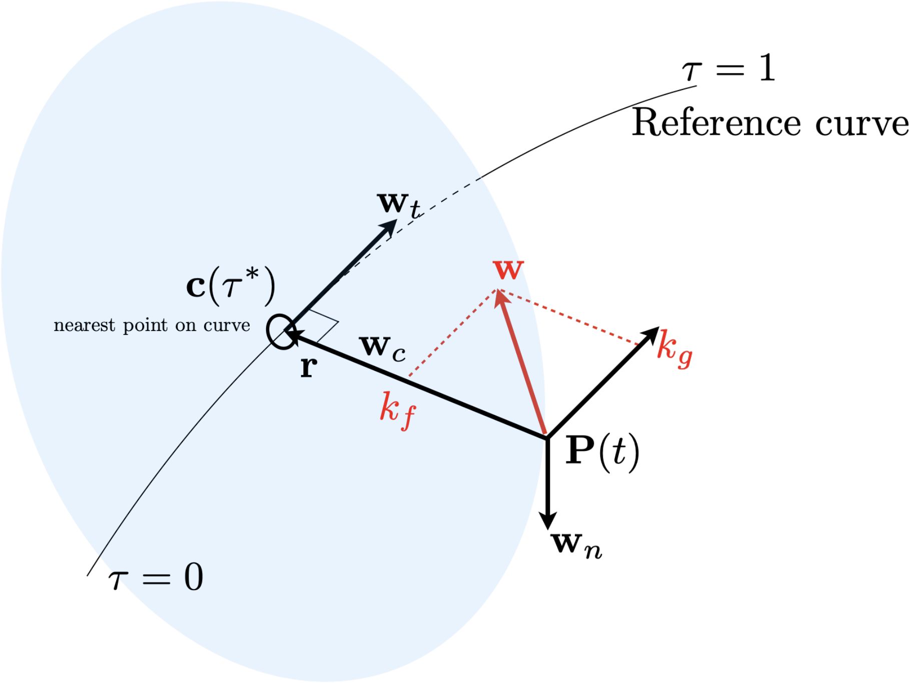


Figure 5.7: Visualization of vector field components in 3D space

5.4 Stability Analysis for Vector Field Represented as Parametric Function

5.4.1 Stability of the Vector Field with Constant Coefficients

Let us define the Lyapunov function V as follows,

$$V \triangleq \frac{1}{2}\mathbf{r}^T\mathbf{r} \tag{5.36}$$

Differentiating V with respect to time yields

$$\begin{aligned}\dot{V}(\mathbf{r}) &= \mathbf{r}^T\dot{\mathbf{r}} \\ &= -V_M k_f \mathbf{r}^T \mathbf{w}_c \\ &= -V_M k_f \mathbf{r}^T \frac{\mathbf{r}}{\|\mathbf{r}\|} \\ &= -V_M k_f \|\mathbf{r}\| \leq 0\end{aligned} \tag{5.37}$$

where

$$\dot{\mathbf{r}} = -V_M k_f \mathbf{w}_{\text{conv}} \tag{5.38}$$

Equation (5.38) is obtained from Assumption 5.2. The Lyapunov function $V(\mathbf{r})$ and the time derivative of $V(\mathbf{r})$ are zero only when $r \triangleq \|\mathbf{r}\| = 0$. The vector field is asymptotically stable to the curve ($r = 0$) by the Lyapunov stability theorem.

5.4.2 Asymptotic Stable Vector Field Design

In this section, the asymptotic stable vector field is designed. The asymptotic stable vector field converges to the reference curve smoothly because not only r but also $\dot{r}$ goes to zero by means of the varying coefficients k_f and k_g.

Let us consider a following Lyapunov candidate function.

$$V \triangleq \frac{1}{2}\mathbf{s}^T \Psi \mathbf{s} \tag{5.39}$$

where

$$\mathbf{s} = [r \quad \dot{r}]^T \tag{5.40}$$

$$\Psi = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{5.41}$$

Note that Ψ is a constant positive definite matrix, i.e., $p_{11} > 0$ and $p_{11}p_{22} - p_{12}^2 > 0$. Differentiating V with respect to time yields

$$\dot{V} = p_{11} r \dot{r} + p_{12} \dot{r}^2 + \ddot{r}(p_{12} r + p_{22} \dot{r}) \tag{5.42}$$

Meanwhile, the time derivative of the distance to the curve, $\dot{r}$, is given as

$$\dot{r} = -V_M \frac{k_f}{\sqrt{k_f^2 + k_g^2}} = -V_M k_f \tag{5.43}$$

Differentiating Eq. (5.43) yields

$$\ddot{r} = -\dot{V}_M k_f - V_M \dot{k}_f = -V_M \dot{k}_f \tag{5.44}$$

Note that the speed of the missile V_m is constant from Assumption 5.3. Substituting Eq. (5.44) into Eq. (5.42) gives,

$$\dot{V} = p_{11} r \dot{r} + p_{12} \dot{r}^2 - V_m \dot{k}_f (p_{12} r + p_{22} \dot{r}) \tag{5.45}$$

Now, the time update rule for k_f can be derived using Eq. (5.45) such that $\dot{V}$ has a specific form. Let us propose the update rule of k_f for the asymptotic stability as follows,

$$\dot{k}_f = \frac{1}{V_M}(p_{12}r + p_{22}\dot{r})^{-1}(p_{11}r\dot{r} + p_{12}\dot{r}^2 + \rho_1 r^2 + \rho_2 \dot{r}^2) \tag{5.46}$$

where ρ_1 and ρ_2 are positive scalars. Then, the time derivative of the Lyapunov function can be written as

$$\dot{V} = -\rho_1 r^2 - \rho_2 \dot{r}^2 \tag{5.47}$$

Because $\dot{V}$ is negative definite, the system is asymptotically stable for the point $(r, \dot{r}) = (0, 0)$. Note that the time update law in Eq. (5.46) is not defined when $p_{12}r + p_{22}\dot{r} = 0$, that is, the missile is on the reference curve. By choosing $\dot{k}_f = 0$ on the point $(r, \dot{r}) = (0, 0)$, $\dot{k}_f$ can be shown as a continuous function near zero. A detailed analysis is similar to that of the 2D case in Eq. (5.19).

5.4.3 Finite-time Convergent Vector Field Design

Let us propose a following k_f update rule.

$$\dot{k}_f = \frac{1}{V_M}(p_{12}r + p_{22}\dot{r})^{-1}(p_{11}r\dot{r} + p_{12}\dot{r}^2 + \alpha V^{\lambda}) \tag{5.48}$$

Then, the time derivative of V becomes

$$\dot{V} = -\alpha V^{\lambda} \tag{5.49}$$

Note that there exists a singular condition of $(p_{12}r + p_{22}\dot{r}) = 0$. Dealing with this singular case is similar to that of the 2D case in Eq. (5.22). Finally, choosing

$k_f = \dot{k}_f = 0$ at the point $(r, \dot{r}) = (0, 0)$ completes a continuous update rule for k_f. The missile will stay on the reference curve if once converged to the reference curve.

Parameter Design without Time-update Law

As given in Eq. (5.33), the vector field parameter k_f can also be designed without time-update law for parameteric reference curve. In this case, no indeterminate case analysis is required and a feasible region for the parameter pair (α, λ) can be obtained with this approach. Let us define Lyapunov function $V(r)$ as follows,

$$V \triangleq r^2 \tag{5.50}$$

Time derivative of $V(r)$ is,

$$\dot{V}(r) = -2V_M k_f r \tag{5.51}$$

For the finite-time convergence, the following inequality should be satisfied.

$$\dot{V} + \alpha V^\lambda = -2V_M k_f r + \alpha r^{2\lambda} \leq 0 \tag{5.52}$$

Rewriting the inequality condition with respect to k_f gives

$$k_{f,\text{lower}} = \frac{\alpha r^{2\lambda-1}}{2V_M} \leq k_f \leq 1 \tag{5.53}$$

Similar to that of the 2D case, choosing $k_f = k_{f,\text{lower}}$ yields a smooth convergence to the reference curve when $\lambda > 0.5$. If $k_{f,\text{lower}}$ is larger than 1, it is not possible to converge in the specific settling time determined by the parameters (α, λ).

5.5 Summary

In this chapter, vector field-based guidance algorithm for missiles with constant speed was addressed. Reference path given for the missile to follow was represented as an implicit function form or in a parametric curve form. Missile is guided to converge to the reference path when it is away from the reference path, and follow the path after the missile approaches the path.

The guidance algorithm can be realized in both two- and three-dimensional space. The asymptotic and finite-time convergence attributes were analyzed and discussed through Lyapunov approach. The stability analysis for the reference path represented as an implicit function was discussed in two-dimensional space, and the stability analysis for the reference path represented as a parameterized curve was discussed in three-dimensional space.

Chapter 6

Numerical Simulation

In this Chapter, numerical simulations are performed to demonstrate effectiveness of the proposed method. The OOTS technique from Chapter 4 is utilzed to obtain an optimal trajectory for the outer-loop slow-scale system of the missile. The obtained trajectory is considered as the reference path for the VFG proposed in Chapter 5. The time-scale separation technique discussed in Chapter 3 allows designers to design the inner-loop and the outer-loop commands separately.

6.1 Optimal Trajectory Design

The outer-loop equations of motion, Eq. (3.1)-(3.3), are considered. The state vector $\mathbf{x}(t)$ and input vector $\mathbf{u}(t)$ are defined as follows,

$$\mathbf{x}(t) = [x, y, z]^T \tag{6.1}$$

$$\mathbf{u}(t) = [\gamma, \chi]^T \tag{6.2}$$

Note that the considerted system is nonlinear MIMO system. Let us define the output variables $h(\mathbf{x})$ as follows.

$$h(\mathbf{x}) = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} y \\ -z \end{bmatrix} \tag{6.3}$$

Differentiating the output variables with respect to time gives

$$\begin{aligned} \frac{d}{dt}h_1(\mathbf{x}) &= \dot{y} = V_m \cos\gamma \sin\chi \\ \frac{d}{dt}h_2(\mathbf{x}) &= -\dot{z} = V_m \sin\gamma \end{aligned} \tag{6.4}$$

Because the relative degrees are $\rho_1 = 1$ and $\rho_2 = 1$, the total relative degree is $\rho = \rho_1 + \rho_2 = 2$ and is smaller than the number of state variables, 3. Therefore, there exists an internal state variable.

6.1.1 Control Input Design

The error state varaibles $\mathbf{z}$ can be defined as follows,

$$\mathbf{z} = [z_1, z_2]^T = \begin{bmatrix} h_1(\mathbf{x}) - p_1(\tau) \\ h_2(\mathbf{x}) - p_2(\tau) \end{bmatrix} \tag{6.5}$$

The corresponding error dynamic equation can be written as

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} V_m \cos\gamma \sin\chi - \mu p_1' \\ V_m \sin\gamma - \mu p_2' \end{bmatrix} \tag{6.6}$$

The control inputs γ, χ are designed to satisfy the following equation.

$$
\begin{aligned}
\sin\gamma &\triangleq \frac{1}{V_m}\left(\mu p_2' - k_2 z_2\right) \\
\cos\gamma\sin\chi &\triangleq \frac{1}{V_m}\left(\mu p_1' - k_1 z_1\right)
\end{aligned}
\tag{6.7}
$$

Then, Eq. (6.6) becomes

$$
\dot{\mathbf{z}} = -[k_1 \quad k_2]\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = -K\mathbf{z}
\tag{6.8}
$$

where $K = [k_1, k_2]$ is a control gain matrix with positive scalars k_1 and k_2.

6.1.2 State and Input Parameterization

Based on Assumption 4.1, the state parameterization is obtained as follows,

$$
\begin{aligned}
\bar{y}(\tau) &= p_1(\tau) \\
\bar{z}(\tau) &= -p_2(\tau)
\end{aligned}
\tag{6.9}
$$

The input parameterization is obtained as follows,

$$
\begin{aligned}
\bar{\gamma}(\tau) &= \arcsin\left(\frac{\mu p_2'(\tau)}{V_m}\right) && \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\
\bar{\chi}(\tau) &= \arcsin\left(\frac{\mu p_1'(\tau)}{V_m\cos\bar{\gamma}(\tau)}\right) && \in [-\frac{\pi}{2}, \frac{\pi}{2}]
\end{aligned}
\tag{6.10}
$$

The state variables y and z are parameterized as $\bar{y}$ and $\bar{z}$, and inputs γ, χ are also parameterized as $\bar{\gamma}$ and $\bar{\chi}$.

6.1.3 Boundary Conditions

Let us consider the boundary conditions for states $(\bar{y}, \bar{z})$, internal state $(\bar{x})$, and input $(\bar{\gamma}, \bar{\chi})$. The boundary conditions for the states are given as follows,

$$\begin{aligned} \bar{y}(0) = y(0) = y_0, \quad \bar{y}(1) = y(t_f) = y_f \\ \bar{z}(0) = z(0) = z_0, \quad \bar{z}(1) = z(t_f) = z_f \end{aligned} \tag{6.11}$$

The parametric curves $p_1(\tau)$ and $p_2(\tau)$ are Bezier curves that can be represented as

$$p_1(\tau) = \sum_{v=0}^{N_1} \binom{N_1}{v} \tau^v (1-\tau)^{N_1 - v} \mathbf{P}_v^1 \tag{6.12}$$

$$p_2(\tau) = \sum_{v=0}^{N_2} \binom{N_2}{v} \tau^v (1-\tau)^{N_2 - v} \mathbf{P}_v^2 \tag{6.13}$$

where N_1 and N_2 are degrees, and $\mathbf{P}_v^1$ and $\mathbf{P}_v^2$ are the control points of each curves. From Eq. (6.9), the control points can be determined to satisfy the boundary conditions as

$$\bar{y}(0) = p_1(0) = \mathbf{P}_0^1 = y_0, \qquad \bar{y}(1) = p_1(1) = \mathbf{P}_{N_1}^1 = y_f \tag{6.14}$$

$$\bar{z}(0) = -p_2(0) = -\mathbf{P}_0^2 = z_0, \qquad \bar{z}(1) = -p_2(1) = -\mathbf{P}_{N_2}^2 = z_f \tag{6.15}$$

Note that the first $(\mathbf{P}_0^1, \mathbf{P}_0^2)$ and the last $(\mathbf{P}_{N_1}^1, \mathbf{P}_{N_2}^2)$ control points are determined by the boundary conditions of the state variables.

The boundary conditions for the inputs are given as follows,

$$\begin{aligned} \bar{\gamma}(0) = \gamma(0) = \gamma_0, \quad \bar{\gamma}(1) = \gamma(t_f) = \gamma_f \\ \bar{\chi}(0) = \chi(0) = \chi_0, \quad \bar{\chi}(1) = \chi(t_f) = \chi_f \end{aligned} \tag{6.16}$$

The derivative of the parametric curves $p_1(\tau)$ and $p_2(\tau)$ can be represented as

$$\begin{aligned} p_1'(\tau) = \sum_{v=0}^{N_1-1} & \binom{N_1}{v+1}(v+1)\tau^v(1-\tau)^{N_1-v-1}\mathbf{P}^1_{v+1} \\ & - \binom{N_1}{v}(N_1-V)\tau^v(1-\tau)^{N_1-v-1}\mathbf{P}^1_v \end{aligned} \tag{6.17}$$

$$\begin{aligned} p_2'(\tau) = \sum_{v=0}^{N_2-1} & \binom{N_2}{v+1}(v+1)\tau^v(1-\tau)^{N_2-v-1}\mathbf{P}^2_{v+1} \\ & - \binom{N_2}{v}(N_2-V)\tau^v(1-\tau)^{N_2-v-1}\mathbf{P}^2_v \end{aligned} \tag{6.18}$$

Similarly from Eq. (6.10), the control points can be determined to satisfy the boundary conditions as

$$\bar{\gamma}(0) = \arcsin\left(\frac{\mu p_2'(0)}{V_m}\right) = \arcsin\left(\frac{\mu N_2(\mathbf{P}^2_1 - \mathbf{P}^2_0)}{V_m}\right) = \gamma_0 \tag{6.19}$$

$$\bar{\gamma}(1) = \arcsin\left(\frac{\mu p_2'(1)}{V_m}\right) = \arcsin\left(\frac{\mu N_2(\mathbf{P}^2_{N_2} - \mathbf{P}^2_{N_2-1})}{V_m}\right) = \gamma_f \tag{6.20}$$

$$\bar{\chi}(0) = \arcsin\left(\frac{\mu p_1'(0)}{V_m\cos\bar{\gamma}(0)}\right) = \arcsin\left(\frac{\mu N_1(\mathbf{P}^1_1 - \mathbf{P}^1_0)}{V_m\cos\bar{\gamma}(0)}\right) = \chi_0 \tag{6.21}$$

$$\bar{\chi}(1) = \arcsin\left(\frac{\mu p_1'(1)}{V_m\cos\bar{\gamma}(1)}\right) = \arcsin\left(\frac{\mu N_1(\mathbf{P}^1_{N_1} - \mathbf{P}^1_{N_1-1})}{V_m\cos\bar{\gamma}(1)}\right) = \chi_f \tag{6.22}$$

Therefore, the control points are determined as follows,

$$\mathbf{P}^2_1 = \mathbf{P}^2_0 + \frac{V_m\sin\gamma_0}{\mu N_2} \tag{6.23}$$

$$\mathbf{P}^2_{N_2-1} = \mathbf{P}^2_{N_2} + \frac{V_m\sin\gamma_f}{\mu N_2} \tag{6.24}$$

$$\mathbf{P}^1_1 = \mathbf{P}^1_0 + \frac{V_m\cos\gamma_0\sin\chi_0}{\mu N_1} \tag{6.25}$$

$$\mathbf{P}^1_{N_1-1} = \mathbf{P}^1_{N_1} + \frac{V_m\cos\gamma_f\sin\chi_f}{\mu N_1} \tag{6.26}$$

In summary, eight boundary conditions $(y_0, y_f, z_0, z_f, \gamma_0, \gamma_f, \chi_0, \chi_f)$ are determined by eight control points $(\mathbf{P}_0^1, \mathbf{P}_{N_1}^1, \mathbf{P}_0^2, \mathbf{P}_{N_2}^2, \mathbf{P}_1^1, \mathbf{P}_{N_1-1}^1, \mathbf{P}_1^2, \mathbf{P}_{N_2-1}^2)$.

6.1.4 Internal State Parameterization

The parameterization of the internal state $x(t)$ is given as an integral form. From Eq. (3.1), the differentiation with respect to τ is given as follows,

$$\mu \bar{x}'(\tau) = V_m \cos\bar{\gamma}(\tau) \cos\bar{\chi}(\tau) \tag{6.27}$$

The definite integral representation is obtained as follows,

$$\mu\left(\bar{x}(\tau_1) - \bar{x}(0)\right) = \int_0^{\tau_1} V_m \cos\bar{\gamma} \cos\bar{\chi} d\tau \tag{6.28}$$

The boundary condition for the internal state can be represented as follows,

$$\bar{x}(1) = \bar{x}(0) + \frac{1}{\mu}\int_0^{\tau_1} V_m \cos\bar{\gamma} \cos\bar{\chi} d\tau \equiv x_f \tag{6.29}$$

Note from Eq. (6.29) that the final value $\bar{x}(1)$ is determined by the initial value $\bar{x}(0)$ and the shape of $\bar{\gamma}(\tau)$ and $\bar{\chi}(\tau)$.

It is hard to analyticall express the definite integral of the polynomials in the transcedental function, and therefore Eq. (6.29) can be satisfied by solving the following optimization problem.

6.1.5 Optimization Problem Formulation

The optimal trajectory can be attained by the parameter optimization problem of the control points of the curves, $p_1(\tau)$ and $p_2(\tau)$. The optimization variables for optimization are $\mathbf{P}_v^1, \mathbf{P}_v^2, v \in \{2, \cdots, N_i - 2\}, i \in \{1, 2\}$, and μ.

Let us consider the following cost function for the optimization problem.

$$\min J = \phi(\mathbf{x}(\mathbf{P}_v^1, \mathbf{P}_v^2), \mu)\Big|_{\tau=1} + \int_0^1 L(\mathbf{x}(\mathbf{P}_v^1, \mathbf{P}_v^2), \mu) d\tau \tag{6.30}$$

where

$$\phi(\mathbf{x}) = (\bar{x}(1) - x_f)^2 \tag{6.31}$$

$$L(\mathbf{x}) = 1/\mu \tag{6.32}$$

Note that the cost function consists of two terms. The first term $\phi(\mathbf{x})$ is for the final value of the internal state $\bar{x}(1)$, the second term $L(\mathbf{x})$ is for the minimum time problem, minimizing $1/\mu = t_f$.

Because the optimization problem formulated above belongs to a typical nonlinlear parameter optimization problem, the optimal solution can be obtained by numerical optimization algorithms including sequential quadratic programming algorothm.

6.1.6 Nonlinear Constraints

Nonlinear constraints, or the path constraints, are the conditions that should be satisfied throughout the trajectory. In this study, the limits on the angular rates are considered, which are represented as the nonlinear constraints. The angular rate is directly connected to the lateral acceleration of the missile. Therefore, the limit on the angular rate can be understood as the limit constraint on the lateral accelaration.

From Eq. (6.10), the angular rates can be obtained as follows,

$$\sin\bar{\gamma} = \frac{\mu}{V_m} p_2' \tag{6.33}$$

$$\sin\bar{\chi} = \frac{\mu p_1'}{V_m \cos\bar{\gamma}} \tag{6.34}$$

Differentiating with respect to τ yields

$$\bar{\gamma}'(\tau) = \frac{\mu p_2''}{V_m \cos\bar{\gamma}} \tag{6.35}$$

$$\bar{\chi}'(\tau) = \frac{\mu}{V_m \cos\bar{\chi}} \left(\frac{p_1''}{\cos\bar{\gamma}} + \frac{p_1' \sin\bar{\gamma}\bar{\gamma}'}{(\cos\bar{\gamma})^2} \right) \tag{6.36}$$

The resulting nonlinear constraints are given as follows,

$$\begin{aligned} |\bar{\gamma}'(\tau)| &\leq \bar{\gamma}'_{\text{lim}} \\ |\bar{\chi}'(\tau)| &\leq \bar{\chi}'_{\text{lim}} \end{aligned} \tag{6.37}$$

6.2 Simulation Demonstration

The results of the numerical simulation are provided in this section. The boundary conditions considered in the simulation are summarized in Table. 6.1. The speed of the missile is set to $V_m = 100\text{m/s}$.

6.2.1 Optimal Output Trajectory Shaping

Figures 6.1 and 6.2 show the minimum time optimal states and trajectory with boundary conditions in Table 6.1. A Feasible solution satisfying the boundary condition is obtained. Figures 6.3 and 6.4 show the minimum time optimal states and trajectory with additional nonlinear constraints, Eq. (6.37). The constriained $\dot{\gamma}$ and $\dot{\chi}$ history are shown in Fig. 6.5. The angular acceleration limits are given as $\dot{\gamma}_{\text{lim}} = 1.2\text{deg/s}$ and $\dot{\chi}_{\text{lim}} = 1.2$ deg/s. Note that a feasible solution is found that satisfies the boundary condition and the nonlinear constraint. The optimization result are summarized in Table 6.2. The final time of the optimal solution with nonlinear constraints is bigger than that without considering nonlinear constraints. The maximum altitude of the solution considering nonlinear constraints is much higher.

Initial States $[\bar{x}_0, \bar{y}_0, \bar{z}_0]$, m	[0, 0, 0]
Initial Inputs $[\bar{\gamma}_0, \bar{\chi}_0]$, deg	[80, 0]
Final States $[\bar{x}_1, \bar{y}_1, \bar{z}_1]$, m	[10,000, 0, 0]
Final Inputs $[\bar{\gamma}_1, \bar{\chi}_1]$, deg	[-80, 0]

Table 6.1: Boundary conditions

NL constraints		Optimized coefficients and final time
No	p_1	[0, 0, 0, 0, 0]
	p_2	[0, 2580.9, -228.12, 2680.9, 0]
	t_f	108.89 sec
Yes	p_1	[0, 0, 168.55, 0, 0]
	p_2	[0, 3858.2, 8211.7, 3858.2, 0]
	t_f	156.71 sec

Table 6.2: Optimal Bézier curve paramters and final time

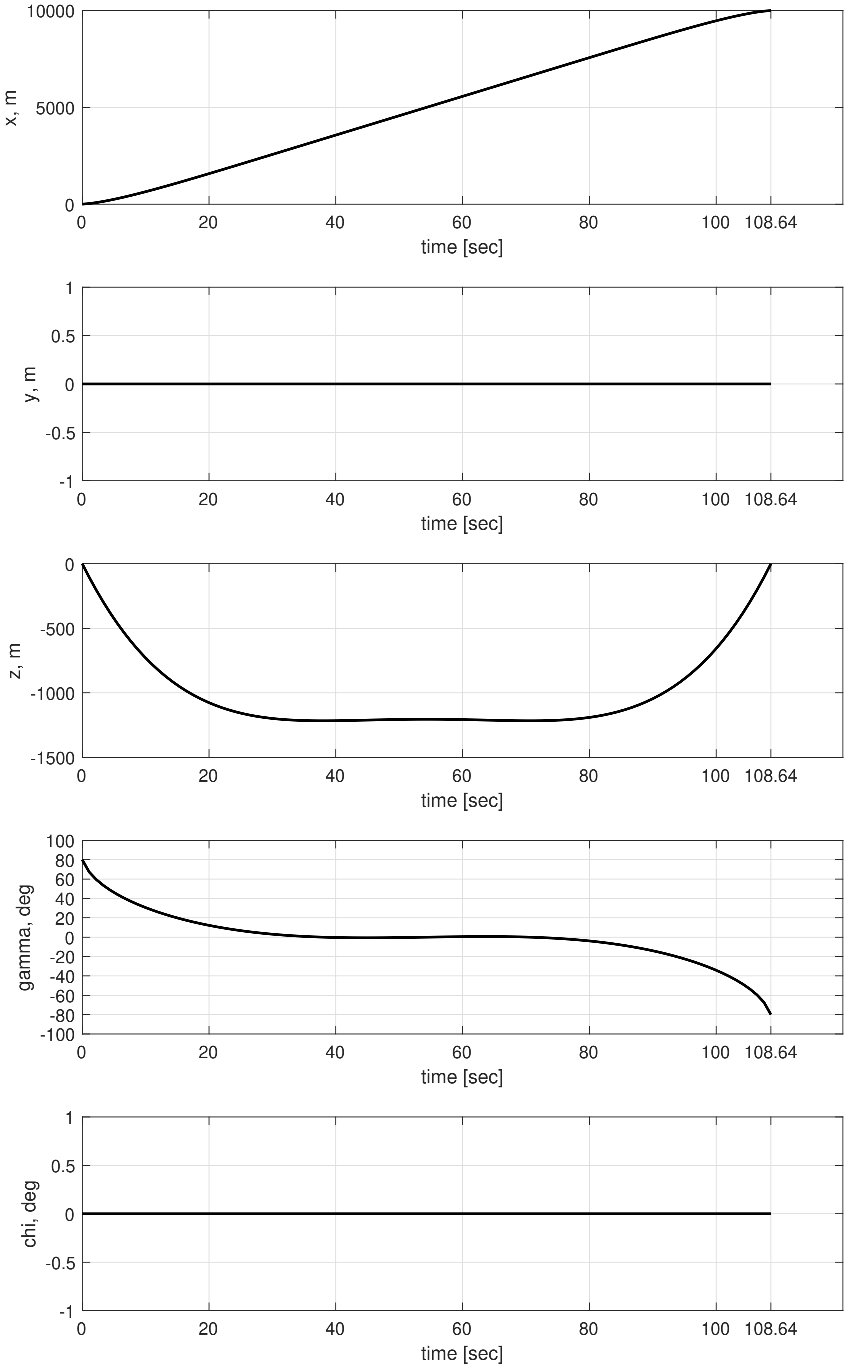


Figure 6.1: Optimized state history (minimum time)

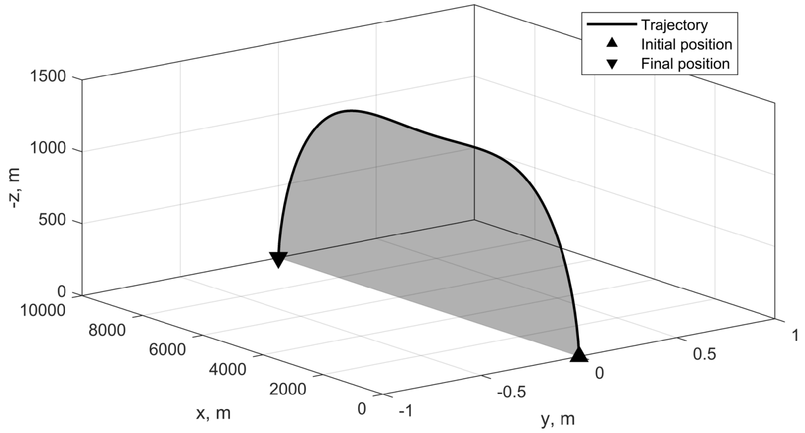


Figure 6.2: Optimized trajectory (minimum time)

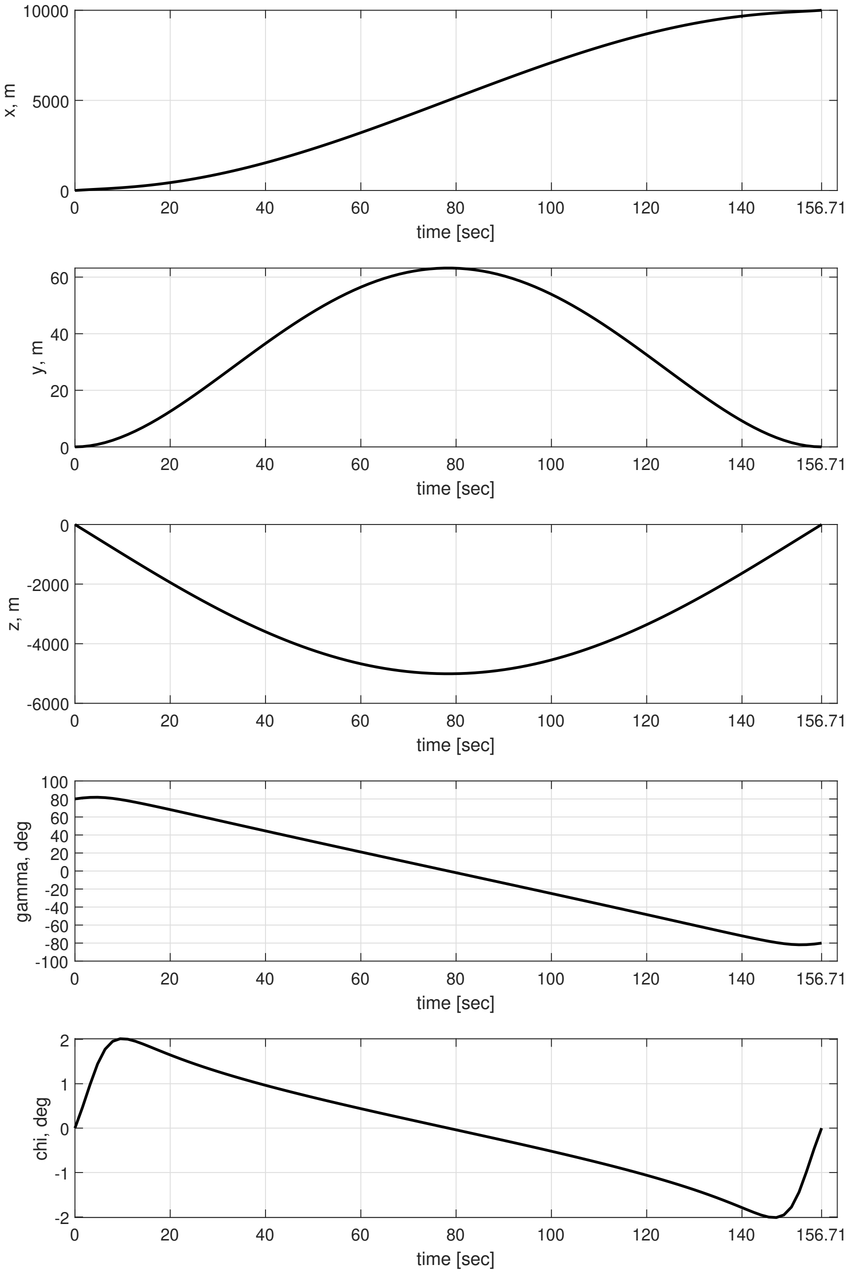


Figure 6.3: Optimized state history (minimum time, constrained)

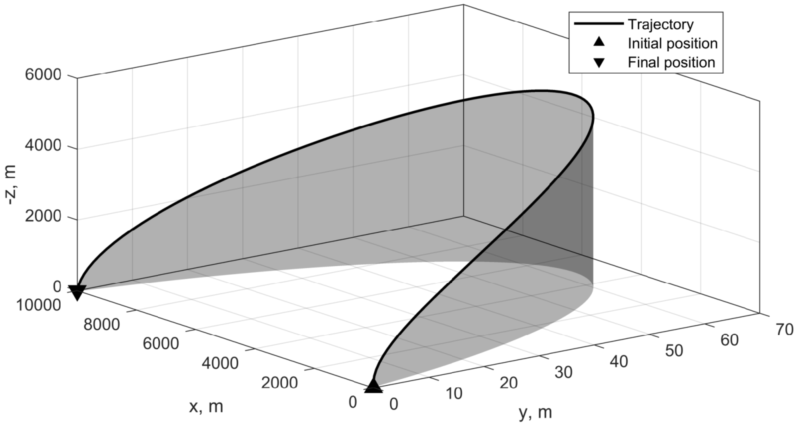


Figure 6.4: Optimized trajectory (minimum time, constrained)

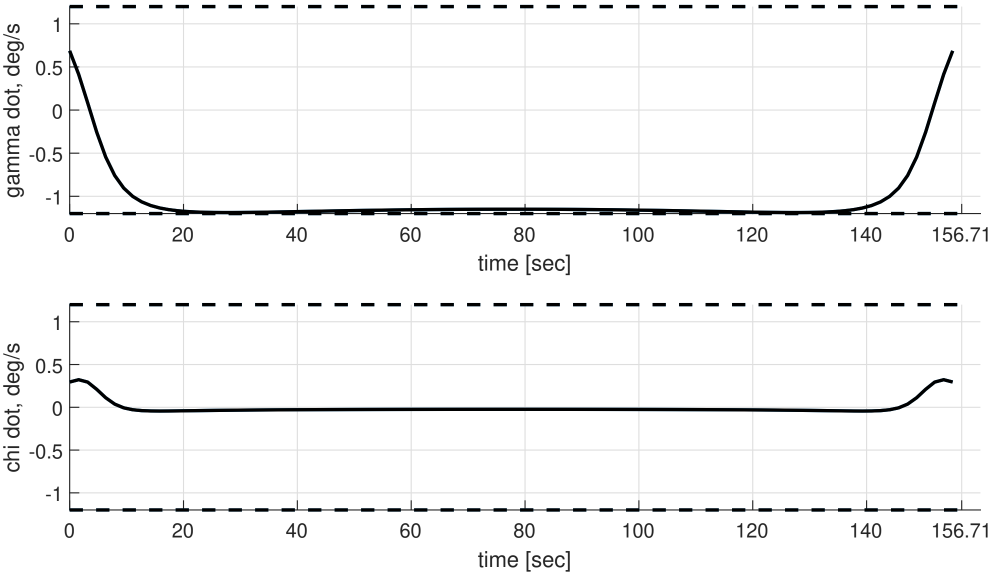


Figure 6.5: Optimized constrained variable history (minimum time, constrained)

Analysis on Optimized Trajectory

The optimized trajectories are analyzed by numerical simulation. The feedback linearization controller using Eq. (6.8) is used, but the limited lateral acceleration is considered in the simulation. Therefore, the missile may not have enough maneuverability to follow the optimal trajectory if the nonlinear constraints are not considered in the optimization. Figures 6.6 - 6.8 show the simulation result. The optimized trajectory requires a sharp maneuver in γ. However, the missile cannot follow the given reference trajectory because of the limitation in the lateral acceleration.

Figures 6.9 - 6.11 show the simulation result with the optimal trajectory considering the nonlinear constraints. Because the nonlinear constraints are considered to generate the reference trajectory in the optimization process, the resulting lateral acceleration command is small enough for the missile to follow the reference trajectory.

The optimal trajectory obtained by utilizing the proposed OOTS algorithm can be used to follow reference trajectory by any arbitrary path-following guidance algorithms. In this dissertation, the reference trajectory is designed for the VFG algorithm. Because the reference trajectory is given in three-dimensional space, the three-dimensional VFG proposed in Sec. 5.3 is used for the design of the vector field. In this study, three types of vector fields are considered: i) Asymptotic stable vector field, Eq. (5.46), ii) a vector field with finite-time convergent k_f with update law, Eq. (5.48), and iii) a vector field with inite-time convergent k_f without update law, Eq. (5.53).

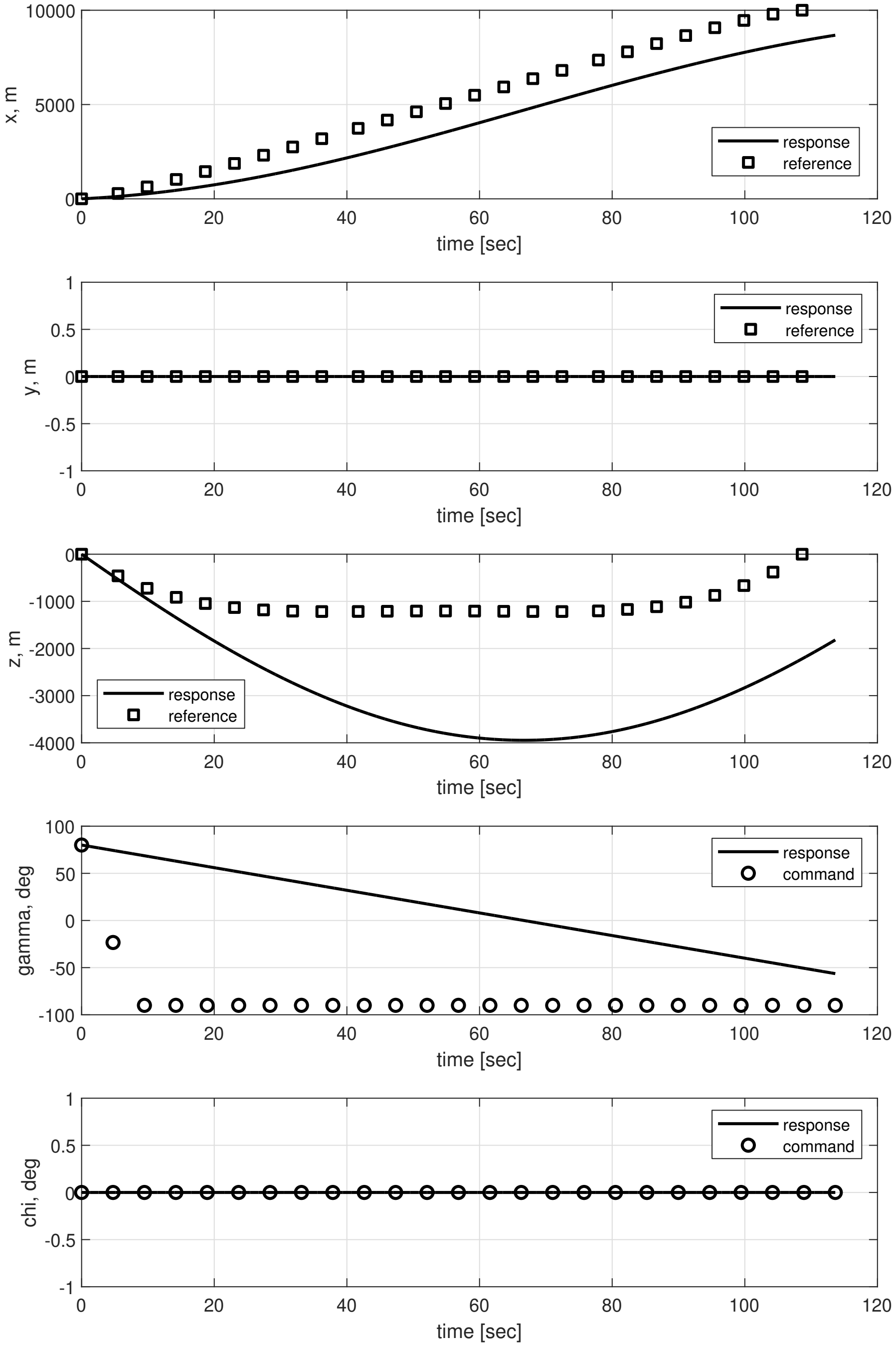


Figure 6.6: OOTS simulation result - states (w/o NL constraints)

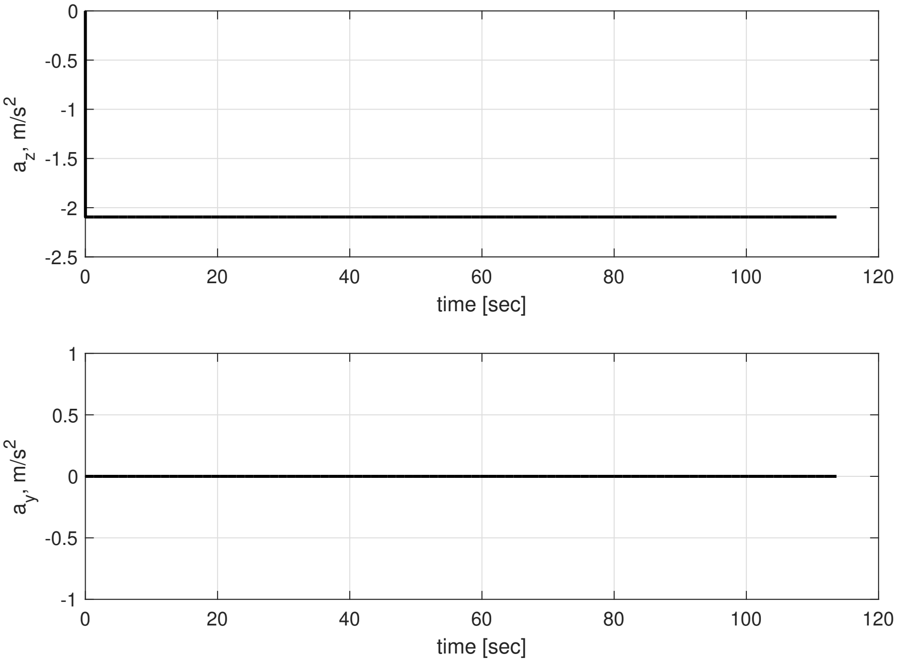


Figure 6.7: OOTS simulation result - accelerations (w/o NL constraints)

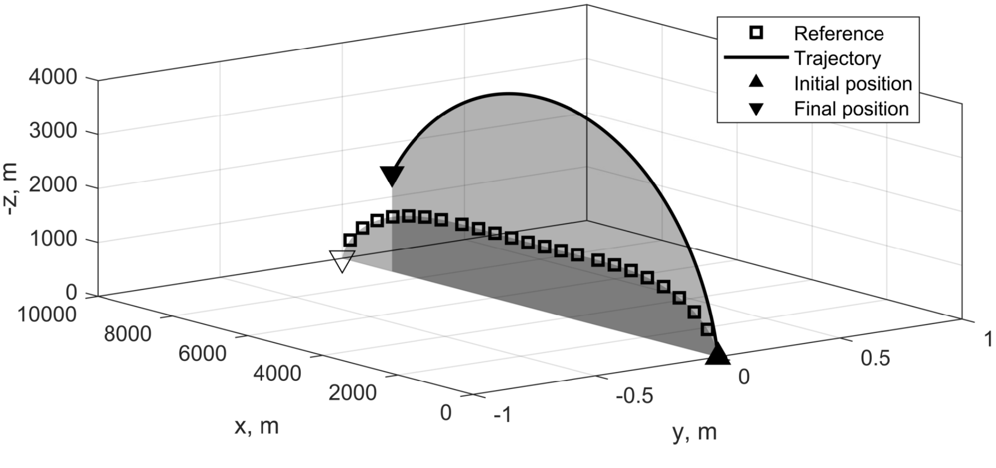


Figure 6.8: OOTS simulation result - trajectory (w/o NL constraints)

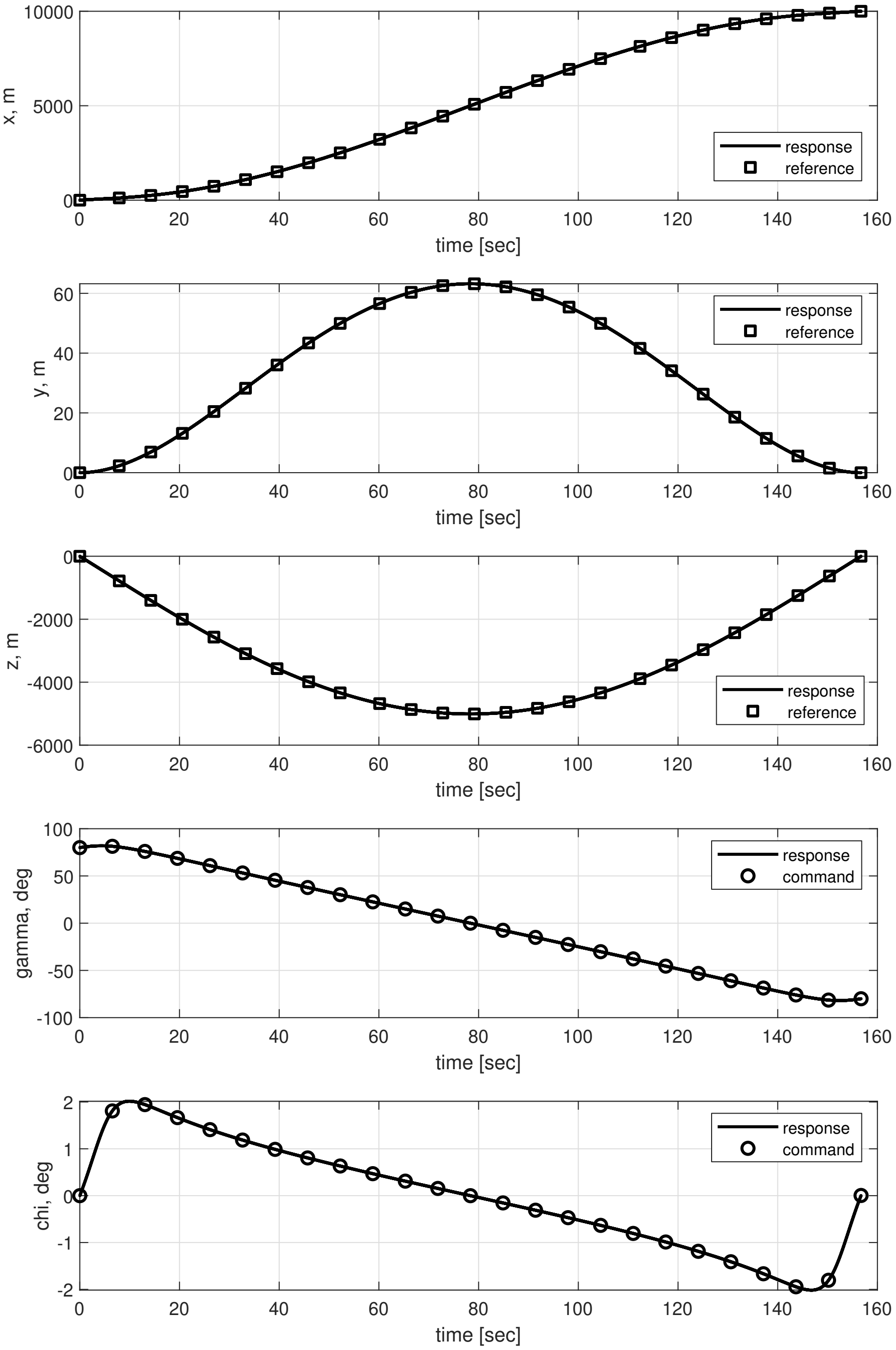


Figure 6.9: OOTS simulation result - states (with NL constraints)

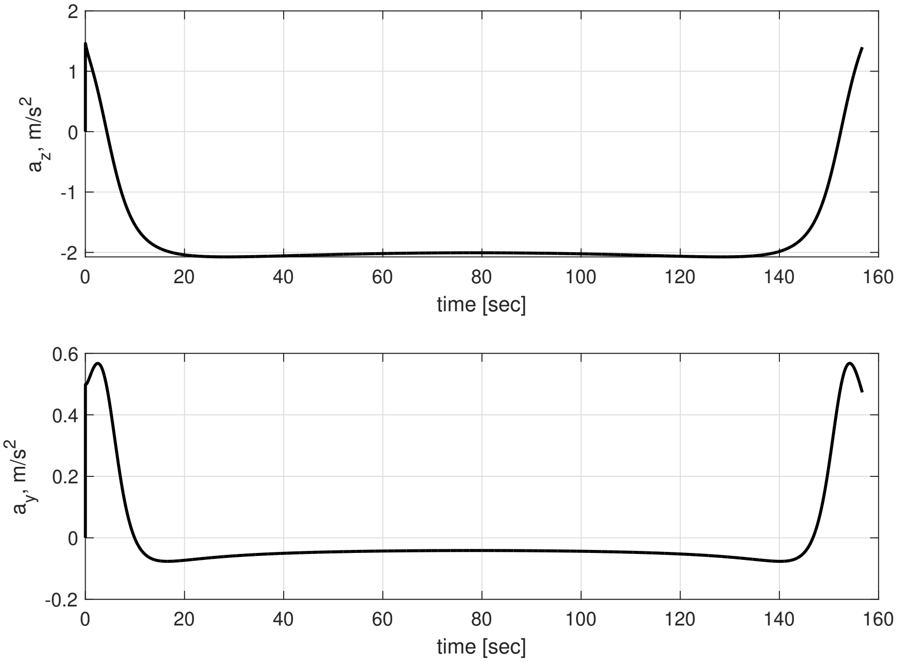


Figure 6.10: OOTS simulation result - accelerations (with NL constraints)

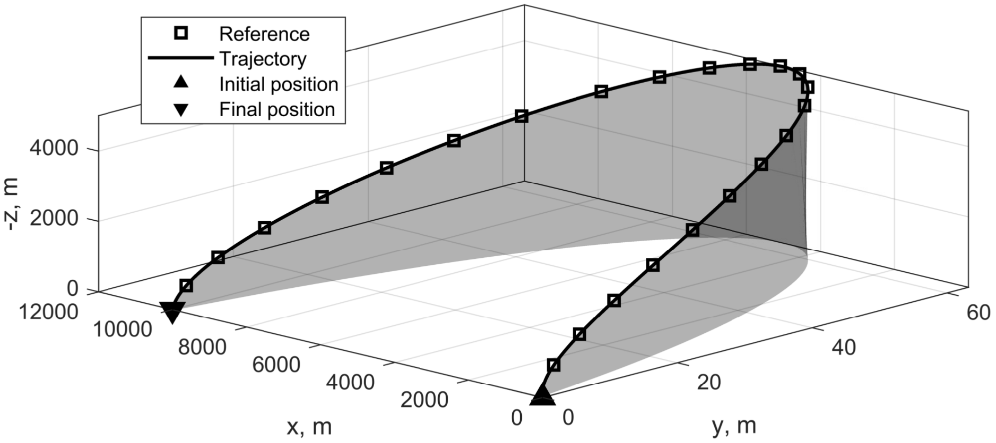


Figure 6.11: OOTS simulation result - trajectory (with NL constraints)

6.2.2 OOTS and Asymptotically Stable VFG

In this section, the OOTS with feedback linearization controller, Eq (6.8) is compared with the asymptotically stable VFG, Eq. (5.46).

In this simulation, the initial position error in (x, y, z) is considered. The feedback linearization controller regulates the error between the reference trajectory and the system output variables, y, and z. However, the controller cannot consider x, the internal state variable, and therefore the initial error affects the boundary condition of the internal state variable not to be satisfied. Meanwhile, VFG is robust against the initial position error than the feedback linearization controller. The optimal trajectory is given as the reference curve to VFG, and VFG regulates the range and the range-rate from the curve.

The initial position error, $\mathbf{e}_0$, considered in the simulation is $\mathbf{e}_0 = [0, 100, -100]$m, where $\mathbf{P}(0) = \mathbf{P}_0 + \mathbf{e}_0$. Figures 6.12 - 6.14 show the simulation result using OOTS. The OOTS regulates the output trajectory error. However, the internal state, x, could not meet the desired final value, and a considerable miss distance occurs because of the initial position error. Figures 6.15 - 6.17 show the simulation result using VFG. Results of the OOTS result and VFG are summarized in Table. 6.3.

Outer-loop controller	Miss distance, m
OOTS	280.43
VFG, Asymptotic stable	0.87

Table 6.3: OOTS and asymptotically stable VFC (initial position error $\mathbf{e}_0 = [0, 100, -100]$)

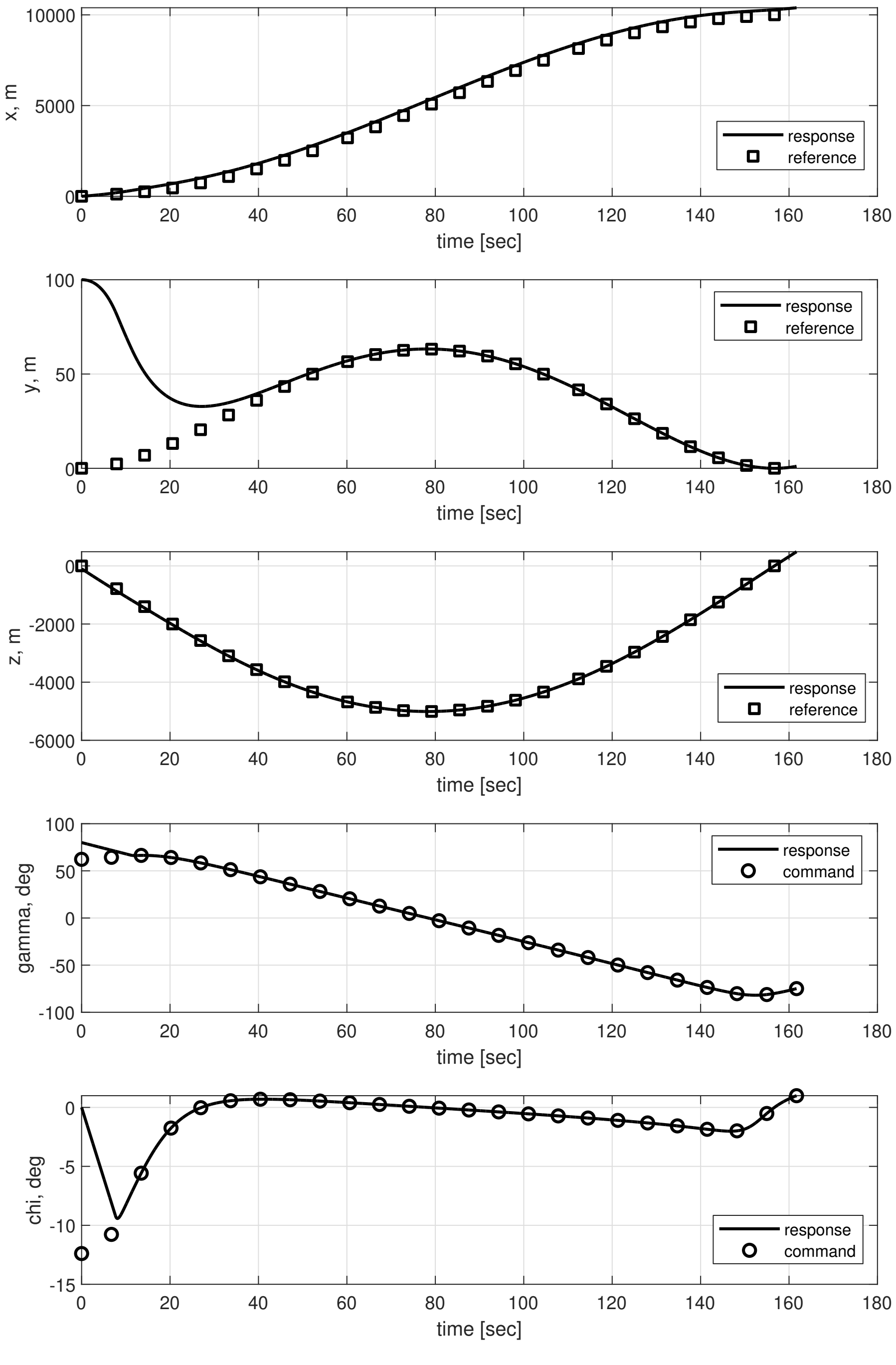


Figure 6.12: Trajectory-following simulation using OOTS, states

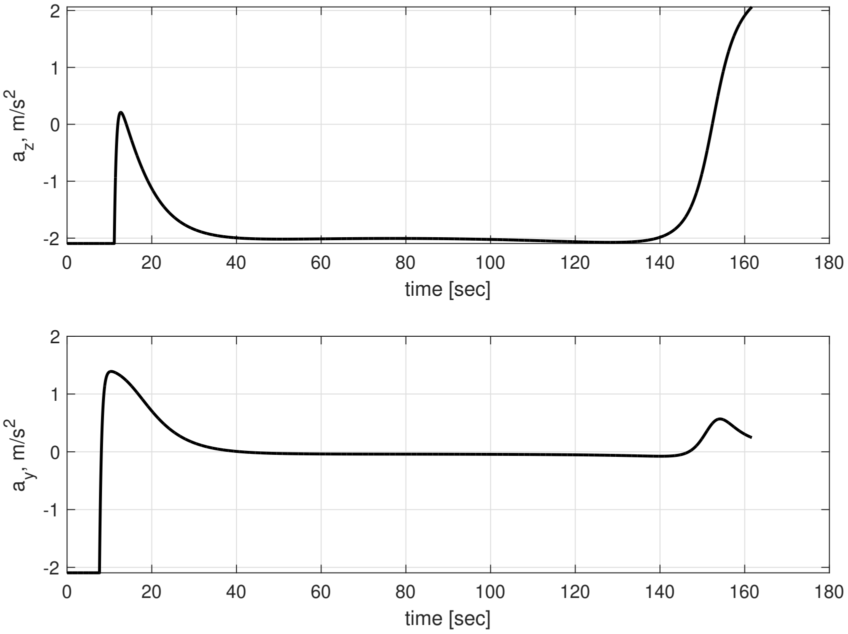


Figure 6.13: Trajectory-following simulation using OOTS, accelerations

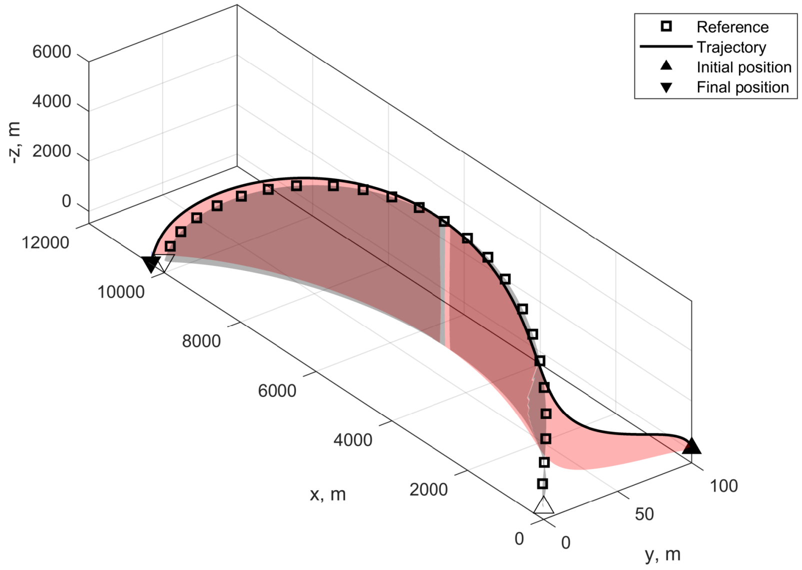


Figure 6.14: Trajectory-following simulation using OOTS, trajectory

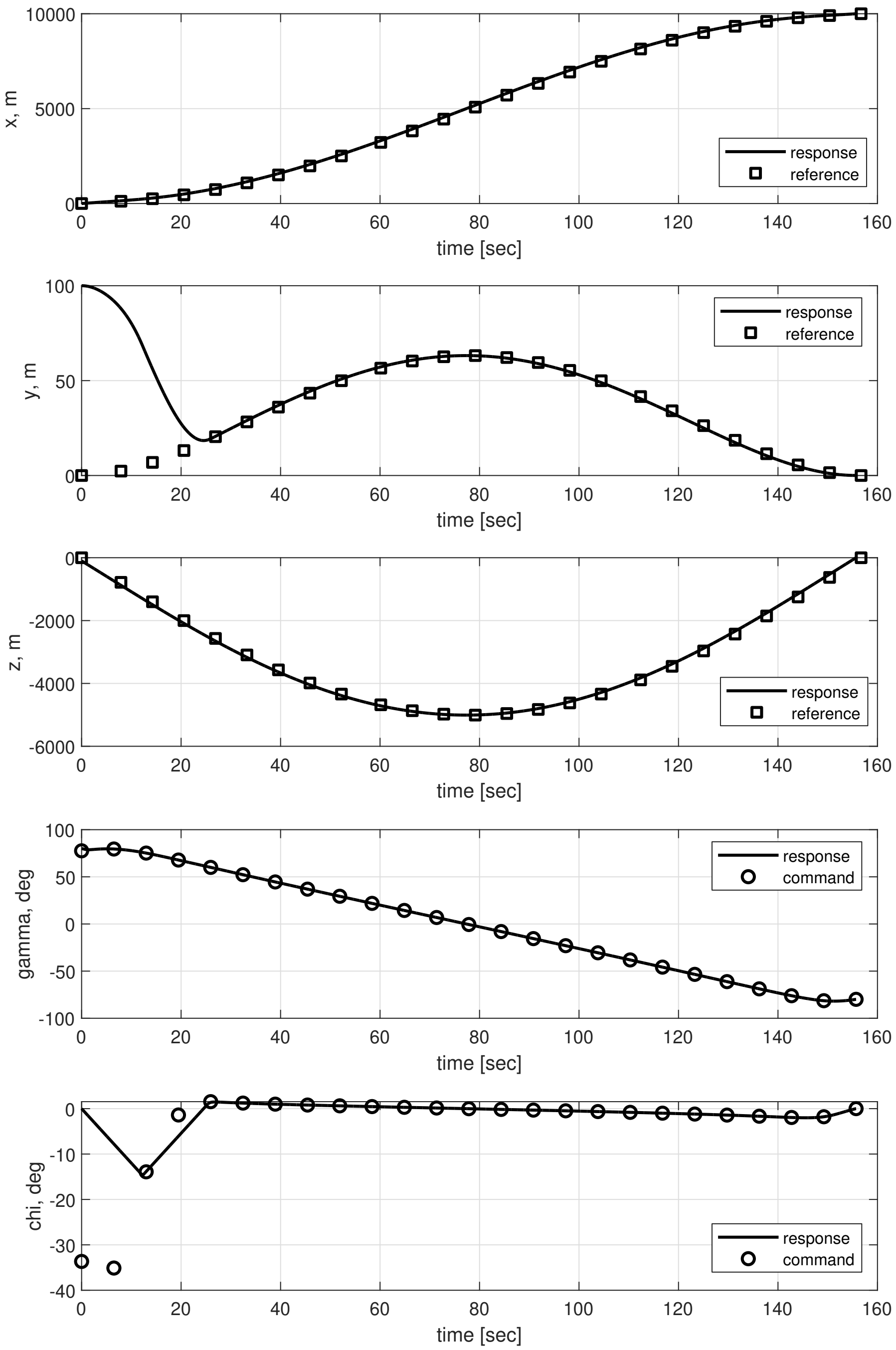


Figure 6.15: Trajectory-following simulation using VFG, states

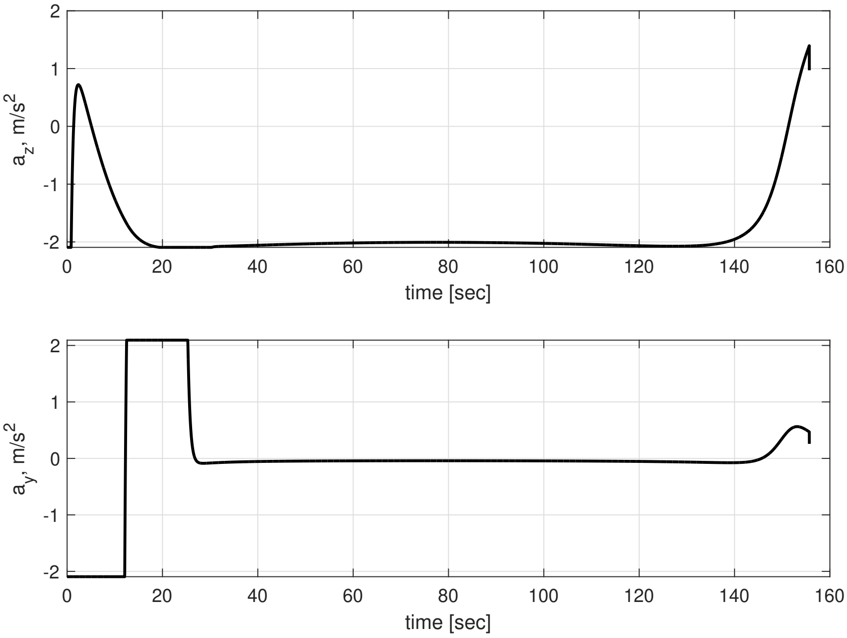


Figure 6.16: Trajectory-following simulation using VFG, accelerations

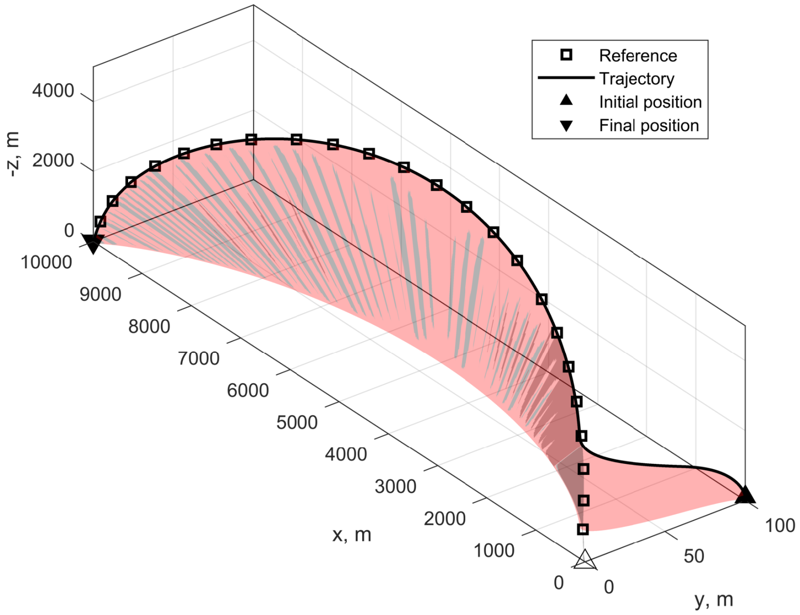


Figure 6.17: Trajectory-following simulation using VFG, trajectory

6.2.3 Monte-Carlo Simulations for VFGs

To verify the proposed method, Monte-Carlo simulation is performed. The initial position error is considered as a random vector, $\mathbf{p}_e \sim \mathcal{N}(0, 50^2)$, from the Gaussian distribution. A hundred number of Monte-Carlo simulations are conducted. The random seed is controlled for a fail comparison between various parameters. That is, the same set of 100 random simulations is considered for each parameter settings. The distributions of the initial position error are shown in Figs 6.18 and 6.19. The parmeters for the Monte-Carlo simulations are summarized in Table. 6.4. The reference curve is the optimal trajectory obtained by the OOTS considering the nonlinear constraints. The lateral accelerations' limits are also considered in the simulation.

The Monte-Carlo simulations are conducted for three different outer-loop controllers, that is, the proposed VFG algorithms: asymptotically stable VFG and finite-time convergent VFGs. The mean values of the miss distance using three VFG algorithms are summarized in Table. 6.5.

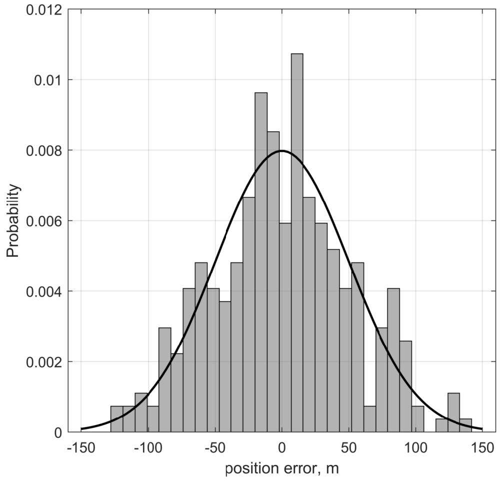


Figure 6.18: Gaussian distribution of the initial position error for Monte-Carlo simulation

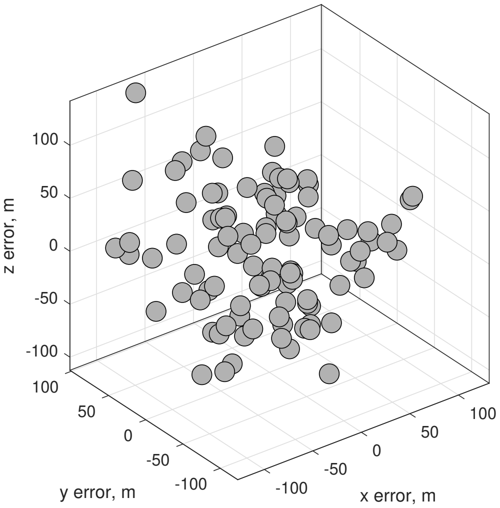


Figure 6.19: The scattered initial position errors for Monte-Carlo simulation

Algorithm	Parameter	Value
Asymptotically stable k_f	P	$\begin{bmatrix} 5 & 0.05 \\ 0.05 & 0.05 \end{bmatrix}$
	(ρ_1, ρ_2)	(0.6, 0.8)
Finite-time convergent k_f, #1	P	$\begin{bmatrix} 5 & 0.05 \\ 0.05 & 0.05 \end{bmatrix}$
	(α, λ)	(0.4, 0.9)
Finite-time convergent k_f, #2	(α, λ)	(0,4. 0.9)

Table 6.4: Parameter settings for the proposed algorithms

Algorithm	Mean miss distance, m
Asymptotically stable k_f	0.8783
Finite-time stable k_f, #1	0.5909
Finite-time stable k_f, #2	0.5247

Table 6.5: Performance summary of proposed guidance algorithms

Asymptotically Stable VFG

The asymptotically stable VFG useds the k_f update law, Eq. (5.46). For a given random initial position error from the curve, the missile is guided to follow the reference curve. The mean values of the miss distance of the Monte-Carlo simulation is 0.8783m.

Finite-Time Convergent VFG #1

The finite-time convergent VFG #1 uses the k_f update law, Eq. (5.48). For a given random initial position error from the curve, the missile is guided to follow the curve using VFG with the finite-time convergent gain update law ($\dot{k}_f$). The upper limit of the settling time, Eq. (2.54), is compared with the actual convergent time. For comparison, the actual convergent time is defined as when the normalized Lyapunov function value, the Lyapunov function value divided by the initial value, becomes smaller than 0.01.

Figure 6.20 shows the settling time's upper limit, which is computed from Eq. (2.54), and the actual settling time, which is computed from the normalized Lyapunov function. Note that the actual settling time is smaller than the upper limit for all Monte-Carlo simulations. Therefore, it can be stated that Eq. (2.54) estimates the settling time conservatively. The mean value of the miss distance of the Monte-Carlo simulation is 0.5909m.

Finite-Time Convergent VFG #2

The finite-time convergent VFG #2 uses the k_f, Eq. (5.53). For a given random initial position error from the curve, the missile is guided to follow the curve using VFG with the finite-time convergent gain (k_f). The upper limit of the settling time, Eq. (2.54), is compared with the actual convergent time. The actual convergent time, defined before, is again used for the comparison.

Figure 6.21 shows the settling time's upper limit and the actual settling time. Again, the actual settling time is smaller than the upper limit for all Monte-Carlo simulations, and it can be stated that Eq. (2.54) conservatively estimates the settling time. Although the same α and λ values are used for both of the finite-time convergent VFG #1 and #2, the upper limit of the settling time is different because the definition of the Lyapunov function is different for each other. The mean value of the miss distance of the Monte-Carlo simulation is 0.5247m.

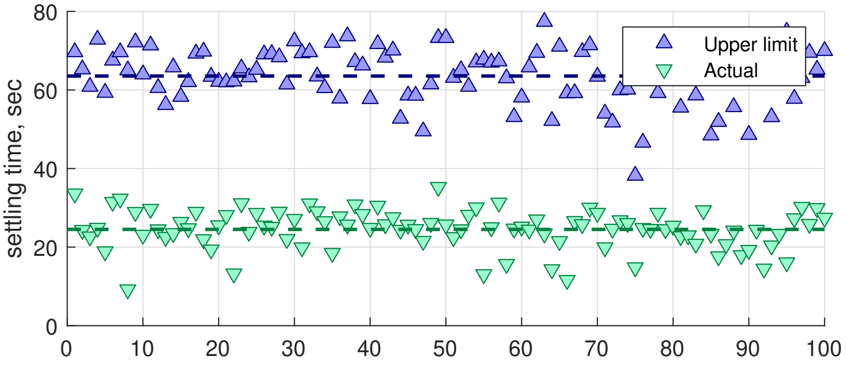


Figure 6.20: Settling time comparison between the upper limit and the actual, finite-time convergent VFG #1

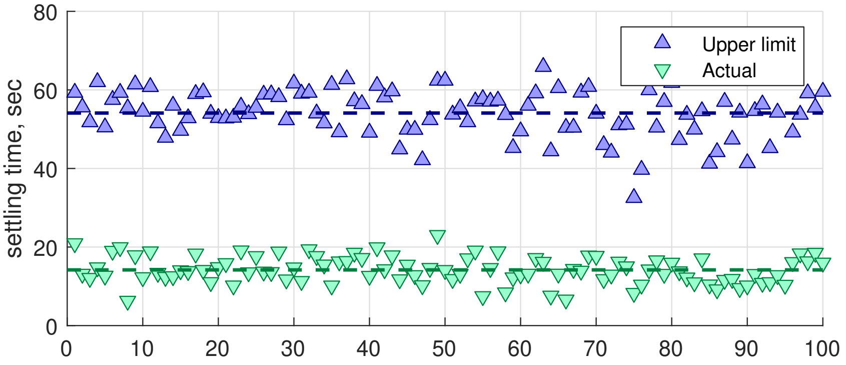


Figure 6.21: Settling time comparison between the upper limit and the actual, finite-time convergent VFG #2

Chapter 7

Conclusions

7.1 Concluding Remarks

A framework for the generation of optimal trajectory and guidance algorithms for the missile system was proposed in this dissertation. The proposed framework consists of the time-scale separation of the missile's kinematic system, nonlinear optimal trajectory generation using parameterized curves, and the vector field-based guidance algorithm design.

First, the time-scale separation of the missile's kinematic equations of motion was discussed. The separation divides the system into the outer-loop(slow-scale) and the inner-loop(fast-scale) systems. In the slow-scale system, the optimal trajectory of the system was designed. The fast-scale system state variables are considered as control input variables in the slow-scale system. The desired control input command designed in the slow-scale system is handed over to the fast-scale system and plays the reference command role. In the fast-scale system, a regulating controller was designed for the system to follow the given reference command from the slow-scale system.

Second, the optimal trajectory was designed as a parameterized curve in the design space. The derivation of the optimal trajectory was accomplished in the slow-scale system of the missile's kinematic system. The output trajectory tracking control algorithm was proposed for attaining the parameterization of the system output, states, and input variables. The boundary conditions and nonlinear constraints of the trajectory can be readily imposed by manipulating the shape of the parameterized curve, i.e., Bézier curve.

Third, the vector field-based guidance algorithm was proposed. The vector field-based guidance algorithm is a two- or three-dimensional guidance algorithm for a missile with constant speed. The reference curve represented in an implicit function or a parameterized curve was considered for the path following-guidance of the missile. The vector field was computed in the design space and used as the directional command for the missile. The missile was controlled to follow the vector field direction. The design of the vector field focuses on the asymptotic and finite-time convergence on the reference curve.

The proposed methodologies construct the framework for the optimal trajectory-generation and following-guidance of the missile in the three-dimensional space. Numerical simulations were conducted for the demonstration and performance evaluation of the proposed framework. The methodologies proposed in this dissertation could be utilized for any appropriate systems and applications.

7.2 Future Work

The methodologies proposed in this dissertation should be further developed for the improvements. The improvements can be a generalization of the method, utilizing various functions, and an expansion that gives more degree of freedom.

First, it is required to secure the robustness of the feedback linearization controller in OOTS. The numerical simulation demonstrated that the feedback linearization controller is weak to the error in the initial configuration for satisfying the boundary conditions, especially for the internal states.

Second, though a linear time-parameterization and Bézier curve-based state-parameterization are considered in this study, different parametric functions and curves can also be used for the parameterization. For instance, a nonlinear parameterization of time may enhance the representation availability on the state trajectory shaping.

Third, the vector field-based guidance algorithm should be expanded to consider time-varying speed. In the proposed method, the vector field generates a directional command input to the missile system. Usually, assumption on constant speed is widely used in designing the missile guidance laws. However, it is not the case for other platforms including UAVs or cars. Therefore, the vector field-based guidance algorithm should be expanded to the guidance algorithm with time-varying speed, which gives the directional command and the command with magnitude. The vector field with time-varying magnitude has more degree of freedom than that of the vector field without magnitude command.

Bibliography

[1] Beard, R. W. and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, Princeton, NJ, 2012.

[2] Tsourdos, A., White, B., and Shanmugavel, M., *Cooperative Path Planning of Unmanned Aerial Vehicles*, Vol. 32, John Wiley & Sons, Hoboken, NJ, 2010.

[3] Yomchinda, T., Horn, J. F., and Langelaan, J. W., "Modified Dubins Parameterization for Aircraft Emergency Trajectory Planning," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 231, No. 2, 2017, pp. 374–393.

[4] Askari, A., Mortazavi, M., Talebi, H., and Motamedi, A., "A new approach in UAV path planning using Bezier–Dubins continuous curvature path," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 230, No. 6, 2016, pp. 1103–1113.

[5] Vinokursky, D. L., Mezentceva, O. S., and Samoylov, P. V., "Trajectory Planning of UAV Group: Pythagorean Hodograph and Bernstein-Bezier Composite Curves in the Plane," *2020 International Russian Automation Conference (RusAutoCon)*, Sochi, Russia, Sep. 2020.

[6] LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge, 2006.

[7] Lawrance, N. and Sukkarieh, S., “Wind Energy Based Path Planning for a Small Gliding Unmanned Aerial Vehicle,” *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, Aug. 2009.

[8] Cho, N., Lee, S., Kim, J., Kim, Y., Park, S., and Song, C., “Wind Compensation Framework for Unpowered Aircraft Using Online Waypoint Correction,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 1, 2020, pp. 698–710.

[9] Ghose, D., *Guidance of missiles*, Indian Institute of Science, Bangalore, India, 2012.

[10] Lee, S., Cho, N., and Kim, Y., “Impact-Time-Control Guidance Strategy with a Composite Structure Considering the Seeker’s Field-of-View Constraint,” *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1566–1574.

[11] Sujit, P., Saripalli, S., and Sousa, J. B., “Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicless,” *IEEE Control Systems*, Vol. 34, No. 1, 2014, pp. 42–59.

[12] Cho, N., Kim, Y., and Park, S., “Three-Dimensional Nonlinear Differential Geometric Path-Following Guidance Law,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 12, 2015, pp. 2366–2385.

[13] Xu, H. and Guedes Soares, C., "Vector Field Path Following for Surface Marine Vessel and Parameter Identification Based on LS-SVM," *Ocean Engineering*, Vol. 113, 2016, pp. 151–161.

[14] Lee, J. and Kim, Y., "Neural Network-based Nonlinear Dynamic Inversion Control of Variable-span Morphing Aircraft," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 234, No. 10, 2020, pp. 1624–1637.

[15] Ann, S., Lee, S., Kim, Y., and Ahn, J., "Midcourse Guidance for Exoatmospheric Interception Using Response Surface based Trajectory Shaping," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 234, No. 10, 2020, pp. 3655–3673.

[16] Kelley, H. J., "Aircraft Maneuver Optimization by Reduced-Order Approximation," *Control and Dynamic Systems*, Vol. 10, 1973, pp. 131–178.

[17] Sesak, J. and Higgins, T., "Sensitivity-constrained Linear Optimal Control Forced Singular Perturbation Model Reduction," *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, San Diego, CA, 1978.

[18] Sesak, J. and Coradetti, T., "Decentralized Control of Large Space Structures via Forced Singular Perturbation," *17th Aerospace Sciences Meeting*, New Orleans, LA, Jan. 1979.

[19] Calise, A., "Singular Perturbation Methods for Variational Problems in Aircraft Flight," *IEEE Transactions on Automatic Control*, Vol. 21, No. 3, 1976, pp. 345–353.

[20] Khalil, H., “Stabilization of Multiparameter Singularly Perturbed Systems,” *IEEE Transactions on Automatic Control*, Vol. 24, No. 5, 1979, pp. 790–791.

[21] Shinar, J., “On Applications of Singular Perturbation Techniques in Nonlinear Optimal Control,” *Automatica*, Vol. 19, No. 2, 1983, pp. 203–211.

[22] Shinar, J. and Farber, N., “Horizontal Variable-speed Interception Game Solved by Forced Singular Perturbation Technique,” *Journal of Optimization Theory and Applications*, Vol. 42, No. 4, 1984, pp. 603–636.

[23] Menon, P. K. A. and Briggs, M. M., “Near-optimal Midcourse Guidance for Air-to-air Missiles,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 4, 1990, pp. 596–602.

[24] Vrabel, R., “On the Approximation of The Boundary Layers for the Controllability Problem of Nonlinear Singularly Perturbed Systems,” *Systems & Control Letters*, Vol. 61, No. 3, 2012, pp. 422–426.

[25] Sridhar, B. and Gupta, N., “Missile Guidance Laws Based on Singular Perturbation Methodology,” *Journal of Guidance and Control*, Vol. 3, No. 2, 1980, pp. 158–165.

[26] Cheng, V. H. L. and Gupta, N. K., “Advanced Midcourse Guidance for Air-to-air Missiles,” *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 2, 1986, pp. 135–142.

[27] Manickavasagam, M., Sarkar, A. K., and Vaithiyanathan, V., “A Singular Perturbation Based Midcourse Guidance Law for Realistic Air-to-air Engagement,” *Defence Science Journal*, Vol. 67, No. 1, 2017, pp. 108–118.

[28] Bharadwaj, S. and Wu, M., "Identifying Time-Scale Structure for Simplified Guidance Law Development Linear Variational System Associated With Measures of Time-Scale Separation," *AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1997.

[29] Mease, K. D., Bharadwaj, S., and Iravanchy, S., "Timescale Analysis for Nonlinear Dynamical Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 318–330.

[30] Mease, K. and Topcu, U., "Two Timescale Nonlinear Dynamics and Slow Manifold Determination," *AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, Aug. 2005, pp. 307–315.

[31] Topcu, U., *Lyapunov Exponents and Vectors for Determining The Geometric Structure of Nonlinear Dynamical Systems*, MS Thesis, Department of Mechanical and Aerospace Engineering, University of Califormia at Irvine, Irvine, CA, 2005.

[32] Shapira, I. and Ben-Asher, J. Z., "Singular Perturbation Analysis of Optimal Glide," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 915–918.

[33] Dhananjay, N., Ghose, D., and Bhat, M. S., "Performance Analysis of Guidance Laws Based on Timescale Gap," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 3, 2010, pp. 574–590.

[34] Menon, P., Iragavarapu, V., Ohlmeyer, E., Menon, P., Iragavarapu, V., and Ohlmeyer, E., "Nonlinear Missile Autopilot Design Using Time-scale

Separation," *Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, New Orleans, LA, Aug. 1997.

[35] Naidu, D. S. and Calise, A. J., "Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1057–1078.

[36] Wenjing Ren, Jiang, B., and Hao Yang, "A Survey on Singular Perturbation Theory in Aerospace Application," *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, Aug. 2016.

[37] Naidu, D. S., "Singular Perturbations and Time Scales in Control Theory and Applications: An Overview," *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Application and Algorithm*, Vol. 9, No. 2, 2002, pp. 233–278.

[38] Bryson, A. E., *Applied Optimal Control: Optimization, Estimation And Control*, CRC Press, New York, NY, 1975.

[39] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[40] Gan, W. Y., Zhu, D. Q., Xu, W. L., and Sun, B., "Survey of Trajectory Tracking Control of Autonomous Underwater Vehicles," *Journal of Marine Science and Technology*, Vol. 25, No. 6, 2017, pp. 722–731.

[41] Lee, S. and Kim, Y., "Vector Field-Based Guidance for Exo-Atmospheric Target Interception," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 6, 2020, pp. 4353–4366.

[42] Cho, N., Kim, Y., and Shin, H.-S., "Generalized Formulation of Linear Nonquadratic Weighted Optimal Error Shaping Guidance Laws," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 6, 2020, pp. 1143–1153.

[43] Farouki, R. T., Giannelli, C., Mugnaini, D., and Sestini, A., "Path Planning with Pythagorean-hodograph Curves for Unmanned or Autonomous Vehicles," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 232, No. 7, 2018, pp. 1361–1372.

[44] Neto, A. A., Macharet, D. G., and Campos, M. F., "Feasible Path Planning for Fixed-wing UAVs Using Seventh Order Bézier Curves," *Journal of the Brazilian Computer Society*, Vol. 19, No. 2, 2013, pp. 193–203.

[45] Faigl, J. and Vana, P., "Surveillance Planning with Bézier Curves," *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, 2018, pp. 750–757.

[46] Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N., and Pascoal, A. M., "Consistent Approximation of Optimal Control Problems Using Bernstein Polynomials," *IEEE 58th Conference on Decision and Control (CDC)*, Nice, France, Dec. 2019.

[47] Ricciardi, L. A. and Vasile, M., "Direct Transcription of Optimal Control Problems with Finite Elements on Bernstein Basis," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 2, 2019, pp. 229–243.

[48] Ghomanjani, F., Farahi, M., and Gachpazan, M., "Bézier Control Points Method to Solve Constrained Quadratic Optimal Control of Time Varying Linear Systems," *Computational & Applied Mathematics*, Vol. 31, No. 3, 2012, pp. 433–456.

[49] de Dilectis, F., Mortari, D., and Zanetti, R., "Bézier Description of Space Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 11, 2016, pp. 2535–2539.

[50] Choe, R., Puig-Navarro, J., Cichella, V., Xargay, E., and Hovakimyan, N., "Cooperative Trajectory Generation Using Pythagorean Hodograph Bézier Curves," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 8, 2016, pp. 1744–1763.

[51] Cichella, V., Kaminer, I., Walton, C., and Hovakimyan, N., "Optimal Motion Planning for Differentially Flat Systems Using Bernstein Approximation," *IEEE Control Systems Letters*, Vol. 2, No. 1, 2018, pp. 181–186.

[52] Varma, S. A. and Kothari, M., "A 3D Pitch and Impact-angle Constrained Guidance Scheme," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 5, 2019, pp. 1571–1584.

[53] Moon, G.-H., Shim, S.-W., and Tahk, M.-J., "Augmented Polynomial Guidance for Terminal Velocity Constraints," *30th Congress of the International Council of the Aeronautical Sciences*, Daejeon, Korea, Sep. 2016.

[54] Tyan, F., “Analysis Of 3d Ppn Guidance Laws for Nonmaneuvering Target,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 4, 2015, pp. 2932–2943.

[55] Kim, T.-H., Lee, C.-H., Jeon, I.-S., and Tahk, M.-J., “Augmented Polynomial Guidance with Impact Time and Angle Constraints,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 4, 2013, pp. 2806–2817.

[56] Gu, W., Zhao, H., and Zhang, R., “A Three-dimensional Proportional Guidance Law Based on RBF Neural Network,” *7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 2008.

[57] Ghosh, S., Ghose, D., and Raha, S., “Capturability of Augmented Pure Proportional Navigation Guidance Against Time-Varying Target Maneuvers,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 5, 2014, pp. 1446–1461.

[58] Kumar, S. R., Rao, S., and Ghose, D., “Nonsingular Terminal Sliding Mode Guidance with Impact Angle Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, 2014, pp. 1114–1130.

[59] Shin, H.-S., Tsourdos, A., and Li, K.-B., “A New Three-Dimensional Sliding Mode Guidance Law Variation with Finite Time Convergence,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No. 5, 2017, pp. 2221–2232.

[60] Ariff, O., Zbikowski, R., Tsourdos, A., and White, B. A., “Differential Geometric Guidance Based on the Involute of the Target’s Trajectory,” *Jour-*

nal of Guidance, Control, and Dynamics, Vol. 28, No. 5, 2005, pp. 990–996.

[61] Tsalik, R. and Shima, T., “Circular Impact-Time Guidance,” *Journal of Guidance, Control, and Dynamics*, Published Online, May 2019, pp. 1–12.

[62] Tsalik, R. and Shima, T., “Inscribed Angle Guidance,” *Journal of Guidance Control and Dynamics*, Vol. 38, No. 1, 2015, pp. 30–40.

[63] Kumar, S. R., Tsalik, R., and Shima, T. Y., “Nonlinear Robust Inscribed Angle Guidance for Stationary Targets,” *AIAA Guidance, Navigation, and Control Conference*, Vol. 40, Grapevine, TX, Jan. 2017, pp. 1815–1823.

[64] Tsalik, R. and Shima, T., “Inscribed-Angle Guidance Against Moving Targets,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3211–3225.

[65] Livermore, R., Tsalik, R., and Shima, T., “Elliptic Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 11, 2018, pp. 2435–2444.

[66] Griffiths, S., “Vector Field Approach for Curved Path Following for Miniature Aerial Vehicles,” *AIAA Guidance, Navigation, and Control Conference*, Vol. 5, Keystone, CO, Aug. 2006, pp. 3375–3389.

[67] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W., “Vector Field Path Following for Miniature Air Vehicles,” *IEEE Transactions on Robotics*, Vol. 23, No. 3, 2007, pp. 519–529.

[68] Frew, E. W., Lawrence, D. A., and Morris, S., “Coordinated Standoff Tracking of Moving Targets using Lyapunov Guidance Vector Fields,”

Journal of Guidance, Control, and Dynamics, Vol. 31, No. 2, 2008, pp. 290–306.

[69] Chen, H., Chang, K., and Agate, C. S., "UAV Path Planning with Tangent-plus-Lyapunov Vector Field Guidance and Obstacle Avoidance," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 2, 2013, pp. 840–856.

[70] Lim, S. and Bang, H., "Guidance Laws for Target Localization using Vector Field Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 3, 2014, pp. 1991–2003.

[71] Pothen, A. A. and Ratnoo, A., "Curvature-Constrained Lyapunov Vector Field for Standoff Target Tracking," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2729–2736.

[72] Sun, S., Wang, H., Liu, J., and He, Y., "Fast Lyapunov Vector Field Guidance for Standoff Target Tracking Based on Offline Search," *IEEE Access*, Vol. 7, 2019, pp. 124797–124808.

[73] Olavo, J. L. G., Thums, G. D., Jesus, T. A., de Araujo Pimenta, L. C., Torres, L. A. B., and Palhares, R. M., "Robust Guidance Strategy for Target Circulation by Controlled UAV," *IEEE Transactions on Aerospace An Electronic Systems*, Vol. 54, No. 3, 2018, pp. 1415–1431.

[74] Oh, H., Kim, S., Shin, H. S., and Tsourdos, A., "Coordinated Standoff Tracking of Moving Target Groups Using Multiple UAVs," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 2, 2015, pp. 1501–1514.

[75] Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., Dutra, B. C. O., and Pereira, G. A. S., "Vector Fields for Robot Navigation along Time-Varying Curves in n-Dimensions," *IEEE Transactions on Robotics*, Vol. 26, No. 4, 2010, pp. 647–659.

[76] Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., Pereira, G. A. S., Dutra, B. C. O., Michael, N., Fink, J., and Kumar, V., "Circulation of Curves Using Vector Fields: Actual Robot Experiments in 2D and 3D Workspaces," *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 65, IEEE, Anchorage, AK, May 2010, pp. 1136–1141.

[77] Frew, E. W. and Lawrence, D., "Tracking Dynamic Star Curves Using Guidance Vector Fields," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 6, 2017, pp. 1488–1495.

[78] Lim, S., Jung, W., and Bang, H., "Vector Field Guidance for Path Following and Arrival Angle Control," *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, May 2014.

[79] Goncalves, V. M., Mclaughlin, R., and Pereira, G. A., "Precise Landing of Autonomous Aerial Vehicles Using Vector Fields," *IEEE Robotics and Automation Letters*, Vol. 5, No. 3, 2020, pp. 4337–4344.

[80] Panagou, D., "Motion Planning and Collision Avoidance using Navigation Vector Fields," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014.

[81] Clem, G. S., *An Optimized Circulating Vector Field Obstacle Avoidance Guidance for Unmanned Aerial Vehicles*, MS Thesis, Department of Mechanical Engineering, Ohio University, Athenes, OH, 2018.

[82] Zhou, D. and Schwager, M., “Vector Field Following for Quadrotors Using Differential Flatness,” *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Hong Kong, China, May 2014, pp. 6567–6572.

[83] Liang, Y., Jia, Y., Du, J., and Zhang, J., “Vector Field Guidance for Three-dimensional Curved Path Following with Fixed-wing UAVs,” *American Control Conference (ACC)*, Chicago, IL, July 2015.

[84] Liang, Y. and Jia, Y., “Tangent Vector Field Approach for Curved Path Following with Input Saturation,” *Systems & Control Letters*, Vol. 104, June 2017, pp. 49–58.

[85] Lee, S., Lee, S., Ann, S., Lee, J., and Kim, Y., “Vector Field Based Guidance Law for Intercepting Maneuvering Target,” *26th Mediterranean Conference on Control and Automation (MED)*, Zadar, Croatia, June 2018.

[86] Holmes, M. H., *Introduction to Perturbation Methods*, Springer, New York, NY, 2013.

[87] Holmes, M. H., *Introduction to the Foundations of Applied Mathematics*, Vol. 56 of *Texts in Applied Mathematics*, Springer, New York, NY, 2009.

[88] Khalil, H., *Nonlinear Control*, Always Learning, Pearson, Upper Saddle River, NJ, 2015.

[89] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Vol. 199, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[90] Natanson, I. P., *Constructive Function Theory: Uniform Approximation*, Vol. 1, Ungar, New York, NY, 1964.

[91] Doha, E., Bhrawy, A., and Saker, M., "On the Derivatives of Bernstein Polynomials: An Application for the Solution of High Even-Order Differential Equations," *Boundary Value Problems*, Vol. 2011, No. 1, 2011, pp. 1–16.

[92] Bhat, S. and Bernstein, D., "Continuous Finite-time Stabilization of the Translational and Rotational Double Integrators," *IEEE Transactions on Automatic Control*, Vol. 43, No. 5, 1998, pp. 678–682.

[93] Bhat, S. P. and Bernstein, D. S., "Finite-Time Stability of Continuous Autonomous Systems," *SIAM Journal on Control and Optimization*, Vol. 38, No. 3, 2000, pp. 751–766.

[94] Hong, Y., "Finite-time Stabilization and Stabilizability of a Class of Controllable Systems," *Systems & Control Letters*, Vol. 46, No. 4, 2002, pp. 231–236.

[95] Lan, Q., Li, S., Khoo, S., and Shi, P., "Global Finite-time Stabilisation for a Class of Stochastic Nonlinear Systems by Output Feedback," *International Journal of Control*, Vol. 88, No. 3, 2015, pp. 494–506.

[96] Su, Y. and Zheng, C., "Robust Finite-time Output Feedback Control of Perturbed Double Integrator," *Automatica*, Vol. 60, Oct. 2015, pp. 86–91.

[97] Li, F. and Liu, Y., "Global Finite-time Output-feedback Stabilization for Uncertain Nonlinear Systems with Unknown Growth Rate," *35th Chinese Control Conference (CCC)*, Chengdu, China, July 2016.

[98] Gao, J. and Cai, Y.-l., "Three-Dimensional Impact Angle Constrained Guidance Laws with Fixed-Time Convergence," *Asian Journal of Control*, Vol. 19, No. 6, 2017, pp. 2240–2254.

[99] Song, Y., Wang, Y., and Krstic, M., "Time-varying Feedback for Stabilization in Prescribed Finite Time," *International Journal of Robust and Nonlinear Control*, Vol. 29, No. 3, Mar. 2018, pp. 4–8.

[100] Zhou, D., Sun, S., and Teo, K. L., "Guidance Laws with Finite Time Convergence," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 6, 2009, pp. 1838–1846.

[101] Sun, S., Zhou, D., and Hou, W. T., "A Guidance Law with Finite Time Convergence Accounting for Autopilot Lag," *Aerospace Science and Technology*, Vol. 25, No. 1, 2013, pp. 132–137.

[102] Zhang, Y., Sun, M., and Chen, Z., "Finite-time Convergent Guidance Law with Impact Angle Constraint Based on Sliding-mode Control," *Nonlinear Dynamics*, Vol. 70, No. 1, 2012, pp. 619–625.

[103] Si, Y. and Song, S., "Adaptive Reaching Law Based Three-dimensional Finite-time Guidance Law Against Maneuvering Targets with Input Saturation," *Aerospace Science and Technology*, Vol. 70, Nov. 2017, pp. 198–210.

[104] Lee, S., Kim, Y., and Song, C., "Impact Angle Control Guidance of Glide-Capable Munition Using a Vector Field Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Published Online, 2020.

[105] Zhao, P., Chen, W., and Yu, W., "Optimal Perturbation Guidance with Constraints on Terminal Flight-path Angle and Angle of Attack," Vol. 233, No. 12, 2019, pp. 4436–4446.

[106] Farouki, R. T. and Sakkalis, T., "Real Rational Curves Are Not 'Unit Speed'," *Computer Aided Geometric Design*, Vol. 8, No. 2, 1991, pp. 151–157.

[107] Farouki, R. T., "Optimal Parameterizations," *Computer Aided Geometric Design*, Vol. 14, No. 2, 1997, pp. 153–168.

[108] Jüttler, B., "A Vegetarian Approach to Optimal Parameterizations," *Computer Aided Geometric Design*, Vol. 14, No. 9, 1997, pp. 887–890.

[109] Farouki, R. T., *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*, Vol. 1 of *Geometry and Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[110] Fliess, M., Lévine, J., Martin, P., and Rouchon, P., "Flatness and Defect of Non-linear Systems: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, No. 6, 1995, pp. 1327–1361.

[111] Tekin, R. and Erer, K. S., "Impact Time and Angle Control Against Moving Targets with Look Angle Shaping," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 5, may 2020, pp. 1020–1025.

Appendix

Calculations of Partial Derivatives of Trajectory Functions

Figure A.1 shows the coordinates in two dimensional engagement. The (X, Y) components of the $XY-$coordinates depend on the function f. The relation between the xy-coordinate and the XY-coordinate is defined as follows,

$$\frac{\partial f}{\partial X} = f_X = \frac{\partial f}{\partial x}\frac{\partial x}{\partial X} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial X} \tag{A.1}$$

$$\frac{\partial f}{\partial Y} = f_Y = \frac{\partial f}{\partial x}\frac{\partial x}{\partial Y} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial Y} \tag{A.2}$$

Equations (A.1) and (A.2) can be rewritten in matrix form as

$$\begin{bmatrix} f_X \\ f_Y \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial y}{\partial X} \\ \frac{\partial x}{\partial Y} & \frac{\partial y}{\partial Y} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} \triangleq R \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{A.3}$$

In Eq. (A.3), the transformation matrix R is a rotational matrix with the rotation angle α, which is shown in Fig. A.1, as

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \tag{A.4}$$

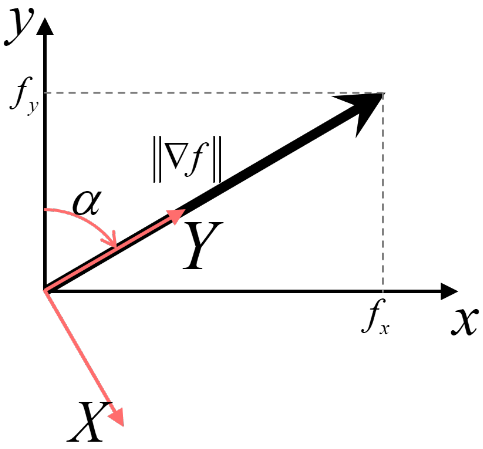


Figure A.1: Rotation angle between the xy and XY coordinates.

where

$$\sin\alpha = \frac{f_x}{\|\nabla f\|} \tag{A.5}$$

$$\cos\alpha = \frac{f_y}{\|\nabla f\|} \tag{A.6}$$

Note that $\|\nabla f\| = \sqrt{f_x^2 + f_y^2}$.

From Eqs. (A.3) - (A.6), the partial derivatives of f in XY-coordinate can be derived as follows,

$$\begin{aligned} \begin{bmatrix} f_X \\ f_Y \end{bmatrix} &= \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix} \|\nabla f\| \\ &= \begin{bmatrix} \cos\alpha\sin\alpha - \cos\alpha\sin\alpha \\ \sin^2\alpha + \cos^2\alpha \end{bmatrix} \|\nabla f\| = \begin{bmatrix} 0 \\ \sqrt{f_x^2 + f_y^2} \end{bmatrix} \end{aligned} \tag{A.7}$$

Differentiating Eq. (A.7) with respect to the components of the $XY-$coordinates yields

$$f_{XX} = 0, \quad f_{XY} = 0 \tag{A.8}$$

$$f_{YX} = 0, \quad f_{YY} = \frac{\partial f_Y}{\partial x}\frac{\partial x}{\partial Y} + \frac{\partial f_Y}{\partial y}\frac{\partial y}{\partial Y} \tag{A.9}$$

Partially differentiating f_Y from Eq. (A.7) in XY-coordinate, the following equations are obtained.

$$\frac{\partial f_Y}{\partial x} = \frac{\partial(\sqrt{f_x^2 + f_y^2})}{\partial x} = \frac{1}{2}\left(f_x^2 + f_y^2\right)^{-1/2}\left(2f_x f_{xx} + 2f_y fxy\right) = \frac{1}{f_Y}\left(f_x f_{xx} + f_y f_{xy}\right) \tag{A.10}$$

$$\frac{\partial f_Y}{\partial y} = \frac{1}{f_Y}\left(f_x f_{xy} + f_y f_{yy}\right) \tag{A.11}$$

Finally, using Eqs. (A.4)-(A.6) and Eqs. (A.10)-(A.11) in Eq. (A.9), f_{YY} can be obtaind as follows,

$$f_{YY} = \frac{1}{f_Y^2}\left(\left(f_x f_{xx} + f_y f_{xy}\right) f_x + \left(f_x f_{xy} + f_y f_{yy}\right) f_y\right) \tag{A.12}$$

국문초록

본 논문에서는 유도탄에 적용가능한 최적 경로계획 기법과 경로추종 유도법칙을 결합한 프레임워크에 대해 다룬다. 첫째로, 3차원 유도탄 운동학 방정식을 시간스케일에 따라 분할하여 두 개의 동역학 시스템으로 분할한다. 시간 스케일에 따라 동역학 시스템을 분할함으로써 경로를 계획하고 분석하기가 용이해진다. 시간 스케일에 따라 분할된 유도탄의 3차원 동역학 시스템은 이중루프 구조를 갖게 된다. 내부루프는 빠른 시간 스케일을 갖는 시스템으로 명령추종을 위한 제어기가 설계된다. 내부루프는 외부루프에 비해 매우 빠른 시간 스케일을 갖고 있으므로 내부루프 상태변수 오차는 매우 빠르게 0으로 수렴한다. 따라서 느린 시간 스케일에서의 내부루프 상태변수는 외부루프의 제어입력 변수로 간주할 수 있다. 한편, 외부루프는 상대적으로 느린 시간 스케일을 갖는 시스템으로, 외부루프 시스템에 대해 경로계획 기법이 수행된다. 본 논문에서 다루는 시스템 출력 궤적 설계 기법을 이용하여 외부루프 시스템 상태변수의 최적궤적을 얻는다.

본 논문에서 제안하는 프레임워크의 두 번째 부분은 최적 시스템 출력 궤적 설계 알고리즘이다. 이 기법에서는 비선형 단일입력 단일출력 시스템을 고려하며, 매개변수화된 곡선을 사용하여 궤적최적화를 구현한다. 매개변수화된 곡선으로는 다항함수의 일종인 베지어 곡선을 선택하였다. 베지어 곡선은 직관적으로 이해하기 쉬운 구조를 가지고 있기 때문에 동역학 시스템의 궤적을 성형하기에 유용하다. 본 논문에서는 다양한 예시를 통해 제안한 시스템 출력 성형 알고리즘의 유용성을 보인다.

마지막으로 본 논문에서는 벡터필드 기반 유도기법을 다룬다. 벡터필드 기반 유도기법은 2차원 혹은 3차원 공간상에서 적용가능한 경로추종 유도기법이다. 추

종하고자 하는 곡선은 음함수 형태나 매개변수화된 곡선 형태로 주어질 수 있는데, 본 논문에서는 두 가지 형태의 곡선을 모두 다루어 벡터필드 기반 유도법칙을 설계한다. 본 논문에서는 벡터필드 기반 유도기법의 수렴특성에 대해 분석하며, 세 가지 서로 다른 수렴특성을 보이는 벡터필드를 설계 및 분석한다.

본 논문에서 제안한 프레임워크를 따라 3차원 유도탄 모델에 대한 수치 시뮬레이션을 수행하여 제안한 기법들의 성능을 검증한다. 한편, 최적 시스템 출력 궤적 설계 기법 및 벡터필드 기반 유도기법 등은 유도탄 동역학 시스템뿐만 아니라 다양한 비선형 시스템에 대해 적용될 수 있다.

주요어: 시간스케일 분리, 궤적최적화, 매개변수화된 곡선, 궤적추종 유도, 벡터필드 유도, 비선형 시스템 제어, 점근안정성, 유한시간 수렴성

학번: 2015-20785