2016

A New Path Planning Guidance Law For Improved Impact Time Control of Missiles and Precision Munitions

Mark Snyder
University of Central Florida

A NEW PATH PLANNING GUIDANCE LAW FOR IMPROVED IMPACT TIME CONTROL OF MISSILES AND PRECISION MUNITIONS

by

MARK G. SNYDER
M.S. University of Central Florida, 2009

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2016

Major Professor: Zhihua Qu

ABSTRACT

A new missile guidance law is proposed for the control of impact time which provides an improved time-to-go calculation by removing error due to trajectory curvature and also provides a family of trajectories for trajectory planning purposes. Unlike conventional optimal guidance laws, the proposed law is non explicit in time-to-go and the linearization of the engagement kinematics in order to gain a closed form solution is not necessary.

To my mother and father who stood behind me throughout it all, and to my companion and best friend Holly, this is what all of my bad moods were about.

ACKNOWLEDGMENTS

I would like to thank Dr. Zhihua Qu for his seemingly infinite amount of patience and support throughout this long and arduous process. I would also like to thank Dr. Richard Hull for his guidance and help in keeping this work on track.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Missiles play a central role in modern warfare operations and provide one of the most versatile and flexible weapons platforms in existence. From short range air-to-air to medium and long range ballistic applications, missile systems are capable of neutralizing a wide range of threats. As technology advances, threat capability becomes increasingly evasive with more effective counter measures. Modern missile technology must be able to effectively outperform threat capability under a wide range of battlefield scenarios.

It has become apparent that the next generation of missile technology must be able to act in a cooperative or coordinated manner in order to maintain an effective offensive or defensive role. The term cooperative in this context is used rather loosely but the technical implications are far reaching. Here, the term cooperative is used to infer that multiple missiles work together as a team in order to neutralize a target. Multiple missiles sharing guidance related data on a target have a greater probability of achieving a target kill when the target is being actively sensed from multiple missile vantage points.

The enhancement of target state estimation as well as target motion prediction is an encouraging possibility. The applications of such a weapons system are immediate, such as the simultaneous arrival in order to overwhelm defensive countermeasures as in the case of ship Close-in Weapons Systems (CIWS). Simultaneous arrival on a CIWS system can completely overwhelm the system and greatly increase the probability of a direct strike on the intended target. In other scenarios, a delayed impact time with specified arrival interval between missiles is desirable as in the case of bunker neutralization. Missiles can arrive one-by-one in short delayed time intervals to maximize the explosive effect of the preceding warhead and pulverize the target.

The technical challenges of a cooperative guidance approach, however, are immense. Any co-

operative engagement is going to require control, in some way, of the missiles arrival time at a particular location, which inherently requires the calculation of the time-to-go until impact. From a technical standpoint, this is very difficult since missile dynamics form an under-actuated system, meaning some aspects of the missiles operation cannot be controlled. In fact, most missiles have no control over the longitudinal velocity, making any coordinated effort to control a missiles timed approach to a target extremely difficult. This is why the term ”cooperative” is generally meaningless and is intended only for illustrative purposes to motivate the larger problem. The problem is in fact one of ”impact time control” first and foremost. Once a viable solution to the impact time control problem is found, any number of cooperative algorithms found in the open literature can be applied.

Without enhanced methods of impact time control, cooperation cannot be considered, and although work has been conducted in the last decade to further the problem, much of the work is based on simple linear models moving at constant velocity. The true problem formulation involves considerable nonlinearity, missile acceleration, and disturbances due to wind and other unmodeled behaviors.

The control of impact time for an under-actuated missile can be dealt with most effectively from a trajectory planning approach as long as improvements are made to the time-to-go calculation. If the computed time-to-go is accurate enough, then the trajectory planning can be employed to either increase or decrease the approach time to the target. This method is straight forward if the missile velocity is constant, or near constant. In this case, changes to the trajectory arc-length can correct for any error between a desired impact time and the currently calculated impact time. The vast majority of missiles, however, exhibit considerable changes in velocity which are difficult to model and inherently contain disturbances which further corrupts calculations for the time-to-go. Even in this case, an improved time-to-go calculation cannot take into account errors introduced by the missiles acceleration. Even though acceleration profiles can be modeled, some error will

always associated with such estimates.

In general, two types of error can be expected with the time-to-go calculation. The first is error due to trajectory curvature. All current methods of computing time-to-go involve the line-of-sight between the missile and target and the relative closing speed, generally referred to as the range-over-range-rate method. The calculation is updated several times a second during the engagement. This calculation method generates precise results as long as the missile is heading directly at the target, as is the case in a head-on intercept or tail chase scenario. However, in many engagement scenarios a considerable amount of trajectory curvature is possible as the missile maneuvers to make intercept. The line-of-sight based calculation does not account for curvature and a considerable error is possible. The second type of possible error in the range-over-range-rate method is due to missile accelerating along the trajectory, which is nearly impossible to compensate for unless the acceleration profile is perfectly known.

Considerable improvements can be made if the time-to-go could be calculated along the trajectory arc-length instead of the line-of sight. This removes the error due to curvature, leaving only the error due to acceleration which can be dealt with through atmospheric modeling. Formulating a time-to-go calculation in terms of arc-length is problematic for several reasons. First, a future known path must be established from the missiles current position to the target. This means integrating the guidance law into the future to find the missiles closest point of approach to the target and then numerically calculating the arc-length along the resulting path. However, iterative calculations of the arc-length are not possible in real-time due to the high computational cost as the time-to-go must be updated many times a second.

The only remaining option is to compute the time-to-go from a closed form solution of the trajectory arc-length. The problem with this approach is that few closed form solutions exist for trajectories of order higher than two. Certain approximations of arc-length can be made but then

error is reintroduced and little improvement is gained. An ideal solution to the problem would be to have a higher order family of trajectories for which an exact closed form solution for arc-length exists. With the existence of a family of trajectories, trajectory planning could be utilized in conjunction with the resulting improvement in time-to-go to gain better control over impact time, especially under rapidly changing velocity due to atmosphere and gravity.

The guidance law proposed here that aims to provide these desirable capabilities is called Quad-Segment-Polynomial-Trajectory, or QSPT for short. The idea behind QSPT is to utilize multiple second order polynomial trajectories joined together in a smooth, continuous fashion while leveraging the closed form solution to arc-length which is possible for second order polynomials. If enough second order segments are used, free design variables result, giving rise to a family of trajectories which can be used to adjust the impact time through trajectory planning. The resulting trajectory appears to exhibit higher order polynomial characteristics yet has a closed form solution to arc-length, thus removing error in the time-to-go calculation due to curvature. In addition, since QSPT is a defined trajectory between the missile and target, estimating the acceleration due to atmosphere, drag, and gravity along the known path becomes a trivial process.

Some major contrasts can be made between QSPT and optimal guidance laws. The first pertains to how closed form solutions are obtained for the optimal guidance law. The engagement kinematics between a missile and target are highly nonlinear as will be shown in chapter 6. In order to solve the resulting optimal control problem, a solution for the resulting two-point boundary value problem must be found but due to the nonlinearities involved, no closed form solution exists and a numerical solution is once again too computationally costly. In order to gain a closed form solution the engagement kinematics are linearized and in the process, valuable fidelity is lost in the process which can result in poor performance under certain real-time conditions. In addition, many optimal guidance laws are explicit in time-to-go and error in the calculation can also induce poor performance.

In contrast, QSPT does not require the linearization of the engagement kinematics and fully utilizes the nonlinear kinematics in the final form of the guidance law. The QSPT guidance law is also non-explicit in time-to-go which is not generally the case with most optimal guidance laws. Guidance parameters are derived from the QSPT trajectory and are used as input to the guidance law. The actual trajectory asymptotically converges to, and remains on the QSPT generated trajectory. The time-to-go computation of QSPT is based on an exact closed form calculation of the arc-length and is free from error due to curvature. This coupled with the family of trajectories generated by QSPT provides better control over the impact time than with conventional means.

This document is organized as follows. In chapter 2 a search of the existing work in the open literature is given. The literature survey covers a broad range of guidance law work that is segregated into 7 groups. In chapter 3, a general overview of missile types, the phases of flight, and the basic components and subsystems of a missile are given. In chapter 4, equations of motion common to both standard guidance laws as well as the guidance law derivation for QSPT is given. The highly nonlinear dynamic equations of motion are simplified into nonlinear kinematic models and further into linear basic models commonly found in simplified guidance derivations. Chapter 5 discusses the basic types of performance indices commonly used to derive optimal guidance laws. Chapter 6 provides a comprehensive development of the optimal guidance law and at the end of that chapter a detailed discussion of the various computation methods for time-to-go is given. In chapter 7 the problem is formulated and the kinematic equations of motion to be used in the guidance law synthesis is presented. Chapter 8 introduces QSPT and a trajectory set with specified boundary conditions is derived. Chapter 9 derives both the open and closed loop guidance laws for QSPT and chapter 10 details the optimization of a QSPT trajectory. Chapter 11 derives the improved arc-length based time-to-go algorithm and discusses the procedure involved in pre-flight analysis. Chapter 13 details the Genex guidance law which is used in performance comparisons against QSPT. Chapter 14 shows the results obtained from a simulation of the performance of QSPT. The

QSPT time-to-go calculation versus the standard range-over-range-rate calculation is compared, as well as robustness of QSPT, effectiveness of the impact time control, and a full performance comparison with the Genex guidance law in the presence of noise. In the next chapter, a search of the existing work in the open literature is given.

CHAPTER 2: LITERATURE REVIEW

Searching the open literature, we can segregate existing work into 7 groups. In group 1, general references can be found on classical and modern guidance laws in [1, 2, 3, 4]. Each of these references provide a clear and basic understanding of the general guidance problem, equations of motion, and missile components and subsystems.

In group 2, we find works of [5, 6, 7] which formulate optimal guidance laws with constraints on impact angle. In these works, missile velocity is considered constant and linearized models of either the missile kinematics or engagement kinematics are used. In these works, no consideration is given to the control of impact time and the standard methods of computing time-to-go are utilized. In fact, while numerous papers are found concerning optimal guidance with terminal angle constraints, very few works pertaining to impact time constraints can be found. This comprises group 3. In this group, reference [8] formulates an optimal guidance law with impact angle as well as impact time constraints and [9] considers a time-optimal formulation with impact angle constraints. In each of these works, missile velocity is constant and linearized models are used in the development. The work in [10] considers an optimal formulation for just impact time control alone. An interesting note about this work is the independent variable in the state model is in terms of distance instead of time. This is done to avoid impact time from explicitly becoming a terminal boundary condition since it is an unknown parameter. However, missile velocity is considered constant and the model used is linear. Reference [11] considers time-of-flight control as well as range extension for precision munitions. Atmosphere and disturbances are considered in this work which make is well suited for some general performance comparisons in the simulation study. However, [11] uses range-over-missile-velocity for calculating the time-to-go.

Attempts to improve estimates of time-to-go for both classical and modern guidance laws comprise

group 4, [12, 13, 14, 15, 16, 17, 18]. In most of these works, linear models are used in the formulation and missile velocity and acceleration profiles are assumed to be fully known or modeled with linear and or linear piecewise equations. No in-flight correction due to disturbance is proposed. The work of [12] attempts to improve on this by deriving a recursive algorithm to correct the time-to-go estimate when either a heading angle error is present or changes in path length occurs. It does not take into account changes in time-to-go due to drag or atmospheric disturbances. The change in path length, however, is estimated from the deviation from a straight line path.

The works contained in group 5, [19, 20, 21], considers the development of optimal guidance laws when the missile velocity is not constant. In reference [20], the derived guidance law relies on predicted velocity profiles which can be updated to compensate for error. However, the law is derived using linearized models and is explicit in a terminal time boundary constraint.

Searching the literature further, we find scarce results for group 6; cooperative guidance laws for missiles. In references [22] and [23], optimal formulations are derived to address the issue of cooperative salvo attack on a target, however, both works assume constant velocity and in [23], the time-to-go estimation methods of [12] are utilized. In [22], attempts to estimate time-to-go are done using an arc-length approximation of the expected engagement. The resulting equation is approximated since closed form solutions for arc-length are impossible to derive in many cases. In [24], a discussion of cooperative missile research work and the problems associated with it are addressed.

In group 7, a path planning approach to guidance synthesis is taken. Reference [26] takes a somewhat similar approach as QSPT Guidance in that distance is segmented and a spline-based trajectory is developed. However, this work uses the standard range-over-missile-velocity for time-to-go calculation and the terminal boundary constraints are a function of the impact time.

CHAPTER 3: MISSILE PHASES OF FLIGHT

In this chapter a brief outline of the different phases of flight and guidance systems that dominate for that particular phase. Finally, a brief description of a missiles subsystems such as guidance and flight controls are given.

Depending on the type of missile system considered, three phases of flight generally exist which require different guidance schemes for each. In the case of short range missiles, the distance from the launch point of the missile to the target is typically small, and a single phase of flight is all that is required to reach the target. Differing phases of flight certainly result for longer range missiles and precision munitions. The distances involved require robust boost phase to generate enough kinetic energy to reach the target long after the boost phase has ended.

Boost Phase

The boost phase is designed to generate enough missile kinetic energy to reach the target long after the boost phase has ended. In the case of certain precision guided munitions, the boost phase powers the warhead to high altitudes and the potential and kinetic energy accumulated during this phase is enough to enable the warhead to reach long downrange distances to a target. The high levels of longitudinal acceleration make guiding the missile a challenge, and so the primary objective for these weapons is to achieve a desired altitude at burnout. In the case of cruise missiles, the boost phase is relatively short and is designed to generate an adequate forward velocity for the missile.

Midcourse Phase

The midcourse phase begins directly after the end of the boost phase and guides the missile over long distances towards a target. One example of a midcourse guidance law is TERCOM, or Terrain Contour Matching, which is used primarily in cruise missile applications. The primary objective in midcourse guidance is to guide the missile to a location close enough to the target for the homing phase to take over. This is generally called target acquisition in which active radar seekers acquire and lock on to the target. Once this is accomplished, the homing phase can begin.

Homing Phase

In this phase, the guidance objective becomes one that reduces the miss distance between the missile and the target while expending as little control energy as possible. The radar seeker is returning range and range rate data to the guidance computers and a Latax or lateral acceleration steering command is generated by the guidance algorithm to steer the missile in a manner that reduces the miss distance. Many different types of guidance laws for use in the homing phase can be found in the open literature.

Basic Subsystems of the Missile

Central to missile performance is the flight control and guidance system as shown in figure 3.1. The flight control is responsible for two performance objectives; stabilizing the body rates, and controlling lateral acceleration. The guidance system receives target related data from (in some cases) on-board active radar or infrared sensors and computes a command to steer the missile towards the target. The steering command is generally, for skid-to-turn missiles, a lateral acceleration, more

commonly referred to as latax. Therefore, the guidance system calculates a desired latax command and the flight control attempts to track this command while stabilizing the body rates. Care must be taken to ensure the guidance system does not over-command the missile. Every missile has a limit to its maneuvering capability and the potential for the guidance system to issue commands the missile cannot achieve is a possibility unless constraints are imposed in the guidance design.

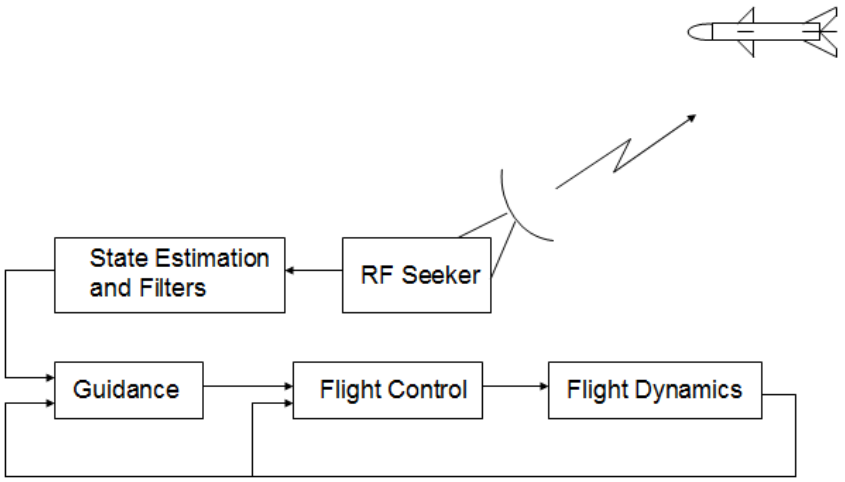

Figure 3.1: Basic components of a missile system

The target state signals generated by the RF seekers are typically corrupted by noise which must be removed by the target state estimators and filters before reaching the guidance processor. In the next chapter, a standard set of dynamic equations representing a 6 dof missile are presented and systematically simplified for use in both standard guidance formulations as well as the QSPT derivation in future chapters.

CHAPTER 4: EQUATIONS OF MOTION FOR GUIDANCE SYNTHESIS

In the open literature, many suitable models for the equations of motion of a body can be found. All of the models, regardless of how simplistic they may be, are derived from highly nonlinear equations of motion which represents the translational, rotational, and aerodynamic properties of the body in its configuration space. For most guidance law derivations, the use of nonlinear equations of motion do not yield closed form solutions to the resulting two-point boundary value problem. In fact, only linear equations of motion will yield such closed forms. This issue will be addressed in considerable detail in chapter 6. For the time being, we will derive several different models which are useful for conducting simulation studies on air vehicles. The 6 dof nonlinear equations of motion for a missile or aircraft are as follows

$$\dot{u} = F_x + g_x + rv - qw$$

$$\dot{v} = F_y + g_y + rv - qw$$

$$\dot{w} = F_z + g_z + rv - qw$$

$$\dot{p} = \frac{I_{zz}L + I_{xz}N - \left(I_{xz}\left(I_{yy} - I_{xx} - I_{zz}\right)p + \left(I_{xz}^2 + I_{zz}\left(I_{zz} - I_{yy}\right)r\right)q\right)}{I_{xx}I_{zz} + I_{xz}^2}$$

$$\dot{q} = \frac{\left(M - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right)}{I_{yy}}$$

$$\dot{r} = \frac{I_{xz}L + I_{xx}N - \left(I_{xz}\left(I_{yy} - I_{xx} - I_{zz}\right)r + \left(I_{xz}^2 + I_{xx}\left(I_{xx} - I_{yy}\right)p\right)q\right)}{I_{xx}I_{zz} + I_{xz}^2}$$

$$
\begin{aligned}
\dot{x}_I &= (\cos\gamma\cos\chi)\,u + (-\cos\phi\sin\chi + \sin\phi\sin\gamma\cos\phi)\,v + (\sin\phi\sin\chi + \\
&\quad + \cos\phi\sin\gamma\cos\phi)\,w \\
\dot{y}_I &= (\cos\gamma\sin\chi)\,u + (\cos\phi\cos\chi + \sin\phi\sin\gamma\sin\phi)\,v + (-\sin\phi\cos\chi + \\
&\quad + \cos\phi\sin\gamma\sin\phi)\,w \\
\dot{z}_I &= (\sin\gamma)\,u + (\sin\phi\cos\gamma)\,v + (\cos\phi\cos\gamma)\,w \\
\dot{\phi} &= p + (q\sin\phi + r\cos\phi)\tan\gamma \\
\dot{\gamma} &= q\cos\phi - r\sin\phi \\
\dot{\chi} &= \frac{(q\sin\phi + r\cos\phi)}{\cos\gamma},
\end{aligned}
\tag{4.1}
$$

where the following variables are defined as

$\begin{bmatrix} u & v & w \end{bmatrix}^T$ missile velocity in body coordinates

$\begin{bmatrix} p & q & r \end{bmatrix}^T$ missile rotational rates in body coordinates (roll rate, pitch rate, yaw rate)

$\begin{bmatrix} x_I & y_I & z_I \end{bmatrix}^T$ missile position in inertial coordinates

$\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T$ forces in body coordinates (x_B, y_B, z_B respectively)

$\begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T$ moments in body coordinates (Roll moment, pitch moment, yaw moment)

$\begin{bmatrix} \phi & \gamma & \chi \end{bmatrix}^T$ roll position, flight path angle, heading angle with sideslip and angle-of-attack

given as

$$
\beta_A = \sin^{-1}\left(\frac{v_A}{V_A}\right) \qquad\qquad \alpha_A = \tan^{-1}\left(\frac{w_A}{u_A}\right) \tag{4.2}
$$

and gravity components, velocity, wind shear, and forces and moments,

$$\begin{aligned} g_x &= -g \sin\gamma \\ g_y &= g \sin\phi \cos\gamma \\ g_z &= g \cos\phi \cos\gamma \\ V_A &= \sqrt{u_A^2 + v_A^2 + w_A^2}, \end{aligned} \tag{4.3}$$

$$\begin{bmatrix} u_A \\ v_A \\ w_A \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \mathbf{H}_\mathbf{I}^\mathbf{B} \bar{\mathbf{w}}, \tag{4.4}$$

$$\begin{aligned} F_x &= C_x \bar{q} S + T_x \\ F_y &= C_y \bar{q} S + T_y \\ F_z &= C_z \bar{q} S + T_z \\ M_x &= C_l \bar{q} S + M_{T_x} \\ M_y &= C_m \bar{q} S + M_{T_y} \\ M_z &= C_n \bar{q} S + M_{T_z}. \end{aligned} \tag{4.5}$$

Matrix $\mathbf{H}_\mathbf{I}^\mathbf{B}$ is the inertial to body coordinate transformation and $\bar{\mathbf{w}}$ is the vector of wind-shear components. If we assume quiescent atmosphere with no wind, $\bar{\mathbf{w}} = 0$, then

$$\begin{bmatrix} u_A \\ v_A \\ w_A \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{4.6}$$

The dynamic equations of motion given by (4.1) are highly nonlinear and coupled both dynamically and aerodynamically. In other words the rotational equations for $\dot{p}, \dot{q}$, and $\dot{r}$ are cross-coupled with one another as well as coupled into the translational velocity equations of $\dot{u}, \dot{v}$, and $\dot{w}$. In addition, the aerodynamic expressions embedded in both the rotational and translational dynamics are themselves cross-coupled with one another. In order to gain a closed form solution for guidance law development, considerable simplifications must be made.

The figures that follow illustrate the relationship between the body coordinates and the fixed inertial coordinates as well as the relationship between the velocity components, incidence angles, and moments. In figure 4.1, the relationship between the body coordinates and the velocity vector through the incident angles are shown. Figure 4.2 shows the moments that relate to the body coordinate system. The moments cause the body axis system to rotate out of alignment with the fixed inertial system through the Euler angles of ϕ, γ, and χ as shown in figure 4.3. In the subsection that follows, a progressive simplifications are made to the 6 dof nonlinear model in order to gain more simplified equations to work with.

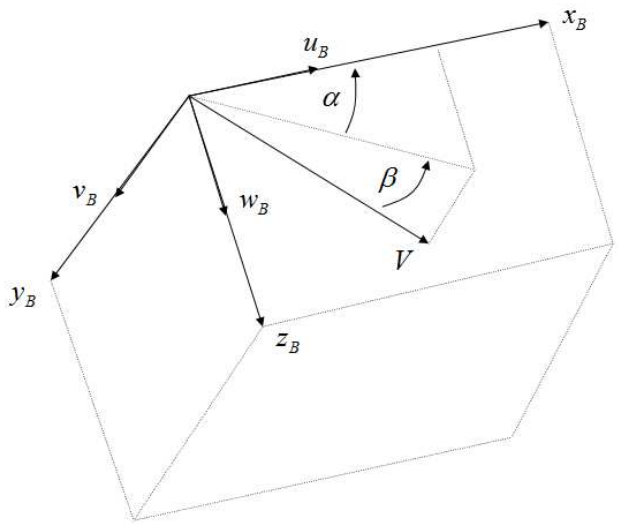

Figure 4.1: Aerodynamic incident angles in relation to body coordinates

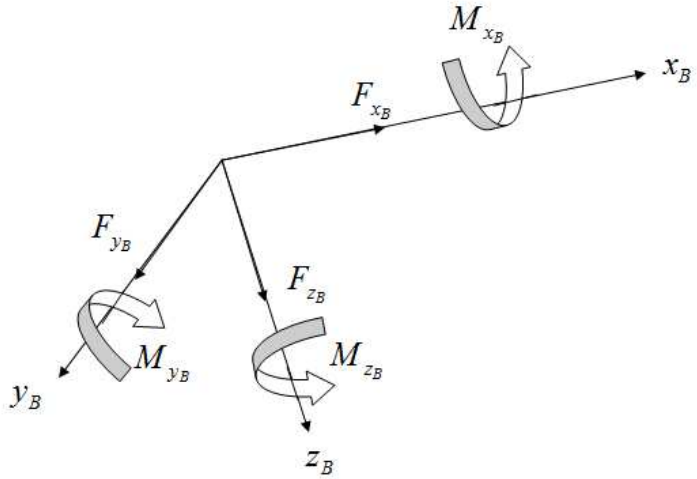

Figure 4.2: Forces and moments in relation to body coordinates

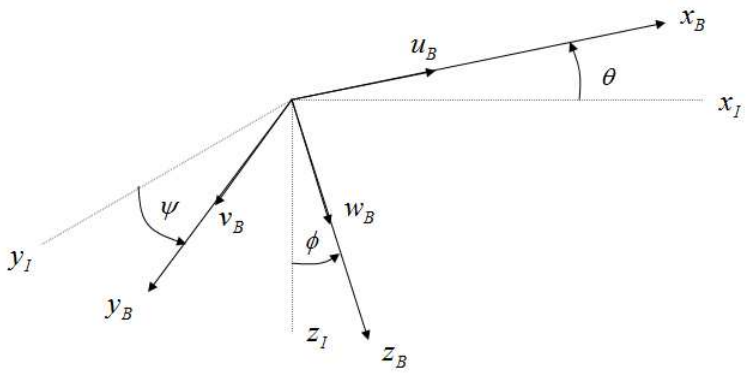

Figure 4.3: Inertial angles in relation to body coordinates

Simplified Nonlinear Model

The first simplifying assumptions we can make on (4.1) is to consider

- Roll angle $\phi = 0$, roll rate $\dot{\phi} = 0$
- Incidence angles $\beta = 0, \quad \alpha = 0$
- Assume missile has tetragonal symmetry, all inertial cross-coupling terms vanish, that is $I_{xz} = 0$.

Assuming a zero roll angle and roll angle rate decouples a considerable portion of the dynamic equations. An assumption of zero incidence angles also removes the dependency on aerodynamic parameters which must be estimated using software such as Missile Datcom or obtained through wind tunnel testing. Because $\beta = 0, \quad \alpha = 0$, we can immediately say that $u = V_A, \quad v = 0 \quad w = 0$. This assumption indicates that the velocity vector V_A remains aligned with the missiles longitudinal

axis. Additionally, most missiles exhibit tetragonal symmetry which eliminates off-diagonal terms in the moment of inertial tensor. These simplifying assumptions lead directly to the nonlinear kinematic model given as

$$\begin{aligned}
\dot{V} &= F_{x_B} + g_x \\
\dot{q} &= \frac{M_y}{I_{yy}} \\
\dot{r} &= \frac{M_z}{I_{zz}} \\
\dot{x}_I &= V \cos\gamma \cos\chi \\
\dot{y}_I &= V \cos\gamma \sin\chi \\
\dot{z}_I &= V \sin\gamma \\
\dot{\gamma} &= q \\
\dot{\chi} &= \frac{r}{\cos\gamma}.
\end{aligned} \tag{4.7}$$

It should be apparent that even though zero incidence angles were assumed, some aerodynamic related components still remain. In this case, F_{x_B} is the total longitudinal drag, and the moments remain primarily as a result of their dependence on control surface deflections. This model is useful for preliminary design of flight control systems.

Simplified Nonlinear Model without Aero Parameters

Next, we can assume the missile to be a point mass. Under this assumption, the moments vanish as

$$M_y = 0, \quad M_z = 0, \tag{4.8}$$

and therefore the remaining body rates become

$$\dot{q} = 0, \quad \dot{r} = 0. \tag{4.9}$$

This results in equation

$$\begin{aligned} \dot{V} &= F_{x_B} + g_x \\ \dot{x}_I &= V \cos\gamma \cos\chi \\ \dot{y}_I &= V \cos\gamma \sin\chi \\ \dot{z}_I &= V \sin\gamma \\ \dot{\gamma} &= q \\ \dot{\chi} &= \frac{r}{\cos\gamma} \end{aligned} \tag{4.10}$$

The last two equations in (4.10) imply that under the previous assumption, the body rates q and r become the inertial rates $\dot{\theta}$ and $\dot{\psi}$. The question remains, how is this model controlled? With the vanishing of the moments, the control surface deflections that used to control the motion are now gone. We can solve this problem by deriving some simple equations to eliminate q and r. First, we consider two orthogonal planes, xy and xz, where the forces due to control and gravity are applied and sum the forces in each plane as,

$$\begin{aligned} F_{xy_{applied}} + F_{xy_{gravity}} &= m\sqrt{\ddot{x}^2 + \ddot{y}^2} = m\sqrt{V^2 \sin^2(\chi)\,\dot{\chi}^2 + V^2 \cos^2(\chi)\,\dot{\chi}^2} \\ F_{xz_{applied}} + F_{xz_{gravity}} &= m\sqrt{\ddot{x}^2 + \ddot{z}^2} = m\sqrt{V^2 \sin^2(\gamma)\,\dot{\gamma}^2 + V^2 \cos^2(\gamma)\,\dot{\gamma}^2}. \end{aligned} \tag{4.11}$$

Substitution of $\dot{\chi} = r$ and $\dot{\gamma} = q$, and taking the square root gives

$$\begin{aligned} m^{-1}\left(F_{xy_{applied}} + F_{xy_{gravity}}\right) &= Vr \\ m^{-1}\left(F_{xz_{applied}} + F_{xz_{gravity}}\right) &= Vq. \end{aligned} \tag{4.12}$$

and therefore the "pseudo controls" q and r are determined to be

$$\begin{aligned} r &= \frac{1}{V}\left(a_{xy_{applied}} + a_{xy_{gravity}}\right) \\ q &= \frac{1}{V}\left(a_{xz_{applied}} + a_{xz_{gravity}}\right) \end{aligned} \tag{4.13}$$

Substitution of the expressions for q and r gives

$$\begin{aligned} \dot{V} &= F_{x_B} + g_x \\ \dot{x}_I &= V\cos\gamma\cos\chi \\ \dot{y}_I &= V\cos\gamma\sin\chi \\ \dot{z}_I &= V\sin\gamma \\ \dot{\gamma} &= \frac{1}{V}\left(a_{xz_{applied}} + a_{xz_{gravity}}\right) \\ \dot{\chi} &= \frac{1}{V\cos\gamma}\left(a_{xy_{applied}} + a_{xy_{gravity}}\right) \end{aligned} \tag{4.14}$$

where the controls are defined to be $u_\chi = a_{xy_{applied}}$ and $u_\gamma = a_{xz_{applied}}$, and the gravity terms are

$a_{xy_{gravity}} = 0$, $a_{xz_{gravity}} = g\cos\gamma$. The final form of the equations of motion are given as

$$\begin{aligned}
\dot{V} &= F_{x_B} + g_x \\
\dot{x}_I &= V\cos\gamma\cos\chi \\
\dot{y}_I &= V\cos\gamma\sin\chi \\
\dot{z}_I &= V\sin\gamma \\
\dot{\gamma} &= \frac{1}{V}\left(u_\gamma + g\cos\gamma\right) \\
\dot{\chi} &= \frac{1}{V\cos\gamma}u_\chi.
\end{aligned} \tag{4.15}$$

The equations of motion given in (4.15) will be the equations used to derive the proposed guidance law in upcoming chapters. In the next subsection, however, we further simplify the equations of (4.15) into a completely linear system, which is useful for the optimal guidance derivations in chapter 6.

The Linearized Model

From the nonlinear kinematic model in (4.15), further simplifying assumptions can are made in order to reduce the equations into fully linear ones suitable for optimal guidance synthesis which will be derived in chapter *. For now, the following simplifying assumptions are made

- Constant velocity, $\dot{V} = 0$
- Small angles γ and χ
- Ignore gravity

- Planar motion - either xy or xz plane

In addition, the equations of (4.15) can be linearized with the use of t he small angle approximation, which is a direct result of the Taylor Series expansion. The following trigonometric functions have the approximated values under the assumption of small angle

$$\begin{aligned} \cos\gamma &\approx 1 - \frac{\gamma^2}{2} \approx 1 \\ \sin\gamma &\approx \gamma \\ \tan\gamma &\approx \gamma. \end{aligned} \tag{4.16}$$

Applying these assumptions results in

$$\begin{aligned} \dot{x}_I &= V & \dot{x}_I &= V \\ \dot{y}_I &= V\chi & \dot{z}_I &= V\gamma \\ \dot{\chi} &= \frac{1}{V}u_\chi & \dot{\gamma} &= \frac{1}{V}u_\gamma \end{aligned} \tag{4.17}$$

An immediate consequence of the small angle linearization is the fact that the planes of operation have been decoupled as shown in (4.17). This is a typical design procedure in the development of certain guidance laws in that planar motion is assumed and the guidance law is synthesized for the two individual planes, i.e. the altitude/downrange plane, and the crossrange/downragne plane. In the next chapter, performance indices common to guidance applications are discussed.

CHAPTER 5: PERFORMANCE INDICES

In this chapter a discussion of performance indices is given in order to motivate the process of "optimization". Many times it is claimed that some process or result is optimal, but one must ask, to what has this result or process been optimized with respect to? In other words, what is the criterion for which this claim of optimization can be made. In the case of missile flight, particularly the three phases previously discussed, time, fuel, control (or divert) energy, and terminal constraints such as miss distance and angle must be considered in the problem formulation. For instance, during the midcourse phase, the missile must reach the point of target acquisition as quickly as possible using as little fuel as possible. Therefore, in the design of the control law, these constraints are considered and the optimization process yields a guidance law optimized with respect to minimum time and fuel usage. In the case of the homing phase, certain terminal constraints must be achieved such as impact angle or impact time while at the same time keeping the divert energy required to achieve a zero miss distance as minimal as possible.

The primary objective in the optimization process is to specify some "performance index" which is related directly to the states of the dynamic system that must be controlled; i.e. control energy, fuel, time. The performance index, typically denoted as J, is itself minimized through a process to be discussed in the next section.

There are, in general, two types of performance indices common to missile guidance applications which appear in practical use with some minor variations of each for the more simplistic cases. The first type is given as

$$J = \int_{t_0}^{t_f} \left(\frac{1}{2} \bar{z} Q z + \frac{1}{2} \bar{u} R u \right) dt \tag{5.1}$$

where the over-bar indicates a transpose. This is a Lagrange-type performance index where only

an integral term is present. The basic premise behind this index is to take the weighted square of the state and the control integrated over the time span, and the find the control u such that J is minimized. The net affect is a balancing act between state excursions in z versus control excursions in u. The second type of performance index is given as

$$J = \frac{1}{2} \parallel z\left(t_f\right) \parallel^2 + \int_{t_0}^{t_f} \left(\frac{1}{2}\bar{z}Qz + \frac{1}{2}\bar{u}Ru\right) dt \tag{5.2}$$

This is known as a Bolza-type index. In this case, an extra term which considers the terminal state is included in addition to the original integral term. The integral portion of the index has the same function as that previously discussed. The addition of the terminal state penalty ensures that a specified terminal objective is met. In the case of missile guidance, some common terminal objectives are generally impact angle and zero miss distance. As stated previously, more simplistic variations of the first and second types can occur and often do. The following is an example of the second type in which only the control is under the integral term. Once again, in the minimization of this index, the control u is found which minimizes J. The control energy is thus kept to a minimum over the time span while the terminal objective is achieved.

$$J = \frac{1}{2} \parallel z\left(t_f\right) \parallel^2 + \frac{1}{2} \int_{t_0}^{t_f} \bar{u}Rudt. \tag{5.3}$$

The most basic performance index used in guidance applications is the result of further simplifying (5.3) where only the integral of the control squared remains as

$$J = \frac{1}{2} \int_{t_0}^{t_f} \bar{u}Rudt. \tag{5.4}$$

In this performance index, no terminal state constraint or terminal state penalty under the integral exists. The minimization of J results in the control u that minimizes the control energy over the time of flight. This is by far the simplest performance index that can be applied to find a closed

form guidance law. It is however devoid of the terminal constraints that are many times necessary in more advanced guidance applications. However, with increasing complexity of the performance index comes increasing difficulty is achieving a closed form solution, which is essential in guidance law development. As will be seen in the next chapter, even the most basic guidance law derivation can be a complex procedure.

Performance Index Weights

In (6.12), Q and R reflect the relative importance of achieving each objective; penalizing control effort over the time interval versus achieving the desired terminal state. For guidance laws designed for use in the homing phase, terminal constraints such as miss distance and the angle of impact are most important in achieving an effective target kill. Minimizing the miss distance is important for obvious reasons, however, specification of the terminal impact angle can maximize the lethality of the warhead, depending on the type of target. The weighting factors of the performance index would then place high priority on the terminal constraints while perhaps lowering the importance of the control energy expended during flight. In the next chapter, we will derive a basic optimal guidance law and note the importance of the control weights in the result.

CHAPTER 6: BASIC OPTIMAL GUIDANCE LAW

Formulation of the optimal guidance law is based on the performance index which has been chosen to achieve specific objectives, i.e., minimum control energy, desired terminal angle, zero terminal miss distance. The minimization of J, however, must result in a control u which is not only in closed form as specified in the last chapter, but which is feasible also. In other words, u must also satisfy the equations of motion. Many real life examples of this can be found. A hiker for instance, can find the minimum-time path to get off of a mountain by jumping over a ledge which drops vertically to the ground. Obviously, this is not a feasible path for the hiker to take and his path down the mountain should be constrained to one that is easy to traverse and will not kill him. The same concept is essentially true for a missile. Index J can be minimized but can the missile follow the path commanded by u?

In order to derive a guidance law that achieves the objectives of the performance index and results in a feasible control, motion constraints must be adjoined to the performance index and then certain conditions for optimality are then applied to find the optimal control u. The issue at hand is the fact that the optimal control is generally a function of the costate, which results from adjoining the dynamic constraints to the performance index. Aside from the original system dynamics, the adjoining process gives rise to additional costate dynamics which tend to be heavily coupled into the state. In order to solve for the control and eliminate the costate variables from the final solution, the two-point boundary value problem must be solved. Typically, only linear dynamics result in a closed form solution to this type of differential equation and hence the linear model derived in chapter 4 will now be leveraged.

Deriving the Engagement Dynamics

To begin the guidance law derivation, relative equations of motion between the missile and target must be formulated. The linear missile model of equation (4.17) is a key component and can can be used to describe target motion as well. We can define the following relative variables for the development as

- $p_r =$ relative distance between missile and target (miss distance)
- $v_r =$ relative velocity
- $a_r =$ relative acceleration
- subscripts M and T denote missile and target, respectively
- bold-face variables indicate vector quantities.

The relative dynamics are then expressed as

$$\begin{aligned} \mathbf{p_{r_{x,y,z}}} &= \mathbf{p_{T_{x,y,z}}} - \mathbf{p_{M_{x,y,z}}} \\ \mathbf{v_{r_{x,y,z}}} &= \mathbf{v_{T_{x,y,z}}} - \mathbf{v_{M_{x,y,z}}} \\ \mathbf{a_{r_{x,y,z}}} &= \mathbf{a_{T_{x,y,z}}} - \mathbf{a_{M_{x,y,z}}}, \end{aligned} \tag{6.1}$$

and therefore

$$\mathbf{p_{r_{x,y,z}}} = \begin{bmatrix} x_{I_T} \\ y_{I_T} \\ z_{I_T} \end{bmatrix} - \begin{bmatrix} x_{I_M} \\ y_{I_M} \\ z_{I_M} \end{bmatrix}, \mathbf{v_{r_{x,y,z}}} = \begin{bmatrix} v_{x_T} \\ v_{y_T} \\ v_{z_T} \end{bmatrix} - \begin{bmatrix} v_{x_M} \\ v_{y_M} \\ v_{z_M} \end{bmatrix}, \mathbf{a_{r_{x,y,z}}} = \begin{bmatrix} a_{x_T} \\ a_{y_T} \\ a_{z_T} \end{bmatrix} - \begin{bmatrix} a_{x_M} \\ a_{y_M} \\ a_{z_M} \end{bmatrix}. \tag{6.2}$$

From this, we define the state vector as

$$\zeta = \begin{bmatrix} \mathbf{p}_{\mathbf{r}_{\mathbf{x,y,z}}} & \mathbf{v}_{\mathbf{r}_{\mathbf{x,y,z}}} \end{bmatrix}, \tag{6.3}$$

with the corresponding state model for the relative dynamics

$$\dot{\zeta} = f(\zeta) + g(\zeta)\, u. \tag{6.4}$$

The standard performance index of

$$J = \frac{1}{2}\bar{\boldsymbol{\zeta}}(t_f)\, Q\boldsymbol{\zeta}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \bar{\mathbf{u}} R\mathbf{u} dt, \tag{6.5}$$

is used which imposes a terminal state constraint. As mentioned previously, a closed form solution to this problem is potentially not possible due to the nonlinearities present, specifically in $f(\zeta)$ and $g(\zeta)$.

We can, however, formulate (6.4) using the nonlinear kinematic model of (4.15) and then fully linearize the system in order to proceed with the development. From (4.15), the nonlinear relative equations of motion are given as

$$\begin{aligned} \ddot{x}_T - \ddot{x}_M &= \left(a_{T_{xy}} \cos\theta_T \sin\psi_T - a_{T_{xz}} \sin\theta_T \cos\psi_T\right) - \\ &\quad - \left(\cos\theta_M \sin\psi_M u_{xy} - \sin\theta_M \cos\psi_M u_{xz}\right) \\ \ddot{y}_T - \ddot{y}_M &= \left(-a_{T_{xz}} \sin\theta_T \sin\psi_T + a_{T_{xy}} \cos\theta_T \cos\psi_T\right) + \\ &\quad + \left(\sin\theta_M \sin\psi_M a_{M_{xz}} - \cos\theta_M \cos\psi_M a_{M_{xy}}\right) \\ \ddot{z}_T - \ddot{z}_M &= a_{T_{xz}} \cos\theta_T - a_{M_{xz}} \cos\theta_M. \end{aligned} \tag{6.6}$$

Linearizing equation (6.6) gives

$$\begin{aligned} a_{r_x} &= \ddot{x}_T - \ddot{x}_M = 0 \\ a_{r_y} &= \ddot{y}_T - \ddot{y}_M = a_{T_{xy}} - a_{M_{xy}} \\ a_{r_z} &= \ddot{z}_T - \ddot{z}_M = a_{T_{xz}} - a_{M_{xz}}, \end{aligned} \tag{6.7}$$

which is a chain of integrators. Some observations can be drawn from equation (6.7) which suggests several things,

- the xy, xz planes are decoupled by the linearization process
- the control action is approximately in the y and z directions and perpendicular to x
- the relative dynamics including miss distance variable p_r form a double integrator system.

The next assumption that is made is that target acceleration in (6.7) is zero. Then, we can assign the control u as the missile acceleration in the y and z directions as

$$\begin{aligned} a_{r_y} &= -\ddot{y}_M = -u_y \\ a_{r_z} &= -\ddot{z}_M = -u_z. \end{aligned} \tag{6.8}$$

The acceleration equations in (6.8) are decoupled from one another and therefore, the guidance synthesis can take place individually for each. We carry out this procedure for the xy plane only, as the process for the xz plane follows in an exact manner.

Considering the chain of integrators formed by (6.8), the state vector is defined as

$$\zeta = \begin{bmatrix} p_r & v_r \end{bmatrix} \tag{6.9}$$

where p_r and q_r are scalars and equal to $p_r = y_r, \quad v_r = v_{r_y}$. We arrive at the linear state space model

$$\dot{\zeta} = A\zeta + Bu \tag{6.10}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -1 \end{bmatrix}. \tag{6.11}$$

The state model of (6.10) will eventually result in a closed form guidance law as will be shown next. The closed form, however, would not be possible without making the necessary linearizations and corresponding assumptions discussed here. The process of finding the optimizing control u for a chosen performance index subject to the linearized motion constraints of (6.10), will proceed next.

Optimization

Referring back to the standard performance index of (6.5) as

$$J = \frac{1}{2}\bar{\zeta}\left(t_f\right) Q\zeta\left(t_f\right) + \frac{1}{2}\int_{t_0}^{t_f} \bar{u}Rudt. \tag{6.12}$$

we will specify the terminal state as the miss distance which we desire to be zero at the end of flight (target strike). The next step involves adjoining the motion constraints to the performance index as

$$J = \frac{1}{2}\bar{\zeta}\left(t_f\right) Q\zeta\left(t_f\right) + \frac{1}{2}\int_{t_0}^{t_f} \left[\bar{u}Ru + \lambda\left(A\zeta + Bu - \dot{\zeta}\right)\right] dt, \tag{6.13}$$

where λ is a vector of costates. The Hamiltonian, which is defined next, is very important in deriving the optimal control. The Hamiltonian is given as

$$H = \frac{1}{2}u^2 + \lambda\left(A\zeta + Bu\right). \tag{6.14}$$

Specific conditions for optimality are then

$$\dot{\lambda}\left(t\right) = -\frac{\partial H}{\partial \zeta}, \tag{6.15}$$

$$\frac{\partial H}{\partial u} = 0, \tag{6.16}$$

subject to the costate boundary conditions of

$$\lambda\left(t_f\right) = \frac{\partial}{\partial \zeta}\left(\frac{1}{2}\bar{\zeta}\left(t_f\right)Q\zeta\left(t_f\right)\right). \tag{6.17}$$

With the weighting matrices defined as

$$Q = \begin{bmatrix} b & 0 \\ 0 & c \end{bmatrix} \qquad R = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}, \tag{6.18}$$

the terminal state constraints are then expressed as

$$\frac{1}{2}\bar{\zeta}\left(t_f\right)Q\zeta\left(t_f\right) = \begin{bmatrix} p_r & v_r \end{bmatrix}\begin{bmatrix} b & 0 \\ 0 & c \end{bmatrix}\begin{bmatrix} p_r \\ v_r \end{bmatrix}. \tag{6.19}$$

The terminal state formulation of equation (6.19) leaves open two design options. For intercept problems, $b > 0$, $c = 0$, and for rendezvous problems $b > 0$, $c > 0$.

Equations (6.15) and (6.16) results in, respectively

$$\dot{\lambda} = -A\lambda, \qquad u = R^{-1}\lambda B \tag{6.20}$$

and therefore by adjoining the motion constraints to the performance index we gain a second dynamic system of costates in which the optimal control is a function of. Substitution of the optimal control u into the state model (12.4) gives

$$\dot{\zeta} = A\zeta + BR^{-1}\lambda B. \tag{6.21}$$

The complete state/costate dynamic system is then expressed as

$$\begin{bmatrix} \dot{\zeta} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B \\ 0 & -A \end{bmatrix} \begin{bmatrix} \zeta \\ \lambda \end{bmatrix} \tag{6.22}$$

where $A \in \Re^{2x2}$ and $BR^{-1}B \in \Re^{2x2}$. Our job now is to solve the state system of (6.22) in order to find solutions for the state and costate trajectories of ζ and λ. Once a closed form for trajectories are found, the optimal control given in (6.20) can then be expressed solely as a function of the state variables.

The system of (6.22) results in a two-point boundary value problem where the state derivative must be integrated from initial time to the terminal time subject to initial state boundary conditions, and the costate derivative must be backwards integrated from the terminal time to the initial time subject to the terminal boundary conditions given by (6.17). A couple of solution methods exist for solving these types of linear two-point boundary value problems and can be found in the open literature. However, the requirement of a linearized system should be clear, and is exactly the reason no closed form solution can be found for more complex and/or nonlinear systems. The integration

of (6.22) is complicated by the fact that the state and costate are cross coupled and only a linear system of this form will have a solution.

We can proceed in finding a solution, if we denote the state matrix of (6.22) as

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B} \\ 0 & -\bar{\mathbf{A}} \end{bmatrix}. \tag{6.23}$$

From linear system theory, we can find a solution for the state vector by computing the matrix exponential

$$\Phi = e^{\mathbf{F}(t-t_0)} = \mathbf{I} + (t-t_0)\,\mathbf{F} + \frac{(t-t_0)^2}{2!}\mathbf{F}^2 + \frac{(t-t_0)^3}{3!}\mathbf{F}^3. \tag{6.24}$$

This results in a state solution from initial time to the current time as

$$\begin{bmatrix} \zeta(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(t-t_0) & \Phi_{12}(t-t_0) \\ \Phi_{21}(t-t_0) & \Phi_{22}(t-t_0) \end{bmatrix} \begin{bmatrix} \zeta_0 \\ \lambda_0 \end{bmatrix} \tag{6.25}$$

where the transition matrix is calculated to be

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & (t-t_0) & \frac{(t-t_0)^3}{6r} & -\frac{(t-t_0)}{2r} \\ 0 & 1 & \frac{(t-t_0)^2}{2r} & -\frac{(t-t_0)}{r} \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & -(t-t_0) & 1 \end{array}\right]. \tag{6.26}$$

Unfortunately, this form is not suitable for guidance since it expresses the elements of the transition matrix as the results of an integration from initial time to current time, which is not useful. These equations must be integrated from the current time to final time. If we consider the time-to-go as

$$t_{go} = t_f - (t+t_0), \tag{6.27}$$

where $t_0 = 0$, and write equation (6.24) in terms of the desired time interval, we have

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & (t_f - t) & \frac{(t_f-t)^3}{6r} & -\frac{(t_f-t)}{2r} \\ 0 & 1 & \frac{(t_f-t)^2}{2r} & -\frac{(t_f-t)}{r} \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & -(t_f - t) & 1 \end{array}\right] = \left[\begin{array}{cc|cc} 1 & t_{go} & \frac{t_{go}^3}{6r} & -\frac{t_{go}}{2r} \\ 0 & 1 & \frac{t_{go}^2}{2r} & -\frac{t_{go}}{r} \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & -t_{go} & 1 \end{array}\right], \tag{6.28}$$

and therefore

$$\begin{bmatrix} \zeta(t_f) \\ \lambda(t_f) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(t_{go}) & \Phi_{12}(t_{go}) \\ \Phi_{21}(t_{go}) & \Phi_{22}(t_{go}) \end{bmatrix} \begin{bmatrix} \zeta(t) \\ \lambda(t) \end{bmatrix}. \tag{6.29}$$

The optimal guidance law is now explicit in time-to-go, as is often the case with many guidance laws. Pre-multiplying the first equation of (6.29) by Q gives

$$Q\zeta(t_f) = Q\Phi_{11}\zeta(t) + Q\Phi_{12}\lambda(t). \tag{6.30}$$

The second equation of (6.29) gives

$$\lambda(t_f) = \Phi_{22}\lambda(t). \tag{6.31}$$

From the costate boundary condition of (6.17), we know that

$$\lambda(t_f) = Q\zeta(t_f) \tag{6.32}$$

and therefore we can equate (6.30) and (6.31) as

$$\Phi_{22}\lambda(t) = Q\Phi_{11}\zeta(t) + Q\Phi_{12}\lambda(t). \tag{6.33}$$

which results in a costate solution in terms of the state as

$$\lambda(t) = \left[\Phi_{22} - Q\Phi_{12}\right]^{-1} Q\Phi_{11}\zeta(t). \tag{6.34}$$

Equation (6.34) now permits us to replace the costate variable in the optimal control of (6.20) as

$$u = -R^{-1}B\left[\Phi_{22} - Q\Phi_{12}\right]^{-1} Q\Phi_{11}\zeta(t). \tag{6.35}$$

Equation (6.35) in its current form is not very useful and requires some further simplifications. We can simplify the terms inside of the inversion as

$$\Phi_{22} - Q\Phi_{12} = \begin{bmatrix} 1 & 0 \\ -t_{go} & 0 \end{bmatrix} - \begin{bmatrix} b & 0 \\ 0 & c \end{bmatrix} \begin{bmatrix} \frac{t_{go}^3}{6r} & -\frac{t_{go}^2}{2r} \\ \frac{t_{go}^2}{2r} & -\frac{t_{go}}{r} \end{bmatrix} = \begin{bmatrix} \left(1 - \frac{bt_{go}^3}{6r}\right) & \frac{bt_{go}^2}{2r} \\ -\left(t_{go} + \frac{ct_{go}^2}{2r}\right) & \left(1 + \frac{ct_{go}}{r}\right) \end{bmatrix}, \tag{6.36}$$

and then algebraically compute the inversion

$$\left(\Phi_{22} - Q\Phi_{12}\right)^{-1} = \frac{1}{det} \begin{bmatrix} \left(1 + \frac{ct_{go}}{2}\right) & -\frac{bt_{go}^2}{2r} \\ \left(t_{go} + \frac{ct_{go}^2}{2r}\right) & \left(1 - \frac{bt_{go}^3}{6r}\right) \end{bmatrix}, \tag{6.37}$$

where

$$det = \frac{1}{\left[\left(1 - \frac{bt_{go}^3}{6r}\right)\left(1 + \frac{ct_{go}}{r}\right) + \left(t_{go} + \frac{ct_{go}^2}{2r}\right)\frac{bt_{go}^2}{2r}\right]}. \tag{6.38}$$

Next, combining (6.37) with (6.35) gives

$$u = -R^{-1}B\frac{1}{det} \begin{bmatrix} \left(1 + \frac{ct_{go}}{2}\right) & -\frac{bt_{go}^2}{2r} \\ \left(t_{go} + \frac{ct_{go}^2}{2r}\right) & \left(1 - \frac{bt_{go}^3}{6r}\right) \end{bmatrix} Q \begin{bmatrix} 1 & t_{go} \\ 0 & 1 \end{bmatrix} \zeta(t), \tag{6.39}$$

with the guidance law expressed in terms of the state variables as

$$u = -R^{-1}B\frac{1}{det}\begin{bmatrix} \left(1+\frac{ct_{go}}{2}\right) & -\frac{bt_{go}^2}{2r} \\ \left(t_{go}+\frac{ct_{go}^2}{2r}\right) & \left(1-\frac{bt_{go}^3}{6r}\right) \end{bmatrix} Q \begin{bmatrix} 1 & t_{go} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_r \\ v_r \end{bmatrix}. \tag{6.40}$$

Equation (6.40) can be simplified to reveal the cartesian form of the Proportional Navigation guidance law. If we set the weights of $b = 1$ and $c = 0$ for an intercept and consider $r = 0$ for no weight on control effort, we can simplify the matrix and determinate expressions of

$$u = -\frac{1}{r}\left(\frac{1}{1+\frac{t_{go}^3}{3r}}\right)\begin{bmatrix} 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & \frac{bt_{go}^2}{2r} \\ t_{go} & 1-\frac{bt_{go}^3}{6r} \end{bmatrix}\begin{bmatrix} b & 0 \\ 0 & c \end{bmatrix}\begin{bmatrix} p_r + v_r t_{go} \\ v_r \end{bmatrix}, \tag{6.41}$$

and finally arrive at the cartesian form of PN as

$$u = -\frac{3}{t_{go}^2}\left(p_r + v_r t_{go}\right). \tag{6.42}$$

This is an expected result since the well known Proportional Navigation Guidance law is an optimal guidance law under the initial assumptions used to begin this derivation. It can be shown that the optimal performance of this law, and others of its type, degrades considerably when the given assumptions do not hold true in real life, such as the target accelerating or evasively maneuvering. In addition, since (6.42) is explicit in time-to-go, the optimality of the guidance law depends to some degree on the fidelity of the time-to-go calculation. As will be discussed in the next chapter, the calculation of time-to-go can contain potential error.

Time-to-go Calculation for Homing Phase Target Intercept

Two standard methods for computing the time-to-go exist in the open literature. The first is the range-over-range rate given by

$$T_{go} = \frac{r}{\dot{r}}, \tag{6.43}$$

and the second method is the range-over-missile-velocity given as

$$T_{go} = \frac{r}{V}. \tag{6.44}$$

Each of the computation methods in (6.43) and (6.44) are based on the line of sight between the target and missile. In real-time applications, these equations are calculated for a snap-shot in time where the velocity and line-of-sight is considered to be constant. The equations are then continuously updated at regular intervals throughout the flight engagement to provide a current estimate of the time-to-go.

The fact that both calculations are based on the line-of-sight raises the concern of potential error in the time-to-go calculation, and in fact, this concern is valid. If the missile is on a straight-line collision course with the target, then each of the given time-to-go calculations will provide an exact value. However, as the trajectory deviates from the ideal straight-line scenario, each of the calculation methods begin to produce more error simply because of the trajectory curvature. This induced error in the guidance law, as well as other error sources will be discussed next.

Potential Errors in the Optimal Guidance Law

The basic guidance law of (6.42) is optimal, as long as the original assumptions hold in real-time applications. For instance, an assumption of zero missile and target acceleration was made. If this

condition holds true in reality, then (6.42) will perform well and the miss distance will be zero or very close to zero under these conditions. The guidance law is even quite tolerable to some degree of error in the time-to-go calculation.

The primary issue is, many missile/target engagements do not accommodate such ideal assumptions. If acceleration of the missile is present due to atmospheric drag, and the target is evasively maneuvering as well, the optimal solution begins to degrade and the miss-distance increases, in some cases, to the point of missing the target completely. Certainly if there is considerable maneuvering, the additional trajectory curvature adds error to the time-to-go calculation further effecting performance of the guidance law. The reasons for such a degrade in performance should be clear; in order to gain the closed form solution of (6.42), the velocity had to be considered constant and the engagement kinematics linearized. This constitutes a considerable loss in fidelity but, it does result in a closed form solution.

Barring the potential error that results from simplifying assumptions and the linearization process, it will be shown in chapter 14 that even under constant velocity and stationary target engagement conditions, trajectory curvature alone can produce several seconds of error in the time-to-go calculation which in turn has an effect on the optimality of the guidance law. All of these potential errors should be taken into consideration based on the mission at hand. For instance, in conducting a tactical surface-to-air missile strike on an aircraft flying in a straight line and at a constant altitude and speed, the guidance law of (6.42) with the corresponding choice for the time-to-go equations of either (6.43) or (6.44), would perform well. On the other hand, this choice would be inadequate for striking an evasively maneuvering target or in a case where the missile itself undergoes a rapid change in velocity with atmospheric disturbances present.

The wide range of specific applications is simply too large to cover in this work, however, improvements can be made to existing methods and applied to a difficult missile guidance problem

in order demonstrate the proposed design and intent of this work. In the chapter that follows, the specific guidance problem of interest will be defined as well as the configuration space, equations of motion, and the trajectory sets to be designed.

CHAPTER 7: PROBLEM FORMULATION

The scope and intent of this work will now become clear; to improve upon existing methods both in guidance law formulation as well as the time-to-go calculation in order to more effectively control the impact time of a missile engaging a target in the presence of and atmosphere, gravity and disturbance. Very little has been discussed concerning the control of impact time since a cursory survey of the basic guidance problem was necessary first. Existing guidance laws allow little freedom in design because of the need for a closed form solution, and the methods for solving the two-point boundary value problem required to archive the desired solution leaves little room for variations in the design. While the literature survey presented several published research papers on impact time control guidance laws, the work is restricted by the need for a closed form solution and thus many simplifying assumptions are made which do not account for potential error in the calculations such as time-to-go nor do they account for disturbance and nonlinearity.

Defining the Problem

The problem of achieving a prescribed impact time for a guided munition on a stationary target located at some terminal position in the downrange/crossrange plane is considered. The trajectory is analytically designed in closed form, and unlike existing methods, the time-to-go is analytically computed from the trajectory arc-length. Although projectile velocity changes due to gravity, drag, and atmosphere, the time-to-go can be estimated online based upon the current velocity and the remainder of the trajectory length. For this work, initial conditions for generation of the guidance solution is considered to be trajectory apogee after boost phase or the deployment point from the bay-door of an aircraft. This particular guidance problem was chosen for its nonlinear complexity regarding the rapid change in velocity and atmospheric disturbances and an effective

solution for this problem can possibly have a wide ranging series of applications not only for missile applications but unmanned ariel aircraft avoidance applications as well.

Configuration Space and Equations of Motion

The distances considered in this work are less than 100 nautical miles and therefore the earths curvature and ellipticity are not considered. The 3 DOF configuration space is shown in figure 7.1 and the kinematic equations of motion used for guidance law synthesis is given by

$$\dot{V} = -a_d - g\sin(\gamma) \tag{7.1}$$

$$\dot{\chi} = \frac{1}{V\cos(\gamma)}u_\chi \tag{7.2}$$

$$\dot{\gamma} = \frac{1}{V}(u_\gamma - g\cos(\gamma)) \tag{7.3}$$

$$\dot{x} = V\cos(\gamma)\cos(\chi) \tag{7.4}$$

$$\dot{y} = V\cos(\gamma)\sin(\chi) \tag{7.5}$$

$$\dot{z} = V\sin(\gamma), \tag{7.6}$$

where the set of initial$_{(0)}$ and terminal$_{(T)}$ conditions is given by

$$\begin{cases} (x_0, y_0), \ (x_0, z_0), \ (\chi_0, \gamma_0) \\ (x_T, y_T), \ (x_T, \ z_T), \ (\chi_T, \gamma_T). \end{cases} \tag{7.7}$$

Velocity vector V is contained within the configuration space consisting of downrange x, and crossrange y, altitude z. Divert controls u_χ and u_γ in (2) and (3) are normal to V, acceleration due to drag a_d is in the negative direction of V, χ and γ are the heading and flight path angles, respectively. The gravity value used in this work is $9.81\frac{m}{s^2}$. Divert controls have no effect on

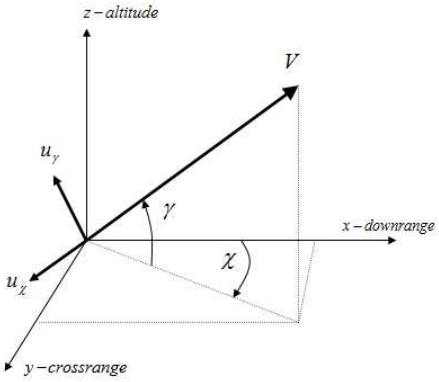

Figure 7.1: Configuration Space

the magnitude of V since they are each normal to V. Only drag and gravity has an effect on V. Equation (7.1) contains unknown drag a_d to be compensated for by trajectory planning and the corresponding impact time control. Note that from thus point forward, the subscript I for x, y, and z is dropped for neatness of the equations. The family of trajectories are prescribed next.

Trajectory Parameterization

In the proposed development, segments of crossrange and altitude trajectories are chosen analytically within the family of second order polynomials of the form

$$\begin{aligned} y_n(x) &= a_n^y + c_n^y x + \kappa_n^y x^2 \\ z_n(x) &= a_n^z + c_n^z x + \kappa_n^z x^2 \end{aligned}, \tag{7.8}$$

where n denotes the index of segments. Parameterization of the trajectories should be chosen to satisfy dynamic equations (4) to (6), or equivalently

$$\frac{\partial y}{\partial x} = \tan(\chi), \quad \frac{\partial z}{\partial x} = \frac{\tan(\gamma)}{\cos(\chi)}, \tag{7.9}$$

which are obtained by dividing (4) into (5) and (6), respectively. In fact, the equations in (7.9) are satisfied for all time if the trajectories in (7.8) satisfy (7.9) at the initial and terminal conditions given by (7.7). Therefore, the six boundary conditions provided by (7.7) map into the path planning boundary conditions required by (7.8) according to

$$\begin{cases} (x_0, y_0)\,,\ \frac{\partial}{\partial x} y_1(x_0) = \tan(\chi_0) \\ (x_T, y_T)\,,\ \frac{\partial}{\partial x} y_4(x_T) = \tan(\chi_T) \end{cases} \tag{7.10}$$

and

$$\begin{cases} (x_0, z_0)\,,\ \frac{\partial}{\partial x} z_1(x_0) = \frac{\tan(\gamma_0)}{\cos(\chi_0)} \\ (x_T, z_T)\,,\ \frac{\partial}{\partial x} z_4(x_T) = \frac{\tan(\gamma_T)}{\cos(\chi_T)}. \end{cases} \tag{7.11}$$

In the next chapter we introduce the proposed guidance law.

CHAPTER 8: QUAD-SEGMENT-POLYNOMIAL-TRAJECTORY GUIDANCE

In chapter 6, potential error sources involving the standard guidance methods were outlined. Specifically, the problems associated with the calculation of time-to-go and the possible improvements that would be of benefit if they were possible. It was discussed that the line-of-sight based time-to-go does not account for curvature of the trajectory and thus error in the calculation can result. When the control of impact time becomes necessary, however, a less error prone equation other than the choices of (6.43) and (6.44) will be required. Other enhancements can be made to improve the performance as well such as alleviating the need to linearize the equations of motion. The development of the QSPT Guidance law aims to provide the following design benefits of

- an arc-length based time-to-go calculation
- no linearization of the kinematic equations of motion required
- guidance law non-explicit in time-to-go
- family of trajectories available

The benefits of the first three items can be readily seen from the previous chapter discussions. The availability of a family of trajectories is of great benefit to the trajectory planning required in order to deal with the under-actuation problem of missiles. It will form the basis for the control of the impact time. In the next section we define the mathematical structure of QSPT Guidance.

Design of QSPT

A unique solution to the arc-length based time-to-go problem is sought which provides a closed form solution for the arc-length of the trajectory set given in (7.8). The solution proposed here and shown in figure 8.1 is to use multiple second order trajectories, in this case four, and connect them all in a smooth and continuous manner to form a single trajectory. Under these conditions, the trajectory would be capable of achieving multi-directional curvature similar to higher order polynomials while providing a closed form solution for arc-length. Much design work is required first and thus the formulation of the QSPT time-to-go will be presented in chapter 11 once all of the necessary groundwork is completed.

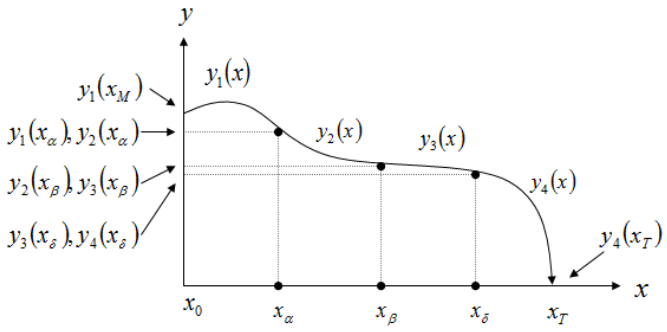

Figure 8.1: Crossrange profile - segmentation of downrange

Figure 8.1 illustrates the basic structure of a QSPT trajectory. The downrange axis between the initial missile location and the target location is segmented into four equal length segments as

$$\begin{aligned} x_\alpha &= x_0 + \frac{1}{4}(x_T - x_0) \\ x_\beta &= x_0 + \frac{1}{2}(x_T - x_0) \\ x_\delta &= x_0 + \frac{3}{4}(x_T - x_0)\,. \end{aligned} \tag{8.1}$$

These downrange locations serve as points to enforce boundary conditions. For any QSPT trajectory, there exists two initial and two terminal boundary conditions. Additional boundary conditions are found at locations x_α, x_β, and x_δ which serve to connect the segments together and are referred to as internal boundary conditions.

Choice of Four Segments

A second order polynomial has available three coefficients and can therefore only satisfy three boundary conditions. A typical trajectory, if a planar one is considered, must satisfy at least two initial and two terminal boundary conditions. That is, initial position and orientation as well as terminal position and orientation. The design of QSPT is required to not only satisfy these boundary conditions, but it must also have a free design coefficient available. The choice of four quadratic segments will yield a total of twelve available coefficients; two for the initial conditions, two for the terminal conditions, and seven to satisfy the internal boundary conditions. This leaves one remaining coefficient to use for trajectory planning purposes.

Referring to figure 8.1, two initial conditions of position and orientation are imposed at the downrange location of x_0, and two terminal conditions of position and orientation are imposed at location x_T. The first of the internal conditions enforcing position and orientation is imposed at location x_α. Position and orientation is imposed at location x_δ as well. The central tie point at location x_β enforces not only position and orientation, but curvature also. The internal boundary locations of x_α and x_δ are designated to be points of inflection for the trajectory and is the reason curvature is not enforced at those locations. The tie point at x_β is designed to join the two halves together in a smooth and continuous manner and is the reason for enforcing the additional constraint of curvature at that location. In the next section, we proceed with the development of the coefficient equations for the trajectory set of $y_n(x)$ in (7.8), keeping in mind the same equations

apply for the coefficients of $z_n(x)$.

QSPT Coefficients

Step 1, Solve the coefficients of segment 1 with respect to initial boundary conditions. Therefore, solve

$$y_1(x) = a_1^y + c_1^y x + \kappa_1^y x^2 \tag{8.2}$$

for $a_1^y,\ c_1^y$ subject to the initial boundary conditions of (7.10). This results in,

$$\begin{aligned} a_1^y &= y_1(x_0) - c_1^y x_0 - \kappa_1^y x_0^2 \\ c_1^y &= \frac{\partial}{\partial x} y_1(x_0) - 2\kappa_1^y x_0. \end{aligned} \tag{8.3}$$

Step 2, join segment 1 to segment 2 by enforcing the internal boundary conditions of (**??**). This requires solving the equations

$$\begin{aligned} y_1(x_\alpha) &= y_2(x_\alpha) \\ \frac{\partial}{\partial x} y_1(x_\alpha) &= \frac{\partial}{\partial x} y_2(x_\alpha) \end{aligned} \tag{8.4}$$

for a_2^y and c_2^y which results in

$$\begin{aligned} a_2^y &= a_1^y + c_1^y x_\alpha + \kappa_1^y x_\alpha^2 - c_2^y x_\alpha - \kappa_2^y x_\alpha^2 \\ c_2^y &= c_1^y + 2\kappa_1^y x_\alpha - 2\kappa_2^y x_\alpha. \end{aligned} \tag{8.5}$$

Step 3, similar to step 1, solve segment 4 subject to the terminal boundary conditions of (7.10). This results in

$$\begin{aligned} a_4^y &= y_T - c_4^y x_T - \kappa_4^y x_T^2 \\ c_4^y &= \frac{\partial}{\partial x} y_4\left(x_T\right) - 2\kappa_4^y x_T. \end{aligned} \tag{8.6}$$

Step 4, join segment 4 to segment 3 by enforcing the internal boundary conditions of (**??**). This requires solving the boundary equations

$$\begin{aligned} y_3\left(x_\delta\right) &= y_4\left(x_\delta\right) \\ \frac{\partial}{\partial x} y_3\left(x_\delta\right) &= \frac{\partial}{\partial x} y_4\left(x_\delta\right) \end{aligned} \tag{8.7}$$

for a_3^y and c_3^y which results in

$$\begin{aligned} a_3^y &= a_4^y + c_4^y x_\delta + \kappa_4^y x_\delta^2 - c_3^y x_\delta - \kappa_3^y x_\delta^2 \\ c_3^y &= c_4^y + 2\kappa_4^y x_\delta - 2\kappa_3^y x_\delta \end{aligned} \tag{8.8}$$

Step 5, join segment 3 to segment 2 by enforcing the boundary conditions in (**??**). This requires solving the equations

$$\begin{aligned} y_2\left(x_\beta\right) &= y_3\left(x_\beta\right) \\ \frac{\partial}{\partial x} y_2\left(x_\beta\right) &= \frac{\partial}{\partial x} y_3\left(x_\beta\right) \\ \frac{\partial^2}{\partial x^2} y_2\left(x_\beta\right) &= \frac{\partial^2}{\partial x^2} y_3\left(x_\beta\right) \end{aligned} \tag{8.9}$$

which results in a vector matrix expression. Simultaneously solving these equations for κ_1^y, κ_2^y, κ_3^y gives

$$\begin{bmatrix} \kappa_1^y \\ \kappa_2^y \\ \kappa_3^y \end{bmatrix} = \begin{bmatrix} C_{11}^y & C_{12}^y & C_{13}^y \\ C_{21}^y & C_{22}^y & C_{23}^y \\ C_{31}^y & C_{32}^y & C_{33}^y \end{bmatrix}^{-1} \begin{bmatrix} D_{11}^y \\ D_{21}^y \\ D_{31}^y \end{bmatrix} = C^{-1}D \tag{8.10}$$

where

$$\begin{aligned} C_{11}^y &= -x_0^2 + x_\alpha^2 + 2x_0x_\beta - 2x_\alpha x_\beta \\ C_{12}^y &= -x_\beta^2 + 2x_\alpha x_\beta - x_\beta^2 \\ C_{13}^y &= x_\delta^2 - 2x_\beta x_\delta + x_\beta^2 \\ C_{21}^y &= 2\left(x_0 - x_\alpha\right) \\ C_{22}^y &= 2\left(x_\alpha - x_\beta\right) \\ C_{23}^y &= 2\left(x_\beta - x_\delta\right) \\ C_{31}^y &= 0 \\ C_{32}^y &= -2 \\ C_{33}^y &= 2 \end{aligned} \tag{8.11}$$

and

$$\begin{aligned} D_{11}^y &= y_1\left(x_0\right) + \frac{\partial}{\partial x} y_1\left(x_0\right)\left(x_\beta - x_0\right) - y_4\left(x_T\right) + \\ &\quad + \frac{\partial}{\partial x} y_4\left(x_T\right)\left(x_T - x_\beta\right) - \\ &\quad - \kappa_4^y\left(x_T^2 - x_\delta^2 - 2x_T x_\beta + 2x_\beta x_\delta\right) \\ D_{21}^y &= \frac{\partial}{\partial x} y_1\left(x_0\right) - \frac{\partial}{\partial x} y_4\left(x_T\right) - 2\kappa_4^y\left(x_\delta - x_T\right) \\ D_{31}^y &= 0. \end{aligned} \tag{8.12}$$

Applying these coefficient equations to the design of the altitude trajectory z_n simply requires replacing $y_1(x_0),\ y_T(x_T)$ with $z_1(x_0),\ z_4(x_T)$ respectively. Therefore, the full set of coefficient equations are given as

$$\begin{aligned}
a_1^z &= z_1(x_0) - c_1^z x_0 - \kappa_1^z x_0^2 \\
c_1^z &= \frac{\partial}{\partial x} z_1(x_0) - 2\kappa_1^z x_0 \\
a_2^z &= a_1^z + c_1^z x_\alpha + \kappa_1^z x_\alpha^2 - c_2^z x_\alpha - \kappa_2^z x_\alpha^2 \\
c_2^z &= c_1^z + 2\kappa_1^z x_\alpha - 2\kappa_2^z x_\alpha \\
a_3^z &= a_4^z + c_4^z x_\delta + \kappa_4^z x_\delta^2 - c_3^z x_\delta - \kappa_3^z x_\delta^2 \\
c_3^z &= c_4^z + 2\kappa_4^z x_\delta - 2\kappa_3^z x_\delta \\
a_4^z &= z_T - c_4^z x_T - \kappa_4^z x_T^2 \\
c_4^z &= \frac{\partial}{\partial x} z_4(x_T) - 2\kappa_4^z x_T,
\end{aligned} \tag{8.13}$$

$$\begin{bmatrix} \kappa_1^z \\ \kappa_2^z \\ \kappa_3^z \end{bmatrix} = \begin{bmatrix} C_{11}^z & C_{12}^z & C_{13}^z \\ C_{21}^z & C_{22}^z & C_{23}^z \\ C_{31}^z & C_{32}^z & C_{33}^z \end{bmatrix}^{-1} \begin{bmatrix} D_{11}^z \\ D_{21}^z \\ D_{31}^z \end{bmatrix} = C^{-1}D, \tag{8.14}$$

$$
\begin{aligned}
C_{11}^z &= -x_0^2 + x_\alpha^2 + 2x_0x_\beta - 2x_\alpha x_\beta \\
C_{12}^z &= -x_\beta^2 + 2x_\alpha x_\beta - x_\beta^2 \\
C_{13}^z &= x_\delta^2 - 2x_\beta x_\delta + x_\beta^2 \\
C_{21}^z &= 2\left(x_0 - x_\alpha\right) \\
C_{22}^z &= 2\left(x_\alpha - x_\beta\right) \\
C_{23}^z &= 2\left(x_\beta - x_\delta\right) \\
C_{31}^z &= 0 \\
C_{32}^z &= -2 \\
C_{33}^z &= 2,
\end{aligned}
\tag{8.15}
$$

and

$$
\begin{aligned}
D_{11}^z &= z_1\left(x_0\right) + \frac{\partial}{\partial x} z_1\left(x_0\right)\left(x_\beta - x_0\right) - z_4\left(x_T\right) + \\
&\quad + \frac{\partial}{\partial x} z_4\left(x_T\right)\left(x_T - x_\beta\right) - \\
&\quad - \kappa_4^z\left(x_T^2 - x_\delta^2 - 2x_Tx_\beta + 2x_\beta x_\delta\right) \\
D_{21}^z &= \frac{\partial}{\partial x} z_1\left(x_0\right) - \frac{\partial}{\partial x} z_4\left(x_T\right) - 2\kappa_4^z\left(x_\delta - x_T\right) \\
D_{31}^z &= 0.
\end{aligned}
\tag{8.16}
$$

Next, we look at the roles the coefficient equations play in the control of impact time.

Coefficient Equations and Impact Time Control

In this section, the functionality of the coefficient sets must be defined, as each set takes on a different role when trajectory planning is required for impact time control. The functionality is determined specifically by the choice of the free variables κ_4^y and κ_4^z. For a constant trajectory in

space, free variables κ_4^y and κ_4^z are initialized to the optimizing values designed in chapter 10 and remain constant throughout the flight as does all of the associated coefficients. For impact time control on the other hand, the following guidelines are now defined

- Crossrange is trajectory planned for impact time adjustment:
  - κ_4^y is a time varying function initialized to the optimizing value
  - equation (8.12) becomes time varying as well
  - equation (8.11) is constant if target is stationary
- Altitude trajectory remains constant:
  - κ_4^z is initialized to optimizing value and remains throughout flight
  - all z-related coefficients are constant if target is stationary

It will be shown in the next section that coefficient time-rates-of-change $\dot{a}^y_{1,2,3,4},\ \dot{c}^y_{1,2,3,4},\ \dot{\kappa}^y_{1,2,3}$, are driven strictly by the time-rate-of-change of κ_4^y.

Free Variable, Time Varying Coefficients

In this work, free variable κ_4^y is prescribed as the state solution to the simple first order system

$$\dot{\kappa}_4^y = \mu, \tag{8.17}$$

for which control μ is designed in section VII to impose changes in crossrange trajectory arc-length in order to satisfy impact time constraints. Therefore, the corresponding impact time control is designed for the crossrange trajectory only, since perturbing the altitude trajectory could have unforeseen consequences in terms of range extension. Because of this, free variable κ_4^z is held constant

at the optimizing value and therefore, coefficient derivatives $\dot{a}_n^z,\ \dot{c}_n^z,\ \dot{\kappa}_n^z$ equal zero. Furthermore, since (8.17) is of first order, second derivatives $\ddot{a}_n^y,\quad \ddot{c}_n^y$ vanish, hence, explaining the differences between $\ddot{y}_n$ and $\ddot{z}_n$ in the closed loop equations of chapter 9.

To find expressions for the coefficient derivatives, we begin with (8.10) and group the D matrix according to terms involving κ_4^y. From direct inspection of (8.12) we find

$$\begin{bmatrix} D_{11} \\ D_{21} \\ D_{31} \end{bmatrix} = \begin{bmatrix} f_{11} + g_{11}\kappa_4^y \\ f_{22} + g_{22}\kappa_4^y \\ 0 \end{bmatrix} \tag{8.18}$$

where

$$\begin{aligned} f_{11} &= y\left(x_0\right) - y_4\left(x_T\right) + \frac{\partial}{\partial x} y_1\left(x_0\right)\left(x_\beta - x_0\right) + \\ &\quad + \frac{\partial}{\partial x} y_4\left(x_T\right)\left(x_T - x_\beta\right) \\ g_{11} &= -x_T^2 + x_\delta^2 + 2x_T x_\beta - 2x_\beta x_\delta \\ f_{22} &= \frac{\partial}{\partial x} y_1\left(x_0\right) - \frac{\partial}{\partial x} y_4\left(x_T\right) \\ g_{22} &= 2\left(x_T - x_\delta\right) \end{aligned} \tag{8.19}$$

which then results in

$$\begin{bmatrix} \kappa_1^y \\ \kappa_2^y \\ \kappa_3^y \end{bmatrix} = C^{-1}\left[\begin{bmatrix} f_{11} \\ f_{22} \\ 0 \end{bmatrix} + \begin{bmatrix} g_{11} \\ g_{22} \\ 0 \end{bmatrix} \kappa_4^y\right]. \tag{8.20}$$

Matrix C^{-1} as well as the equations in (8.19) are constant. Given (8.17), the derivative of (8.20) is

therefore

$$\begin{bmatrix} \dot{\kappa}_1^y \\ \dot{\kappa}_2^y \\ \dot{\kappa}_3^y \end{bmatrix} = C^{-1} \begin{bmatrix} g_{11} \\ g_{22} \\ 0 \end{bmatrix} \mu. \tag{8.21}$$

Then, finding the coefficient derivatives of (8.3), (8.5), (8.6), and (8.8) gives

$$\begin{aligned}
\dot{a}_1^y &= -\dot{c}_1^y x_0 - \dot{\kappa}_1^y x_0^2 \\
\dot{c}_1^y &= -2\dot{\kappa}_1^y x_0 \\
\dot{a}_2^y &= \dot{a}_1^y + \dot{c}_1^y x_1 + \dot{\kappa}_1^y x_\alpha^2 - \dot{c}_2^y x_0 - \dot{\kappa}_2^y x_\alpha^2 \\
\dot{c}_2^y &= \dot{c}_1^y + 2\dot{\kappa}_1^y x_1 - 2\dot{\kappa}_2^y x_1 \\
\dot{a}_3^y &= \dot{a}_4^y + \dot{c}_4^y x_\delta + x_\delta^2 \mu - \dot{c}_3^y x_\delta - \dot{\kappa}_3^y x_\delta^2 \\
\dot{c}_3^y &= \dot{c}_4^y + 2x_\delta \mu - 2\dot{\kappa}_3^y x_3 \\
\dot{a}_4^y &= -\dot{c}_4^y x_T - x_T^2 \mu \\
\dot{c}_4^y &= -2x_T \mu.
\end{aligned} \tag{8.22}$$

The coefficient derivatives of (8.21) and (8.22) are thus strictly driven by the prescribed first order system of (8.17).

If a trajectory is desired which requires changes to the impact time, as in the case of the crossrange, the coefficient derivative equations of (8.21) and (8.22) are implemented and initialized to values set by (8.3), (8.5), (8.6), and (8.8) with the proper optimizing value of κ_4^y chosen. If no changes to the impact time are required, as in the case of the altitude trajectory, the coefficient values are initialized by (8.3), (8.5), (8.6), and (8.8) with the optimizing value for κ_4^z chosen. These z-related coefficients remain constant throughout the engagement. The design of both open and closed loop the guidance laws is given next.

CHAPTER 9: GUIDANCE DESIGN AND IMPLEMENTATION

In this chapter, an open loop and closed loop guidance law is designed explicit in the trajectory set of (7.8) and given in terms of the control inputs of u_χ and u_γ.

Open Loop Guidance Law

The open loop guidance law is derived directly from the equations of (7.2)-(7.6). Dividing (7.2) and (7.3) by (7.4) gives the open loop guidance law as

$$u_\chi = \frac{\partial \chi}{\partial x} V^2 \cos^2\gamma \cos\chi, \quad u_\gamma = \frac{\partial \gamma}{\partial x} V^2 \cos\gamma\cos\chi + g\cos\gamma, \tag{9.1}$$

where according to (7.8) and (7.9), the angular profiles derived in terms of QSPT are given as

$$\begin{aligned} \chi &= \tan^{-1}\left(\frac{\partial}{\partial x} y_n\right) \\ \gamma &= \tan^{-1}\left(\frac{\frac{\partial}{\partial x} z_n}{\cos\left(\tan^{-1}\left(\frac{\partial}{\partial x} y_n\right)\right)}\right), \end{aligned} \tag{9.2}$$

where

$$\frac{\partial}{\partial x} y_n = c_n^y + 2\kappa_n^y x, \qquad \frac{\partial}{\partial x} z_n = c_n^z + 2\kappa_n^z x. \tag{9.3}$$

The bottom equation for γ in (9.2) can be simplified by realizing that

$$\cos\left(\tan^{-1}\left(\frac{\partial}{\partial x} y_n\right)\right) = \frac{1}{\sqrt{1 + \left(\frac{\partial}{\partial x} y_n\right)^2}}, \tag{9.4}$$

which gives

$$\gamma = \tan^{-1}\left(\frac{\frac{\partial}{\partial x}z_n}{\sqrt{1+\left(\frac{\partial}{\partial x}y_n\right)^2}}\right). \tag{9.5}$$

The partial derivatives required by (9.1) are given as

$$\begin{aligned}\frac{\partial\chi}{\partial x} &= \frac{2\kappa_n^y}{1+\left(c_n^y+2\kappa_n^y x\right)^2}\\ \frac{\partial\gamma}{\partial x} &= \frac{2\left(\kappa_n^z+c_n^y\kappa_n^z\left(\frac{\partial}{\partial x}y_n\right)-c_n^z\kappa_n^y\left(\frac{\partial}{\partial x}y_n\right)\right)}{\sqrt{1+\left(\frac{\partial}{\partial x}y_n\right)^2}\left(1+\left(\frac{\partial}{\partial x}y_n\right)^2+\left(\frac{\partial}{\partial x}z_n\right)^2\right)}.\end{aligned} \tag{9.6}$$

which completes the open loop guidance law.

Closed Loop Guidance Law

Alternatively, we can implement the corresponding closed-loop guidance design. To this end, define

$$e_y = y - y_n, \quad e_z = z - z_n, \tag{9.7}$$

where y_n and z_n are given by (7.8) for the period of time when the n^{th} segment of the trajectories are being implemented. Second order time derivatives of (9.7) are explicit in the controls u_χ and u_γ through the equations

$$\begin{aligned}\ddot{y} &= -a_d\cos\gamma\sin\chi - \sin\gamma\sin\chi u_\gamma + \cos\chi u_\chi\\ \ddot{z} &= -a_d\sin\gamma - g + \cos\gamma u_\gamma\\ \ddot{y}_n &= 2\dot{c}_n^y\dot{x} + 2\kappa_n^y\left(\dot{x}\right)^2 + 4\dot{\kappa}_n^y x\dot{x} + c_n^y\ddot{x} + 2\kappa_n^y x\ddot{x}\\ \ddot{z}_n &= c_n^z\ddot{x} + 2\kappa_n^z\left(\dot{x}\right)^2 + 2\kappa_n^z x\ddot{x},\end{aligned} \tag{9.8}$$

where $\dot{x}$ is given from (4), and $\ddot{x}$ is given by

$$\ddot{x} = -a_d \cos\gamma \cos\chi - \sin\gamma \cos\chi u_\gamma - \sin\chi u_\chi. \tag{9.9}$$

It should be noted that the distinct differences between $\ddot{y}_n$ and $\ddot{z}_n$ in (9.8) relating to the coefficient time derivatives are due to the design choices imposed on the free variables κ_4^y and κ_n^z which was discussed in chapter 8.

It follows from the second derivative of (9.7), and the substitution of (9.9) into (9.8), that the error system is given by

$$\begin{bmatrix} \ddot{e}_y \\ \ddot{e}_z \end{bmatrix} = A + B \begin{bmatrix} u_\chi \\ u_\gamma \end{bmatrix}, \tag{9.10}$$

where

$$A = \begin{bmatrix} -a_d \cos\gamma \sin\chi - \left(2\dot{c}_n^y \dot{x} + 4\kappa_n^y x\dot{x} + 2\kappa_n^y (\dot{x})^2\right) \\ +\frac{\partial}{\partial x} y_n a_d \cos\gamma \cos\chi \\ -a_d \sin\gamma - g - 2\kappa_n^z (\dot{x})^2 + \frac{\partial}{\partial x} z_n a_d \cos\gamma \cos\chi \end{bmatrix}, \tag{9.11}$$

$$B = \begin{bmatrix} \cos\chi + \frac{\partial}{\partial x} y_n \sin\chi, & -\sin\gamma \sin\chi + \frac{\partial}{\partial x} y_n \sin\gamma \cos\chi \\ \frac{\partial}{\partial x} z_n \sin\chi, & \cos\gamma + \frac{\partial}{\partial x} z_n \sin\gamma \cos\chi \end{bmatrix}. \tag{9.12}$$

The controls of u_χ, u_γ can be written in terms of a pseudo control as

$$\begin{bmatrix} u_\chi \\ u_\gamma \end{bmatrix} = B^{-1} (\nu - A), \tag{9.13}$$

such that the error system of (9.10) results in the chain of integrators as

$$\begin{bmatrix} \ddot{e}_y \\ \ddot{e}_z \end{bmatrix} = \begin{bmatrix} \nu_y \\ \nu_z \end{bmatrix}. \tag{9.14}$$

A stabilizing control for (9.14) can then be designed as

$$\begin{aligned} \nu_y &= -k_y e_y - k'_y \dot{e}_y \\ \nu_z &= -k_z e_z - k'_z \dot{e}_z. \end{aligned} \tag{9.15}$$

Therefore, the closed loop guidance law is

$$\begin{bmatrix} u_\chi \\ u_\gamma \end{bmatrix} = B^{-1} \left\{ \begin{bmatrix} k_y(y_n - y) + k'_y(\dot{y}_n - \dot{y}) \\ k_z(z_n - z) + k'_z(\dot{z}_n - \dot{z}) \end{bmatrix} - A \right\}, \tag{9.16}$$

under which the actual trajectories converge asymptotically and exponentially to the guidance trajectories of y_n and z_n. Drag acceleration a_d is estimated in real time using the data from the onboard accelerometer.

Inversion Matrix Singularities

The inversion matrix of (9.12) becomes singular if the flight path angle reaches $\pm\frac{\pi}{2}$, however, the slope of second order polynomials cannot reach vertical angles as that would require the partial derivative $\frac{\partial}{\partial x} y_n$ to be undefined. Therefore, the inversion of (9.12) can never reach a singularity under normal operation. On the other hand, it is a requirement in some cases that a precision munition reach a vertical angle in order to drop in on a target from directly above. An easy solution to this problem is to reparameterize (7.8) in terms of a new independent variable, such as z, and switch the guidance law accordingly. This can be done at a point when the projectile is close to the target. Under the reparameterization, (7.8) can operate at vertical angles and the switched form guidance law is free of singularity at $\pm\frac{\pi}{2}$. In this case, we restrict the heading angle to $\chi < \pm\frac{\pi}{2}$, which is a reasonable constraint to impose. The derivation of the switched guidance law is outlined in appendix X and will also be demonstrated in the simulation.

Robustness of the Guidance Law

The QSPT guidance law given in (9.16) utilizes all of the known nonlinear terms of the equations of motion and incorporates then into the final form of the guidance law. The net effect is that the guidance law cancels-out the nonlinear terms and creates a linear chain of integrators which is easily stabilized. Therefore the original tracking problem has been converted to a stabilization problem. The nonlinearities of the guidance law are calculated online and used to dynamically create the linear system which is sometimes referred to as Dynamic Inversion or Input/Output Linearization. This is of course different from the process taken in chapter 6 where the nonlinear terms are linearized and then the guidance law is derived forward from that point. This is an excellent control synthesis approach as long as all of the nonlinearity has been modeled properly. In reality, there is always modeling error and certainly disturbances present. The question that must be asked is, how robust is the guidance law to error and disturbance? The quidance law is explicit in χ and γ which are sensed and delivered to the guidance law by the Inertial Measurement Unit (IMU), which has error and uncertainly associated with it. The performance of the guidance law under these conditions will be analyzed in chapter 14.

CHAPTER 10: QSPT TRAJECTORY OPTIMIZATION

Chapter 6 demonstrated the difficulty in finding a closed form solution for a guidance law when the kinematics involved are less than ideal. The process of linearization gave us the ideal linear system we needed in order to obtain a useful closed form result. Now, an optimization must be found for QSPT subject to the appropriate constraints. The optimization process taken in this chapter begins similarly to the one taken in chapter 6 but quickly diverges primarily because for QSPT, adjoining the motion constraints to the performance index is not necessary since a feasible trajectory already exists. QSPT can be shown to satisfy the equations of (7.9), and therefore, we know that the control u that minimizes J will also satisfy (7.9).

The terminal constraints are enforced by the QSPT boundary conditions and therefore terminal constraints associated with the performance index are not required. The corresponding optimization will be shown to boil down to a parameter optimization problem in which the QSPT free variable is found to produce a minimum control energy trajectory. We first assume constant velocity, which further simplifies the equations of motion as

$$\dot{x} = V\cos\gamma\cos\chi \tag{10.1}$$

$$\dot{y} = V\cos\gamma\sin\chi \tag{10.2}$$

$$\dot{z} = V\sin\gamma \tag{10.3}$$

$$\dot{\gamma} = \frac{1}{V}\left(u_\gamma - g\cos\gamma\right) \tag{10.4}$$

$$\dot{\chi} = \frac{1}{V\cos\gamma}u_\chi. \tag{10.5}$$

The objective is to design control inputs $u = [u_\chi, u_\gamma]$ such that a minimum amount of control energy is expended during flight. The equations of motion are highly nonlinear and no closed

form solution will exist for the two-point boundary value problem. If the equations of (10.5) are linearized, then the methods of chapter 6 are employed and the problem becomes a standard guidance law development.

The main point of this chapter is to leverage the free design variables κ_4^y and κ_4^z to achieve the minimum control energy objective. We start by considering the QSPT trajectory segments of

$$y_n = a_n^y + c_n^y x + \kappa_n^y x^2, \quad z_n = a_n^z + c_n^z x + \kappa_n^z x^2, \tag{10.6}$$

and note that these trajectories satisfy the equations of (10.5) if they equivalently satisfy

$$\frac{\partial y}{\partial x} = \tan\chi, \quad \frac{\partial z}{\partial x} = \frac{\tan\gamma}{\cos\chi}, \tag{10.7}$$

at the initial and terminal boundary conditions. This step is key in the optimization of QSPT. Since trajectories exist which satisfies the equations of motion, adjoining the dynamic constraints to the chosen performance index becomes unnecessary since the adjoining process is done to ensure the minimization of J satisfies the original equations. Being able to circumvent this step is a tremendous advantage since it essentially converts the overall optimization problem to one which is a parameter optimization problem.

Three Dimensional Optimization

Beginning with the simple performance index of

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^T \mathbf{R}\mathbf{u} dt, \tag{10.8}$$

where the control

$$\mathbf{u} = \left[u_\chi \; u_\gamma\right]^T, \tag{10.9}$$

is a vector of the kinematic controls given by (9.1) and superscript T denotes the transpose. For convenience, the 2×2 weighting matrix $\mathbf{R}$ is chosen as unity.

The flightpath and heading angles are written explicitly in terms of QSPT as

$$\chi = \tan^{-1}\left(\frac{\partial y}{\partial x}\right), \quad \gamma = \tan^{-1}\left(\frac{\frac{\partial z}{\partial x}}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2}}\right). \tag{10.10}$$

where

$$\frac{\partial y}{\partial x} = c_n^y + 2\kappa_n^y x, \qquad \frac{\partial z}{\partial x} = c_n^z + 2\kappa_n^z x. \tag{10.11}$$

Neglecting gravity, the next step is to utilize the kinematic controls of (9.1) and substitute for the angles as

$$u_\chi = \frac{\partial \chi}{\partial x} V^2 \cos^2\left(\tan^{-1}\left(\frac{\frac{\partial z}{\partial x}}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2}}\right)\right)\cos\left(\tan^{-1}\left(\frac{\partial y}{\partial x}\right)\right) \tag{10.12}$$

$$u_\gamma = \frac{\partial \gamma}{\partial x} V^2 \cos\left(\tan^{-1}\left(\frac{\frac{\partial z}{\partial x}}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2}}\right)\right)\cos\left(\tan^{-1}\left(\frac{\partial y}{\partial x}\right)\right) \tag{10.13}$$

where the partial derivatives, given by (9.6) and repeated here for convenience, are given as

$$\frac{\partial \chi}{\partial x} = \frac{2\kappa_n^y}{1+\left(\frac{\partial y}{\partial x}\right)}, \qquad \frac{\partial \gamma}{\partial x} = \frac{2\left(\kappa_n^z + c_n^y \kappa_n^z \left(\frac{\partial}{\partial x} y_n\right) - c_n^z \kappa_n^y \left(\frac{\partial}{\partial x} y_n\right)\right)}{\sqrt{1+\left(\frac{\partial}{\partial x} y_n\right)^2}\left(1+\left(\frac{\partial}{\partial x} y_n\right)^2 + \left(\frac{\partial}{\partial x} z_n\right)^2\right)}. \tag{10.14}$$

Substitution of the partial derivatives and simplification gives

$$u_\chi = \frac{2\kappa_n^y V^2}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2}\left(1+\left(\frac{\partial y}{\partial x}\right)^2+\left(\frac{\partial z}{\partial x}\right)^2\right)}, \quad u_\gamma = \frac{2V^2\left(\kappa_n^z+\kappa_n^z c_n^y \frac{\partial y}{\partial x}-\kappa_n^y c_n^z \frac{\partial y}{\partial x}\right)}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)}\left(1+\left(\frac{\partial y}{\partial x}\right)^2+\left(\frac{\partial z}{\partial x}\right)^2\right)^{\frac{3}{2}}}, \quad (10.15)$$

Substitution into the performance index of (10.8) gives

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left[\left(\frac{2\kappa_n^y V^2}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2}\left(1+\left(\frac{\partial y}{\partial x}\right)^2+\left(\frac{\partial z}{\partial x}\right)^2\right)^2}\right)^2 \quad (10.16)$$

$$+\left(\frac{2V^2\left(\kappa_n^z+\kappa_n^z c_n^y \frac{\partial y}{\partial x}-\kappa_n^y c_n^z \frac{\partial y}{\partial x}\right)}{\sqrt{1+\left(\frac{\partial y}{\partial x}\right)}\left(1+\left(\frac{\partial y}{\partial x}\right)^2+\left(\frac{\partial z}{\partial x}\right)^2\right)^{\frac{3}{2}}}\right)^2\Bigg] dt. \quad (10.17)$$

It would be highly beneficial to derive the final form of the performance index such that the variable of integration is in terms of x instead of t. The reasons for this is because there is a considerable amount of uncertainty in the flight time. Transforming the variable of integration to distance x allows the integration to be performed between downrange boundary conditions. It will be shown in the simulation section that the integration with respect to time produces identical results as the integration with respect to x using the transformation given here. The transformation is derived as follows. Velocity can be expressed as

$$V = \sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}. \quad (10.18)$$

Factoring out $\dot{x}$ and separating the differential gives

$$Vdt = \sqrt{1+\left(\frac{\partial y}{\partial x}\right)^2+\left(\frac{\partial z}{\partial x}\right)^2}\,dx, \quad (10.19)$$

where the change of variables is

$$dt = \frac{\sqrt{1 + \left(\frac{\partial y}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial x}\right)^2}}{V} dx. \tag{10.20}$$

Applying this transform to the integral of (10.17) and simplifying gives

$$J = \int_{x_0}^{x_T} \left[\frac{2\left(\kappa_n^y\right)^2 V^3}{\left(1 + \left(\frac{\partial y}{\partial x}\right)^2\right)\left(1 + \left(\frac{\partial y}{\partial x}\right)^2 \left(\frac{\partial z}{\partial x}\right)^2\right)^{\frac{3}{2}}} + \frac{2V^3\left(\kappa_n^z + \kappa_n^z c_n^y \left(\frac{\partial y}{\partial x}\right) - k_n^y c_n^z \left(\frac{\partial y}{\partial x}\right)\right)}{\sqrt{1 + \left(\frac{\partial y}{\partial x}\right)^2}\left(1 + \left(\frac{\partial y}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial x}\right)^2\right)^{\frac{5}{2}}} \right] dx. \tag{10.21}$$

A closed form solution for this integral exists but can only be generated by a software program such as Mathcad or some suitable equivalent. The resulting integral is of immense size and complexity which cannot be reproduced here due space constraints. However, a nice feature of modern symbolic integration utilities is that they can produce the result in various forms suitable for direct implementation in a simulation software such as Matlab. The result can be copied and pasted into a Matblab M-file and the results tested. Once a closed form solution for the integral is obtained, finding the minimum value for J is simply a matter of line searching values for κ_4^y and κ_4^z until the a combination is found which produces a minimum J.

As previously discussed, numerical optimizations are of no use for guidance law implementation due to the computational cost involved, however, the κ_4^y, κ_4^z values that result from the line search are found at the initialization stage and remain constant throughout the flight. Therefore, no in-flight numerical routines are required in order to maintain optimality. The closed form solution to the integral of (10.21) makes the numerical process at initialization much faster and easier since numerical integration is not necessary.

Velocity and the Optimal Solution

As a matter of simplicity, the velocity was considered constant in the previous development in order to reach the final form of the performance index integral of (10.21). A logical question to ask is what happens to the optimality of the solution when the velocity is not constant, which is generally the case in reality. A simple observation of (10.21) reveals that the velocity is merely a scalar term, and the true minimization of J depends explicitly on the QSPT parameters. That is not to say that the velocity does not effect the control energy; it in fact directly impacts the control energy. For instance, the control energy generated for two identical trajectories will be less for a missile traveling at a lower speed than one traveling at a higher speed. The point here is that if the QSPT trajectory is optimized properly, it will generate the minimum possible control energy for a given velocity.

Planar Optimization

The three dimensional optimization of the previous section is an important result, however, it would be beneficial to investigate the planar optimization as possibly a simpler sub-optimal solution to the 3D problem. A comparison between the two approaches is given in chapter 14. Certainly if QSPT guidance is to be used in a planar application, then the results of this section should be used as the optimal solution.

The process involves separately optimizing the trajectories in their respective planes of operation, i.e., the xy and xz planes. If gravity is ignored and velocity assumed to be constant, then choosing

$\gamma = 0$ in the kinematic model gives the nonlinear planar equations of motion for the crossrange as

$$\dot{x} = V \cos \chi \tag{10.22}$$

$$\dot{y} = V \sin \chi \tag{10.23}$$

$$\dot{\chi} = \frac{1}{V} u_\chi. \tag{10.24}$$

From this model, the kinematic control is then given as

$$u_\chi = V^2 \frac{\partial \chi}{\partial x} \cos(\chi) \tag{10.25}$$

with the equation for χ for the corresponding n^{th} segment given as

$$\chi_n = \tan^{-1}\left(c_n^y + 2\kappa_n^y x\right). \tag{10.26}$$

The partial derivative is then

$$\frac{\partial \chi}{\partial x} = \frac{2\kappa_n^y}{1 + \left(\frac{\partial y}{\partial x}\right)}. \tag{10.27}$$

The standard performance index is then

$$J = \int_{t_0}^{t_f} u_\chi^2 dt, \tag{10.28}$$

for which a minimization of J results in a minimization of control energy for the crossrange trajectory. Substitution of (10.26) into (10.25) and (10.25) with the corresponding partial derivative into (10.28) gives the performance index for the n^{th} segment as

$$J_n = \int_{t_0}^{t_f} \left[\frac{2\kappa_n^y V^2}{1 + \left(c_n^y + 2\kappa_n^y x\right)^2} \cos\left(\tan^{-1}\left(c_n^y + 2\kappa_n^y x\right)\right) \right]^2 dt, \tag{10.29}$$

which simplifies to

$$J_n = \int_{t_0}^{t_f} \frac{4\left(\kappa_n^y\right)^2 V^4}{\left(1+\left(c_n^y + 2\kappa_n^y x\right)^2\right)^3} dt. \tag{10.30}$$

We derive a change in the variable of integration in a similar manner as the transformation used in the three dimensional optimization as

$$dt = \frac{\sqrt{1+\left(c_n^y + 2\kappa_n^y x\right)^2}}{V} dx, \tag{10.31}$$

which when substituted for dt in (10.30) gives

$$J_n = \int_{x_0}^{x_f} \frac{4\left(\kappa_n^y\right)^2 V^3}{\left(1+\left(c_n^y + 2\kappa_n^y x\right)^2\right)^{\frac{5}{2}}} dx. \tag{10.32}$$

A closed form solution for the integral of (10.32) is then given as

$$J_n = \frac{2\kappa_n^y V^3 \left(2\left(c_n^y\right)^3 + 12\left(c_n^y\right)^2 \kappa_n^y x + 24 c_n^y \left(\kappa_n^y\right)^2 x^2 + 3c_n^y + 16\left(\kappa_n^y\right)^3 x^3 + 6\kappa_n^y x\right)}{3\left[1+\left(c_n^y + 2\kappa_n^y x\right)^2\right]^{\frac{3}{2}}}, \tag{10.33}$$

which when evaluated at the boundary conditions for the corresponding n^{th} segment, can then be minimized with respect to κ_4^y as

$$J = \min_{\kappa_4^y} \left\{ J_1\Big|_{x_0}^{x_\alpha} + J_2\Big|_{x_\alpha}^{x_\beta} + J_3\Big|_{x_\beta}^{x_\delta} + J_4\Big|_{x_\delta}^{x_T} \right\}. \tag{10.34}$$

which results in the optimizing value ${}^*\kappa_4^y$. The minimization of (10.34) involves a simple numerical line search of κ_4^y values at the initialization stage before launch. This same process is followed for choice of $\chi = 0$ which results in an altitude trajectory minimization of control energy. If gravity is ignored, the same general design equations result. Simulation results will show the effectiveness of the approach.

CHAPTER 11: IMPROVED TIME-TO-GO

In chapter 6, we discussed standard methods for the calculation of time-to-go and the potential error that can corrupt the calculation. In this chapter, a closed form solution for the arc-length of a QSPT trajectory is derived. The resulting time-to-go calculation is free from the error due to trajectory curvature. Constant velocity is considered first and then the case of non constant velocity will be addressed. The equation for time-to-go is given as

$$T_{go} = \frac{S}{V} \tag{11.1}$$

where S is the closed form solution for trajectory arc-length and is

$$S = \int_{x_0}^{x_T} \sqrt{1 + \left(\frac{\partial y}{\partial x}\right)^2 + \left(\frac{\partial y}{\partial x}\right)^2}. \tag{11.2}$$

The equation of (11.2) must be integrated from the initial position to the terminal position while spanning all four segments subject to the corresponding internal boundary conditions. In light of this, the integral of (11.2) becomes

$$\begin{aligned} S = &\int_{x_0}^{x_\alpha} \sqrt{1 + \left(\frac{\partial}{\partial x} y_1(x)\right)^2 + \left(\frac{\partial}{\partial x} z_1(x)\right)^2}\, dx + \int_{x_\alpha}^{x_\beta} \sqrt{1 + \left(\frac{\partial}{\partial x} y_2(x)\right)^2 + \left(\frac{\partial}{\partial x} z_2(x)\right)^2}\, dx \\ &+ \int_{x_\beta}^{x_\delta} \sqrt{1 + \left(\frac{\partial}{\partial x} y_3(x)\right)^2 + \left(\frac{\partial}{\partial x} z_3(x)\right)^2}\, dx + \int_{x_\delta}^{x_T} \sqrt{1 + \left(\frac{\partial}{\partial x} y_4(x)\right)^2 + \left(\frac{\partial}{\partial x} z_4(x)\right)^2}\, dx. \end{aligned} \tag{11.3}$$

A closed form general solution of the integrals can be found representing each n^{th} segment as

$$S_n = \frac{1}{4}\left[\frac{a + b\ln(c) + d\ln(2)}{\left(\left(\kappa_n^y\right)^2 + \left(\kappa_n^z\right)^2\right)^{\frac{3}{2}}}\right], \tag{11.4}$$

where

$$\begin{aligned}
a &= \left[\kappa_n^y \frac{\partial}{\partial x} y_n(x) + \kappa_n^z \frac{\partial}{\partial x} z_n(x)\right] \sqrt{\left(\kappa_n^y\right)^2 + \left(\kappa_n^z\right)^2} \sqrt{1 + \left(\frac{\partial}{\partial x} y_n(x)\right)^2 + \left(\frac{\partial}{\partial x} z_n(x)\right)^2} \\
b &= \left(\kappa_n^y\right)^2 + \left(\kappa_n^z\right)^2 + \left(c_n^y\right)^2 \left(\kappa_n^z\right)^2 + \left(c_n^z\right)^2 \left(\kappa_n^y\right)^2 - 2c_n^y \kappa_n^y c_n^z \kappa_n^z \\
c &= \kappa_n^y \frac{\partial}{\partial x} y_n(x) + \kappa_n^z \frac{\partial}{\partial x} z_n(x) + \sqrt{\left(\kappa_n^y\right)^2 + \left(\kappa_n^z\right)^2} \sqrt{1 + \left(\frac{\partial}{\partial x} y_n(x)\right)^2 + \left(\frac{\partial}{\partial x} z_n(x)\right)^2} \\
d &= 2\left(\kappa_n^y\right)^2 + 2\left(\kappa_n^z\right)^2 + 2\left(c_n^y\right)^2 \left(\kappa_n^z\right)^2 + 2\left(c_n^z\right)^2 \left(\kappa_n^y\right)^2 - 4c_n^y \kappa_n^y c_n^z \kappa_n^z.
\end{aligned}$$

The total initial arc-length is then given as

$$S = S_1\Big|_{x_0}^{x_\alpha} + S_2\Big|_{x_\alpha}^{x_\beta} + S_3\Big|_{x_\beta}^{x_\delta} + S_4\Big|_{x_\delta}^{x_T}. \tag{11.5}$$

Equation (11.5) gives the initial arc length of the trajectory at the beginning of the engagement. In real time applications, (11.5) can be continuously updated with the current downrange value for an updated time-to-go calculation. It will be shown in chapter 14 that under constant velocity, (11.1) produces a linear response over the engagement and is therefore not subject to error due to trajectory curvature. In this case, the time-to-go initially computed at the beginning of the engagement is, in fact, the impact time.

Equations for Impact Time

The removal of curvature error is a substantial improvement over existing methods for computing time-to-go. If the velocity is constant along the trajectory, then the time-to-go response over the engagement is perfectly linear and the impact time is easily determined to be the initial time-to-go. Since the velocity is constant, the impact time remains constant in time. This can be proven in the following manner. The impact time is computed as

$$T_I = T_{go} + t. \tag{11.6}$$

Finding the time derivative of (11.6) gives

$$\dot{T}_I = \dot{T}_{go} + 1, \tag{11.7}$$

where the time-to-go rate is derived from (11.1) as

$$\dot{T}_{go} = \frac{V\dot{S} - S\dot{V}}{V^2}, \tag{11.8}$$

and therefore the impact time rate is

$$\dot{T}_I = \frac{V\dot{S} - S\dot{V}}{V^2} + 1. \tag{11.9}$$

The time-rate-of-change of arc-length, $\dot{S}$, is the negative of the velocity tangent to the curve. That is, $\dot{S} = -V$. If the velocity is also constant, $\dot{V} = 0$, equation (11.9) reduces to

$$\dot{T}_I = 0, \tag{11.10}$$

thus demonstrating that for constant velocity, the impact time remains constant in time and equal to the initially computed time to go.

A brief observation of equation (11.9), however, shows that when the velocity is not constant, or $\dot{V} \neq 0$, equation (11.9) is nonzero and given by

$$\dot{T}_I = -\frac{\dot{V}S}{V^2}. \tag{11.11}$$

Therefore, acceleration of the missile along the trajectory causes changes in the impact time over the engagement. This can be problematic from the fact that exact changes in a missiles acceleration can never be known exactly due to unknown or unmodeled disturbances in the atmosphere. One way to deal with such problems is to conduct preflight modeling and simulation of the mission in order to gain nominal estimates for $\dot{V}$ and ultimately a reference model profile of T_I over the engagement.

Preflight Analysis Methods

The standard approach for conducting a preflight analysis involves integrating the chosen guidance law into the future to find the point of closest approach to the target. This can be an involved process because most guidance laws are explicit in time-to-go. The resulting estimated trajectory and corresponding time-to-go can vary considerably if any disturbances are present. A particular advantage provided by QSPT is that once all of the required boundary conditions are determined, a trajectory fixed in space results and the arc-length is immediately known. Therefore, with a fixed path established, the missiles motion along the trajectory can be simulated before launch using nonlinear drag tables as well as nonlinear atmospheric models. This results in more reliable estimates of the acceleration along the trajectory and hence, the changes in the impact time.

Unmodeled disturbances can be corrected by the impact time control.

The first step in the analysis process is to design a desired QSPT trajectory by establishing the boundary conditions for the engagement. Trajectory set (7.8) along with the corresponding coefficient equations of (8.10), (8.12), (8.13), (8.14), and (8.16) are implemented. The next step involves simulating equations (7.1)-(7.6) with the closed loop guidance law equations of (9.16). The simulation can output equation (11.6) plotted over the downrange distance as shown in figure 11.1.

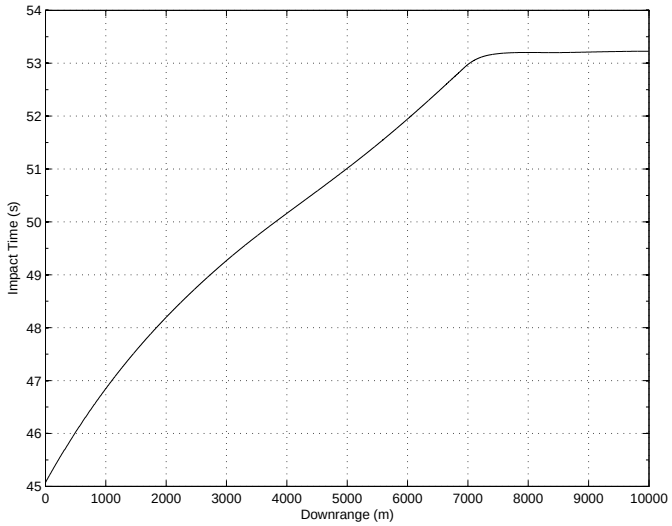

Figure 11.1: Preflight impact time model

Preflight data modeling such as the one produced in figure 11.1 provides a desired reference model for the impact time control to use when adjusting for unmodeled disturbances. In the plot, the effects of the acceleration along the trajectory can be observed. At approximately 7000 meters downrange, the munition begins to reach terminal velocity and the acceleration reduces to low levels. Because of the effects of equation (11.9) and (11.10), the projected final impact time converges to a constant value. In the next chapter a very simple impact time control is designed.

CHAPTER 12: IMPACT TIME CONTROL DESIGN

In this chapter, we leverage the family of trajectories provided by QSPT and the impact time reference model derived at preflight to design a very simple, yet effective impact time controller, which for simplicity, was chosen as a Bang-Bang control. Certainly more advanced designs are possible. The control is designed for the free variable state model of (8.17) shown again for convenience as

$$\dot{\kappa}_4^y = \mu, \tag{12.1}$$

to smoothly and continuously adjust the trajectory to compensate for error between the real-time calculation of the impact time and the preflight reference model. To that end, the real-time calculated impact time is given from (11.6) as

$$T_I = T_{go} + t, \tag{12.2}$$

Equation (12.2) is expected to contain some error due to unmodeled disturbance. We define the impact time reference model developed from preflight as $\hat{T}_I$. Formulating the impact time error gives

$$e = T_I - \hat{T}_I. \tag{12.3}$$

The Bang-Bang control for the prescribed linear system of (8.17) is then given as

$$\mu = \begin{cases} \delta & \text{if} \quad e > 0 \\ 0 & \text{if} \quad e = 0 \\ -\delta & \text{if} \quad e < 0, \end{cases} \tag{12.4}$$

where δ is found empirically. The control of (12.4) is designed to command the state trajectory of κ_4^y to a location which drives the error in (12.3) to zero, or as close as possible. A decision logic is employed to ensure the state trajectory of κ_4^y travels in a direction which minimizes (12.3). It should also monitor the rate-of-change of error versus the amount of applied control μ, since a reduction in error may not always be possible due to wind disturbance late in flight when little trajectory planning can be applied to correct the error. The performance of this simple control will be shown to be effective in reducing the impact time error to fractions of a second in the presence of disturbances.

CHAPTER 13: GUIDANCE LAW COMPARISON FOR PERFORMANCE EVALUATION

The QSPT guidance law has been developed to provide certain advantages over standard optimal guidance laws. In chapter 14, those advantages will be demonstrated through simulation. It should also be shown that the performance of QSPT is comparable to other guidance laws with respect to various measures of optimality such as control energy requirements as well as the satisfaction of any required terminal constraints such as miss distance and terminal impact angle. This chapter is dedicated to preparing the comparison models as well as establishing a fair evaluation criterion.

The Genex Guidance Law

The Genex guidance law presented here is reproduced from [27], and will be used to compare QSPT against. The simplest form of Genex is given below as

$$\mathbf{u} = \frac{V^2}{R}\left[K_1\left(\hat{\mathbf{r}} - \hat{\mathbf{v}}\right) + K_2\left(\hat{\mathbf{v}}_f - \hat{\mathbf{v}}\right)\right], \tag{13.1}$$

where

$$K_1 = (n+2)(n+3), \qquad K_2 = -(n+1)(n+2), \tag{13.2}$$

and the guidance commands

$$\mathbf{u} = \left[u_x, u_y, u_z\right]^T, \tag{13.3}$$

are given in inertial coordinates. The gains are a function of n, a user selected integer gain. An acceptable initial setting for n is 0, however, increasing the value of n increases the trajectory curvature. Therefore, a family of trajectories is possible with Genex, which makes it a suitable law

for comparison. This form of Genex assumes longitudinal controllability, which is not the case for a wide range of missiles. Another form of Genex that addresses this issue is given as

$$\mathbf{u} = \frac{V^2}{R}\left[K_1\left(\hat{\mathbf{r}} - \hat{\mathbf{v}}\cos\delta\right) + K_2\left(\hat{\mathbf{v}}_f - \hat{\mathbf{v}}\cos\mu\right)\right], \tag{13.4}$$

where

$$\cos\delta = \hat{r}\cdot\hat{v}, \quad \cos\mu = \hat{v}_f\cdot\hat{v}, \tag{13.5}$$

and the guidance commands are normal to V and given in inertial coordinates. For the purposes of this work, (13.4) will be utilized since the under-actuation problem is being considered. The primary issue to consider is that the guidance command $\mathbf{u}$ of equation (13.4) is given in terms of inertial coordinates. Unfortunately, the equations of motion given in (13.6) require the guidance commands in terms of u_χ and u_γ, and therefore, a transformation must be derived in order to use Genex with our chosen equations of motion.

Model Used for Comparison

Both QSPT and Genex will be simulated using the kinematics given below as

$$\begin{aligned}
\dot{V} &= -ad - g\sin\gamma \\
\dot{\chi} &= \frac{1}{V\cos\gamma}u_\chi \\
\dot{\gamma} &= \frac{1}{V}\left(u_\gamma - g\cos\gamma\right) \\
\dot{x} &= V\cos\gamma\cos\chi \\
\dot{y} &= V\cos\gamma\sin\chi \\
\dot{z} &= V\sin\gamma,
\end{aligned} \tag{13.6}$$

where $g = -9.81\frac{m}{s^2}$ and the negative sign has been included in the equations.

The QSPT guidance law issues control commands in u_χ and u_γ, making QSPT fully compatible with the given kinematic model. The Genex guidance law, on the other hand, issues acceleration guidance commands in inertial components, i.e., $u_x = \ddot{x}, u_y = \ddot{y}, u_z = \ddot{z}$. Therefore, a coordinate transformation must be derived such that commands are generated in terms of the control inputs u_χ, u_γ required by (13.6).

The transformation of Genex command u_z into u_γ is easily found from the derivative of $\dot{z}$ in (13.6) as

$$\ddot{z} = u_z = \cos\gamma\dot{\gamma} = \cos\gamma u_\gamma, \tag{13.7}$$

and therefore the transformation is found as

$$u_\gamma = \frac{u_z}{\cos\gamma}. \tag{13.8}$$

The remaining transformation for u_x and u_y into u_γ can be found by determining $\ddot{x}$ and $\ddot{y}$ from (13.6) as

$$\ddot{x} = u_x = -\sin\gamma\sin\chi u_\gamma + \cos\chi u_\chi \tag{13.9}$$

$$\ddot{y} = u_y = -\sin\gamma\cos\chi u_\gamma - \sin\chi u_\chi. \tag{13.10}$$

Solving (13.9) for u_γ in gives

$$u_\gamma = \frac{u_x + \sin\chi u_\chi}{-\sin\gamma\cos\chi}, \tag{13.11}$$

which is substituted into (13.10) as

$$\begin{aligned} u_y &= -\sin\gamma \sin\chi \left[\frac{u_x + \sin\chi u_\chi}{-\sin\gamma\cos\chi}\right] + \cos\chi u_\chi \\ &= \tan\chi \left[u_x + \sin\chi u_\chi\right] + \cos\chi u_\chi, \end{aligned} \tag{13.12}$$

which then leads to the final transformation of

$$u_\chi = \cos\chi \left[u_y - u_x \tan\chi\right]. \tag{13.13}$$

With the given input transformations, the Genex guidance law is now compatible with the equations of motion given in (13.6).

CHAPTER 14: SIMULATION STUDY

The following simulation results demonstrate the effectiveness of the optimization algorithms, the improved performance of the time-to-go algorithm as well as show robustness of the guidance law to measurement error. Two impact time control cases are also presented. Case 1 analyzes the ability of the proposed guidance law to reduce the impact time error under a range of unknown disturbances. Case 2 is similar but employs a different engagement scenario.

QSPT Optimization Performance

Table 14.1 shows control energies with corresponding arc-lengths computed for the crossrange trajectory. The initial conditions of $x_0 = 0$, $y_0 = 0$, $\chi_0 = -\frac{\pi}{4}$, and the terminal conditions of $x_T = 10,000$, $y_T = 10,000$, $\chi_T = -\frac{\pi}{4}$, were chosen.

Table 14.1: Crossrange control energy minimization

κ_4^y value	total control energy	arc-length
-8.0000×10^{-4}	1.7111×10^5	1.7001×10^4
-7.0000×10^{-4}	1.6896×10^5	1.6437×10^4
-5.0000×10^{-4}	1.6879×10^5	1.6123×10^4
$-^*4.5929 \times 10^{-4}$	1.6859×10^5	1.6173×10^4
-3.0000×10^{-4}	1.6893×10^5	1.6777×10^4
-2.0000×10^{-4}	1.8257×10^5	1.7571×10^4
-1.0000×10^{-4}	2.1755×10^5	1.8737×10^4

The numeric minimization of (10.34) for the given boundary conditions results in ${}^*\kappa_n^y = -4.5929 \times 10^{-4}$, the starred value in the table. Multiple simulation runs were executed for κ_4^y on either side of the computed minimum with the corresponding control energy. Results show equation (10.34) to be an effective optimization with respect to control energy.

The three dimensional optimization produces the same result for three dimensional trajectories, however, the line search of κ_4^y and κ_4^z is more intense from a computational standpoint.

Monte Carlo Performance Comparison

The previous sections analyzed the control energy optimality of QSPT and then compared that performance to that of Genex. However, other measures of a guidance laws performance should be taken into consideration, such as terminal miss distance error and terminal angle error. In many cases, the terminal error measures are more important than the minimization of control energy due to lethality requirements. In the following Monte Carlo test, Terminal miss distance, terminal angle error, and control energy are measured in the presence of random wind disturbance. The statistical properties are as follows,

Wind: 0 to $20\frac{m}{s}$, random draw, 1σ

Wind Direction: 0 to 2π, random draw, 1σ.

In addition to the random variables, a dynamic lag is added to the equations of motion to simulate the effects of the airframe/flight control and the damping and natural frequency is given as $\zeta = 0.60$, $\omega_n = 1hz$. For these given conditions, the Monte Carlo simulation was run 200 times each for QSPT and Genex with the following average and worst case values of These results not only show that QSPT is more optimal in the sense of control energy expenditure, but the ability to achieve the desired terminal constraints with less error is better as well.

Table 14.2: Average values for 200 runs each

	Terminal Miss	Control Energy	γ_{T_e}	χ_{T_e}
QSPT	0.3421	7.6203×10^3	0.0035	0.0037
Genex	0.4168	7.6744×10^3	0.0077	0.0097

Table 14.3: Worst case value over 200 runs

	Terminal Miss	Control Energy	γ_{T_e}	χ_{T_e}
QSPT	0.3963	8.488×10^3	0.0942	0.0478
Genex	0.4603	9.7868×10^3	0.0389	0.0928

Comparison of Guidance Commands

In this section, a comparison of the guidance commands for QSPT and Genex is given using the initial and terminal conditions of table 14.5. Figures 14.1 and 14.2 show the u_χ and u_γ guidance commands, respectively. The step discontinuities in the commands occur at the points of inflection of the trajectory at the downrange segmentation points of x_α and x_δ. Figures 14.3 and 14.4 show the u_χ and u_γ commands generated by the Genex guidance law. Since commands generated by each of the guidance laws are lateral accelerations, the step discontinuities found in the QSPT commands are as expected because of the second order nature of QSPT.

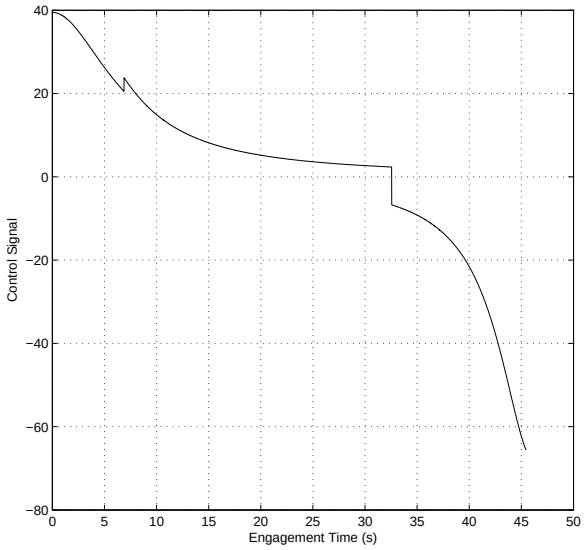

Figure 14.1: u_χ command for QSPT

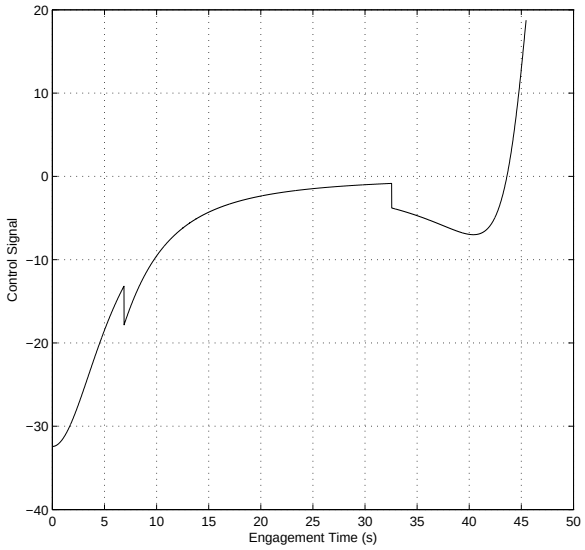

Figure 14.2: u_γ command for QSPT

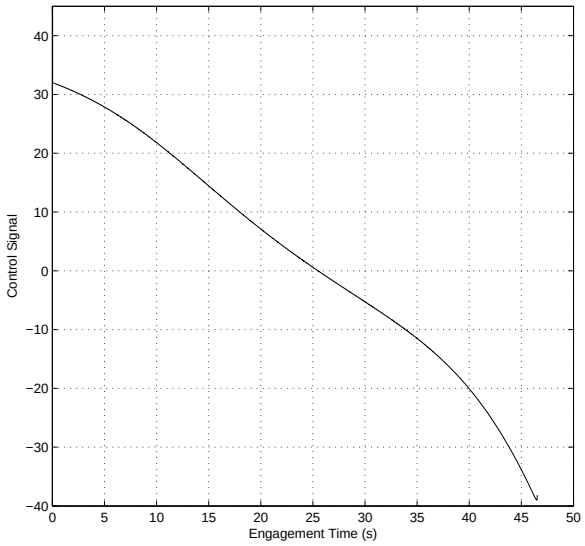

Figure 14.3: u_χ command for Genex

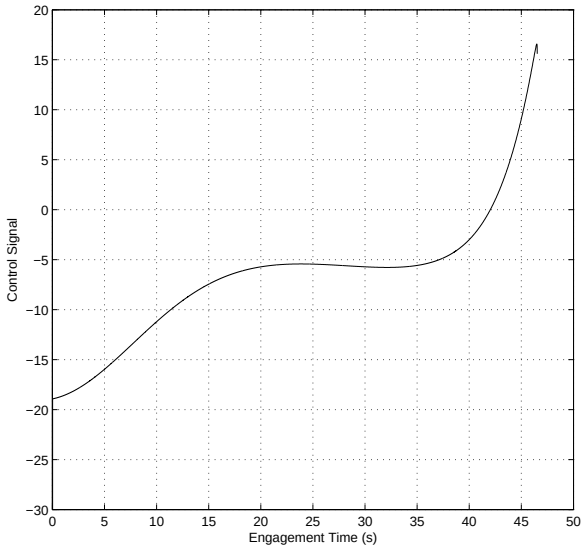

Figure 14.4: u_γ command for Genex

Performance of Time-to-go Algorithm

The time-to-go comparison plot of fig. 14.6 was generated for constant velocity along the trajectory of fig. 14.5, which has a considerable amount of curvature. For clarity, the crossrange/downrange projection of the trajectory has been included in all 3-D plots. Figure 14.6 demonstrates that the time-to-go response generated by (11.1) over the engagement is linear and is thus not effected by curvature of the trajectory. In other types of applications where the velocity along the trajectory could actually be constant, the initially calculated time-to-go is in fact the final impact time as seen in figure 14.6. The performance of the switched form guidance law and reparameterized trajectory of the appendix can also be observed in figure 14.5. The switched form guidance law takes control at the downrange, crossrange, and altitude positions of $10,000$, $10,000$, 1000, respectively.

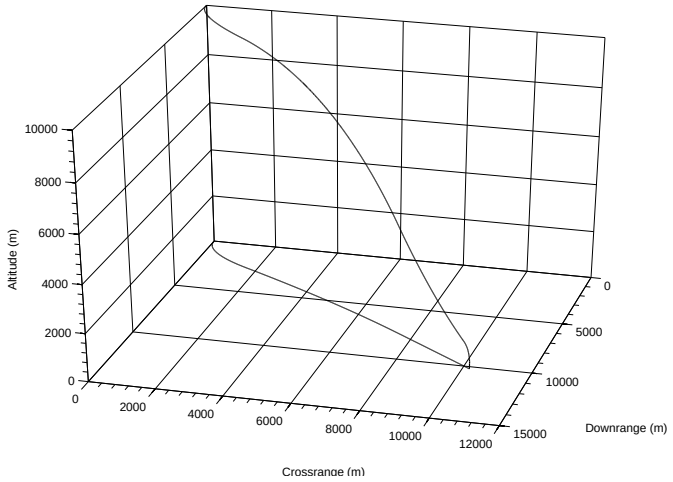

Figure 14.5: Trajectory with Curvature

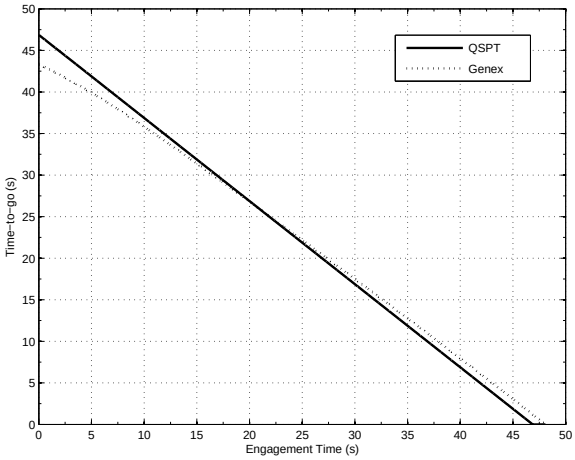

Figure 14.6: Comparison of QSPT time-to-go versus Genex

In contrast, it can be seen in figure 14.6 the time-to-go generated with the Genex guidance law of [6] which uses the range-over-missile-velocity method. The velocity was constant and the initial and terminal angles were the same ones used to generate fig. 14.5. The effects of trajectory curvature on the standard time-to-go calculation can clearly be seen. The initial calculation of time-to-go is 43.3 seconds but the actual final impact time is 48 seconds, nearly a 5 second disparity in time even for constant velocity and a stationary target.

Robustness of the Guidance Law

The remaining simulation results in this section were generated using a 1962 Standard Atmosphere along with tabular drag profiles for a generic projectile. It is assumed that the positional measurements provided by the IMU are perfect. However, since the inversion matrix of the guidance law is explicit in angles γ and χ, gyro errors consistent with a tactical grade IMU are considered. In addition, a percent-error in sensed drag acceleration a_d is considered as well. Table 14.4 lists the

error values used to obtain the following results.

Table 14.4: Sensor and gyro error

Error Type	Units	Measure	Value
Drag Acceleration	$\frac{m}{s^2}$	%	3.0
Gyro Bias	$\frac{deg}{HR}$	1σ	3.0
Gyro Scale Factor	ppm	1σ	300
Gyro Random Walk Noise	$\frac{deg}{\sqrt{HR}}$	nom	0.02

Gyro bias, scale factor, and noise are considered in addition to 3% error in the sensed drag acceleration. Figures 14.7 and 14.8 contrast the measured values against the true values for χ and γ, respectively, produced along the trajectory of figure 14.5.

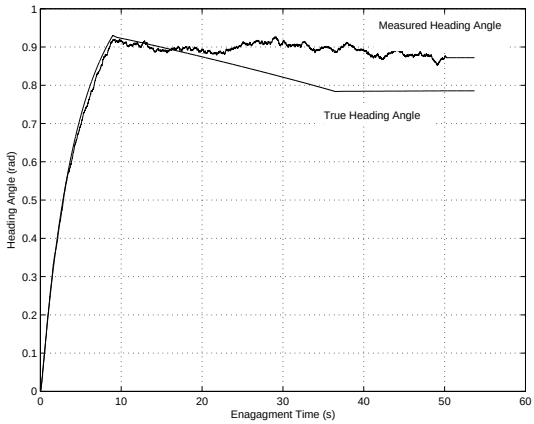

Figure 14.7: True heading angle versus measured

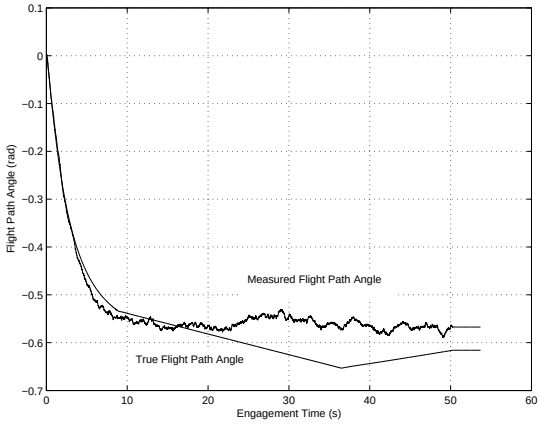

Figure 14.8: True flight path angle versus measured

Figures 14.9 and 14.10 show the tracking error performance of the guidance law using perfect measurements of χ and γ. In contrast, figures 14.11 and 14.12 show the crossrange and altitude tracking performance of (9.16) in response to the measurement errors listed in table 14.4. The proposed guidance law demonstrates good tracking performance with negligible deterioration.

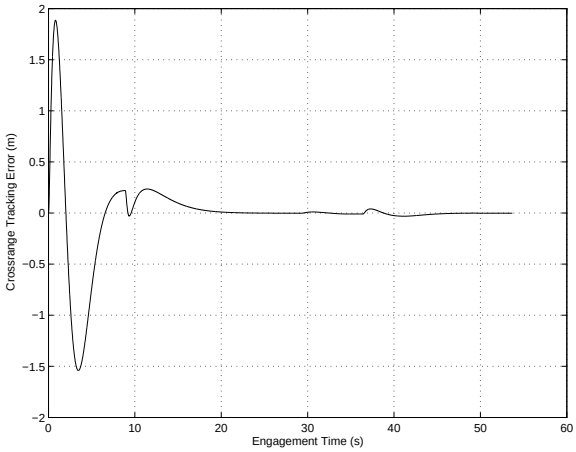

Figure 14.9: Crossrange tracking performance under no disturbance

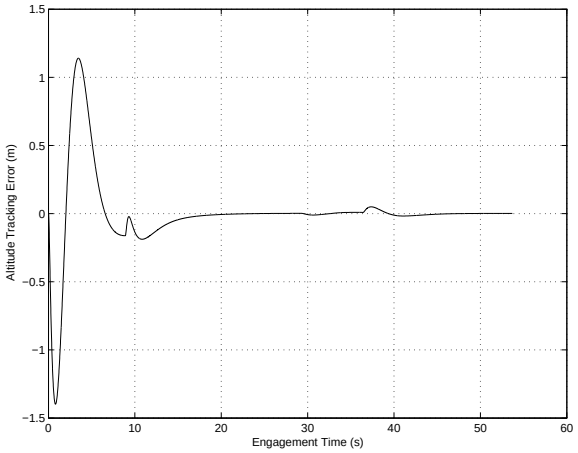

Figure 14.10: Altitude tracking performance under no disturbance

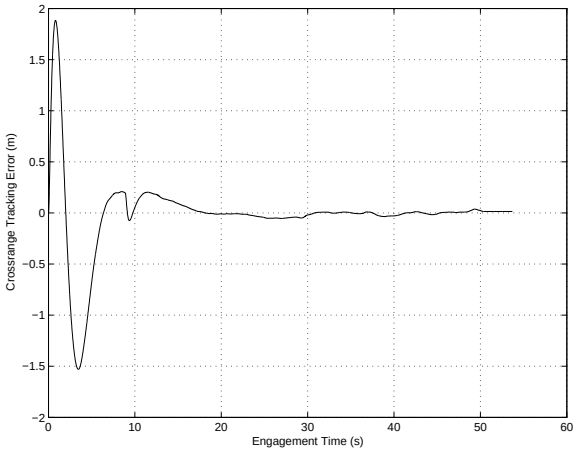

Figure 14.11: Crossrange tracking performance subject to the disturbances of table 1

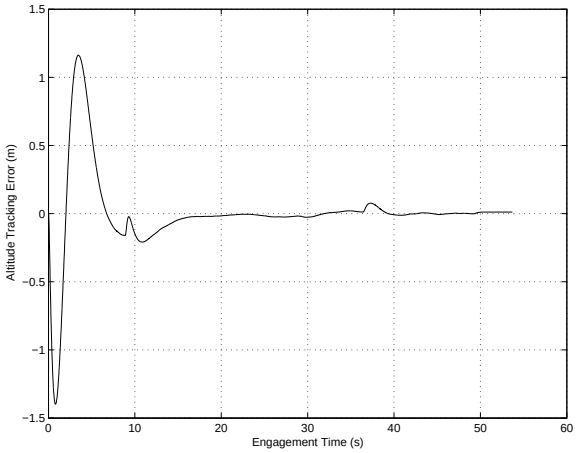

Figure 14.12: Altitude tracking performance subject to the disturbances of table 1

Impact Time Control-Case 1

For the final portion of this study, an unknown wind disturbance of $15\frac{m}{s}$ in the positive down-range direction is considered and in addition to the measurement errors of table 14.4, an unknown lumped nonlinear disturbance is also employed. The wind disturbance spans the full altitude of the engagement from 10,000 meters to the ground and remains constant over that range. The combined effect of the wind and disturbances without any control over impact time produces a final impact time of 48.825 seconds. The pre-flight analysis considering drag and atmosphere, but excluding the unknown disturbances, determines a desired impact time of 53.7 seconds. This requires the corresponding impact time control to correct for an error of 4.875 seconds over the the engagement. The desire is to reduce the error in impact time to fractions of a second under these given conditions. Table 14.5 provides the initial and terminal conditions for the engagement.

Table 14.5: Initial and terminal conditions, case 1

Variable	Units	Initial	Terminal
Velocity	$\frac{m}{s}$	400	–
Downrange	m	0	10,100.0
Downrange	m	0	10,100.0
Altitude	m	10,000.0	0
χ	radians	0	$\frac{\pi}{4}$
γ	radians	0	$-\frac{\pi}{2}$
Required Impact Time	s	–	53.70

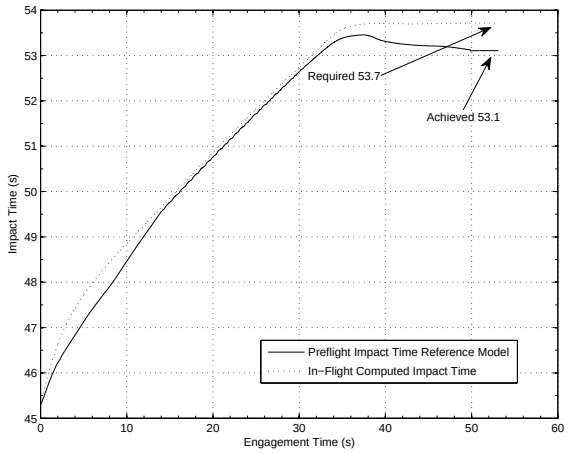

Figure 14.13: Comparison between impact time profiles, desired versus achieved

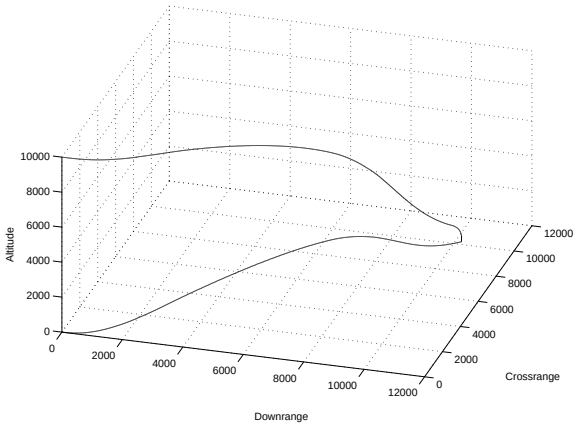

Figure 14.14: Resulting trajectory under impact time control

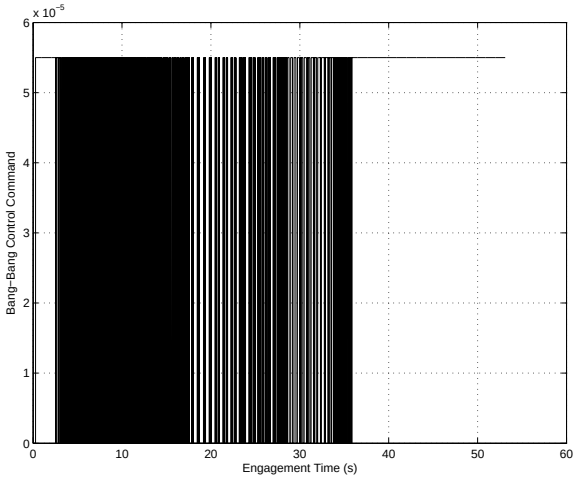

Figure 14.15: Bang bang control commands

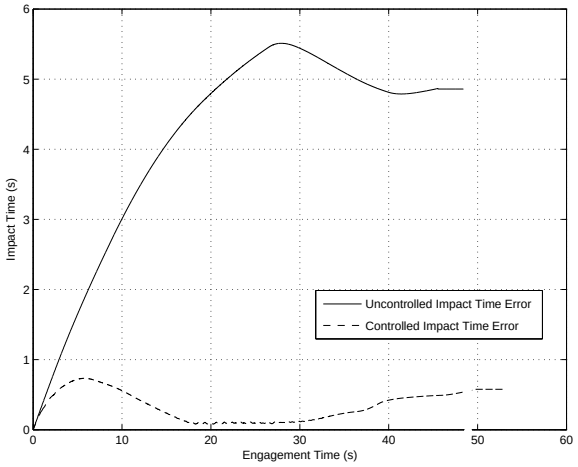

Figure 14.16: Error comparison between controlled and uncontrolled impact time

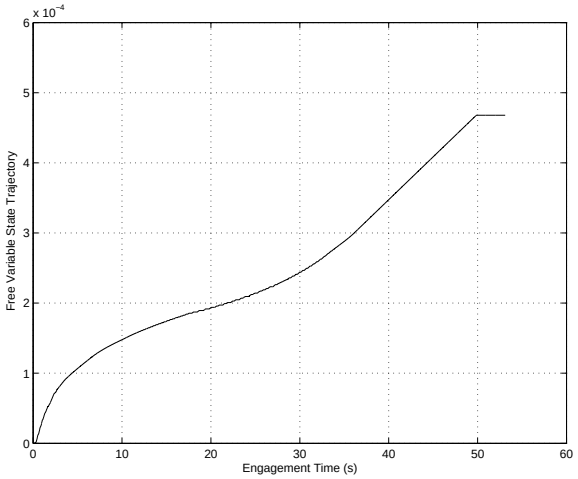

Figure 14.17: State trajectory of κ_4^y

Figure 14.13 contrasts the difference between the impact time reference model and the resulting in-flight impact time and shows that under the Bang-Bang control signals of figure 14.15, the corresponding impact time error was reduced to 0.577 seconds. Figure 14.17 shows the state

trajectory for κ_4^y generated by the control and figure 14.14 is the trajectory that results. Figure 14.16 compares the effect of the control on the impact time error. In that figure, the uncontrolled impact time response is shown against the controlled impact time. The proposed guidance law demonstrates a good ability to reduce the error of (12.3) under heavy disturbances.

In the previous simulation scenario, the constant wind disturbance extended from apogee to the ground. It is difficult to exactly achieve a prescribed impact time under these conditions since the correction of impact time error relies on trajectory planning. Therefore, the ability to correct for disturbances late in flight diminishes. In addition, the trajectory must satisfy impact angle requirements and thus takes priority over satisfying impact time. In the next simulation run, we consider the wind to taper off with lower altitude and reduce to zero near the ground in order to show an improvement in impact time control.

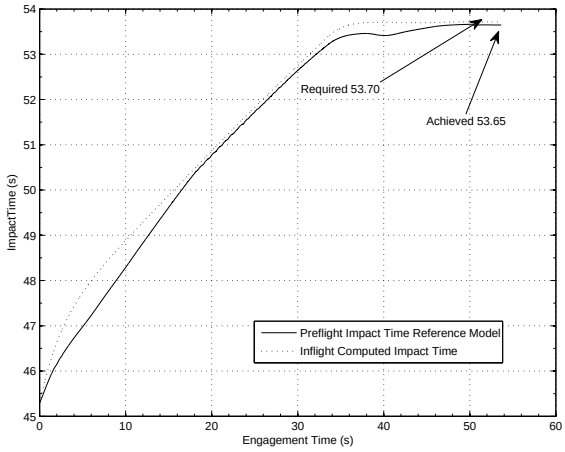

Figure 14.18: Impact time with wind tapering off

Figure 14.18 shows a considerable improvement in the impact time error when the wind diminishes in the lower altitudes. In this scenario, the impact time control is capable of reducing the error to

0.05 seconds, a vast improvement if the disturbances are minimal in the final few seconds of the engagement.

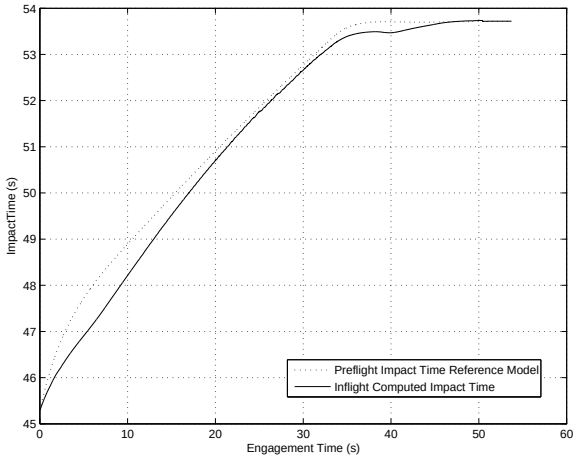

Figure 14.19: Impact time with no wind

Figure 14.19 shows a perfect satisfaction of the desired impact time with negligible or no wind.

Few works exist in the open literature to conduct a fair comparison against. However, the work of [11] considers a time-of-flight control problem for a guided projectile with several error sources. While the overall scope of that work also considered range maximization, some brief comparisons can be made. In [11], a prescribed impact time is achieved as long as perfect knowledge of both muzzle exit velocity and wind is known. The performance of the algorithm is expected to deteriorate under measurement error, and the requirement is then to bring the impact time dispersal between weapons to within approximately 2.0 seconds. As demonstrated in the previous three impact time control scenarios, the proposed QSPT guidance algorithm can reduce the impact time error to well below 1 second under a series of unknown disturbances.

Impact Time Control-Case 2

In case 2, all of the previous measurement errors are considered and an unknown wind disturbance of $7\frac{m}{s}$ in the positive downrange direction is present. Table 14.6 details the engagement parameters.

Table 14.6: Initial and terminal conditions, case 2

Variable	Units	Initial	Terminal
Velocity	$\frac{m}{s}$	400	–
Downrange	m	0	10,000.0
Downrange	m	0	10,000.0
Altitude	m	10,000.0	0
χ	radians	$\frac{\pi}{4}$	$-\frac{\pi}{18}$
γ	radians	0	$-\frac{\pi}{4}$
Required Impact Time	s	–	56.16

From the preflight analysis stage, a desired impact time was determined to be 56.16 seconds. The unknown wind disturbance causes the projectile to reach the target earlier at 54.17 seconds. The corresponding impact time control reduced the impact time error to 0.298 seconds as shown in figure 14.22. Figure 14.20 shows the optimized trajectory set by ${}^*\kappa_4^y$ and ${}^*\kappa_4^z$ with no impact time adjustment. In contrast, figure 14.21 shows the trajectory which results from impact time control.

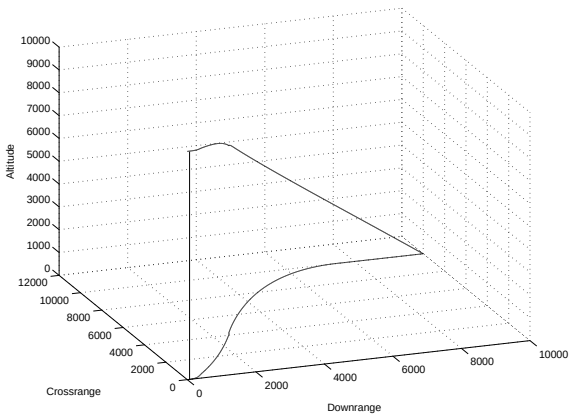

Figure 14.20: Minimum curvature trajectory with no impact time control

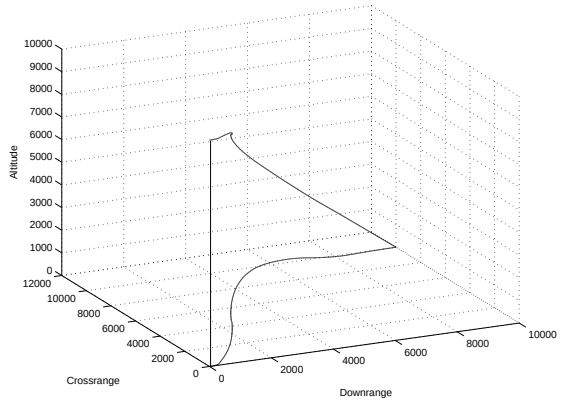

Figure 14.21: Trajectory resulting from controlled impact time

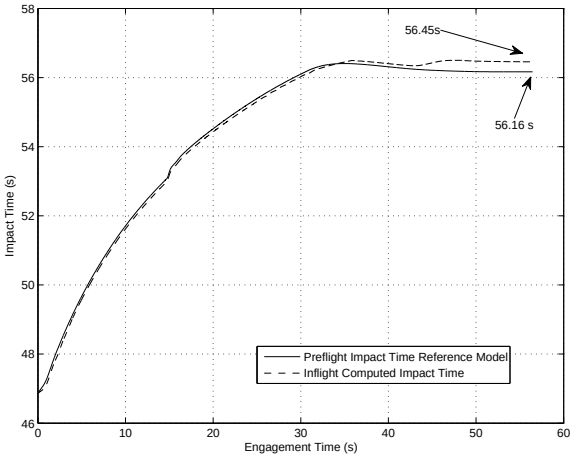

Figure 14.22: Preflight reference model vs. inflight computed impact time

CHAPTER 15: CONCLUSION

Simulation results show a considerable improvement in the time-to-go calculation of QSPT versus the result generated by the Genex guidance law using the range-over-missile-velocity computation. For the constant velocity comparison, error due to trajectory curvature has been eliminated, thus improving the fidelity of the result. For applications where the velocity along the trajectory is constant, the initially calculated time-to-go is in fact the final impact time.

Further comparison with Genex shows that QSPT is more optimal with respect to control energy expenditure over the flight. Since the design of Genex assumes a linear system, the presence of nonlinearity as well as accelerations tend to degrade the optimal solution, which is not the case with QSPT since nonlinearity of the system is considered in the generation of the guidance law. A better comparison measure is one in which terminal constraint satisfaction as well as control energy is considered. In the Monte Carlo runs that were presented, QSPT is shown to be an improvement over Genex with respect to terminal miss distance error, terminal impact angle error, and control energy expenditure for a wind range of wind disturbances.

In the final comparisons, QSPT was shown to reduce the impact time error down to fractions of a second under heavy disturbances. The improved time-to-go calculation along with an improved preflight estimate of the final impact time and robustness of QSPT provides considerable improvement over existing results.

LIST OF REFERENCES

[1] P.H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Education Series, 2000.

[2] C.F. Lin, *Modern Navigation, Guidance, and Control Processing*, Prentice Hall, 1991.

[3] P. Zarchan, *Tactical and Strategic Missile Guidance*, 4th ed. American Institute of Aeronautics and Astronautics, 2002.

[4] A.E. Bryson and Y.c. Ho, *Applied Optimal Control*, Hemisphere Publishing, 1975.

[5] C.K. Ryoo, H. Cho, M.J. Tahk, *Closed-Form Solutions of Optimal Guidance with Terminal Impact Angle Constraint*, Proceedings of the 2003 Conference on Control Applications, pages 504-509, Vol. 1, June 23-25, 2003.

[6] E.J. Ohlmeyer, *Control of Terminal Engagement Geometry using Generalized Vector Explicit Guidance*, Proceedings of the American Control Conference, Denver, Colorado, June 2003.

[7] C.K. Ryoo, H. Cho, M.J. Tahk, *Optimal Guidance Laws with Terminal Impact Angle Constraint*, Journal of Guidance, Control, and Dynamics, Vol. 28, No. 4, July-August 2005.

[8] J.I. Lee, I.S. Jeon, M.J. Tahk, *Guidance Law to Control Impact Time and Angle*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 43, No. 1, January 2007.

[9] T.L. Song, S.J. Shin, *Time-Optimal Impact Angle Control for Vertical Plane Engagements*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No. 2, April 1999.

[10] J.I. Lee, I.S. Jeon, M.J. Tahk, *Impact-Time-Control Guidance Law for Anti-Ship Missiles*, IEEE Transactions on Aerospace and Electronic Systems, Vol. 14, No. 2, March, 2006.

[11] C.A. Phillips, *Guidance Algorithm for Range Maximization and Time-of-Flight Control of a Guided Projectile*, Naval Surface Warefare Center, Dahlgren, Virginia. DOI:10.2514/1.31327

[12] C.K. Ryoo, M.J. Tahk, H. Cho, *Practical Time-to-go Estimation Methods for Optimal Guidance*, American Institute of Aeronautics and Astronautics, AIAA-99-4143.

[13] D.G. Hull, R.E. Mack, *Prediction of Time-to-go for a Homing Missile using Bang-Bang Control*, American Institute of Aeronautics and Astronautics, 88-4065-CP.

[14] I.H. Whang, W.S. Ra, *Time-to-go Estimation Filter for Anti-Ship Missile Applications*, SICE Annual Conference, The University Electro-Communications, Japan, August 2008.

[15] L.A. Weitz, J.E. Hurtado, *A time-to-go Control Law for Spacing Vehicles at a Point*, AIAA Guidance, Navigation, and Control Conference, Portland Oregon, August 2011.

[16] V.C. Lam, *AIAA Guidance, Navigation, and Control Conference*, AIAA Guidance, Navigation, and Control Conference, San Francisco, California , August 2005.

[17] T.H. Kim, C.H. Lee, M.J. Tahk, *Time-to-go Polynomial Guidance Laws with Terminal Impact Angle/Acceleration Constraints*, IFAC World Congress, Milano, Italy, August 28 - September 2, 2011

[18] D.G. Hull, J.J. Radke, *Time-to-go Prediction for a Homing Missile Based on Minimum-Time Intercepts*, American Institute of Aeronautics and Astronautics, 88-4064-CP

[19] H. Cho, C.K. Ryoo, M.J. Tahk, *Closed-Form Optimal Guidance Law for Missiles of Time-Varying Velocity*, Journal of Guidance, Control, and Dynamics, Vol. 19, No. 5, September - October 1996

[20] H. Cho, C.K. Ryoo, M.J. Tahk, *Implementation of Optimal Guidance Laws Using Predicted Missile Velocity Profiles*, Journal of Guidance, Control, and Dynamics, Vol. 22, No. 4, July - August 1996

[21] Y. Baba, T. Takehira, H. Takano, *New Guidance Law for a Missile with Varying Velocity*, AIAA-94-3565-CP

[22] J.I. Lee, I.S. Jeon, M.J. Tahk, *Guidance Law Using Augmented Trajectory-Reshaping Command for Salvo Attack of Multiple Missiles*, Copyright 2002, USTARTH

[23] S. Arita, S. Ueno, *Improvement of Guidance Law for Simultaneous Attack*, SICE Annual Conference 2011, September 2011, Waseda University, Tokyo, Japan.

[24] J. Wu, P. Ma, j. Ji, *Research on Cooperative Control Method of Saturation Attack*, Proceedings of the IEEE International Conference on Automation and Logistics, August 2007, Jinan, China

[25] M.G. Snyder, *A New Impact Time Control Guidance Law for Precise Time-on-Target Missile Strike*, Aerospace Sciences Conference, Kissimmee, Florida, January, 2015

[26] C.L. Lin, Y.P. Lin, K.M. Chen, *On the Design of Fuzzified Trajectory Shaping Guidance Law*, ISA Transactions, vol. 48, pp. 148-155, 2009

[27] E.J. Ohlmeyer, C.A. Phillips,*Generalized Vector Explicit Guidance*, Journal of Guidance, Control, and Dynamics, Vol. 29, No. 2, March-April 2006.