2013

New near-optimal feedback guidance algorithms for space missions

Matt Hawkins
Iowa State University

Recommended Citation

Hawkins, Matt, "New near-optimal feedback guidance algorithms for space missions" (2013). *Graduate Theses and Dissertations*. 13103.
https://lib.dr.iastate.edu/etd/13103

New near-optimal feedback guidance algorithms for space missions

by

Matthew Jay Hawkins

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:

Bong Wie, Major Professor

Ran Dai

Hui Hu

Ping Lu

Peter Sherman

Umesh Vaidya

Iowa State University

Ames, Iowa

2013

DEDICATION

I would like to dedicate this dissertation to my parents, and to all of my family and friends who have helped me along the way. I couldn't have done it without your support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. Introduction

This dissertation describes several different spacecraft guidance algorithms, with applications including asteroid intercept and rendezvous, planetary landing, and orbital transfer. The research described here was conducted at the Asteroid Deflection Research Center (ADRC) at Iowa State University.

1.1 Spacecraft Guidance

The fundamental goal of a spacecraft guidance law is to take a spacecraft from some known current state (position and velocity) to some desired final state. The guidance law uses some measured state information, such as the spacecraft's current position or the angle between the target position and the spacecraft's line of sight, as input to determine how to fire the spacecraft's thrusters to ultimately reach the desired final state. Different guidance laws use different inputs, and can have different requirements on the final desired state. For instance, for soft landing the final velocity is zero, or for some missions a particular total mission time may be specified.

Classical guidance laws were the first to be studied, starting with proportional navigation (PN). [1] Also called parallel navigation, PN guidance is based on physical intuition, and in fact animals are observed to use it to pursue prey. [2] Extensions of PN guidance are possible when more information about the target's acceleration is known. More sophisticated predictive guidance is also possible for spacecraft in orbit, either based on solving Lambert's problem, a well-known orbital mechanics problem, or based on finding a state transition matrix that relates the current relative position of the spacecraft to the estimated future relative position. [3]

Further refinement is possible with optimal guidance laws. Such laws are "optimal" in that they can be shown to minimize some quantity of interest, such as control effort used (called the performance index). The classical guidance laws are special cases of optimal guidance laws, with some approximations or simplifications.

Chapter 2 provides a review of the guidance literature, and describes several guidance laws as applied to asteroid intercept and rendezvous. This chapter summarizes guidance law research done at the ADRC. The so-called “ZEM/ZEV” (Zero-Effort Miss/Zero-Effort Velocity) feedback guidance algorithm, first described in Ref. [4], is introduced. This guidance law is studied in detail in the next two chapters.

1.2 Optimal Guidance

Before describing ZEM/ZEV guidance, a brief discussion of optimal guidance is in order. The optimal laws described so far are “closed-loop” optimal feedback laws. This means that the guidance laws generate thruster commands based only on current state information, and possibly some knowledge of the system dynamics. This contrasts with “open-loop” optimal guidance, which uses a computer to find the best control history for the entire mission.

Closed-loop feedback has advantages including computational simplicity and robustness in the presence of disturbances. Disadvantages include that closed-loop laws will in general not minimize the performance index, and that some constraints, such as remaining above the surface when landing, are difficult to enforce. Open-loop laws will find the best performance index, but the solution must be tracked somehow, and may need to be recomputed. A closed-loop guidance law that achieves similar performance to open-loop is thus desirable.

1.3 ZEM/ZEV Feedback Guidance

The ZEM/ZEV algorithm was first described in [4]. It is a further generalization of the optimal guidance laws, which is conceptually simple while maintaining the advantages of feedback guidance. The law performs well for many applications, and with some refinements can achieve near-optimal performance for many missions.

The Zero-Effort Miss distance, or ZEM, is defined as the amount by which the spacecraft will miss the target if no further accelerations are applied. Driving this quantity to zero means the spacecraft will hit the target. The Zero-Effort Velocity error, or ZEV, is a similar measure for the velocity. If there is a required final velocity (often zero for soft landing or rendezvous), driving the ZEV to zero means a successful mission. With both the ZEM and ZEV defined in terms of the desired final state, it is reasonable to use them as input to the guidance law.

Chapter 3 describes the ZEM/ZEV feedback guidance algorithm, then shows its use in a variety of applications. The applications shown are missile intercept, asteroid orbital operations, orbit transfer, and orbit raising. The ballistic missile intercept example considers only ZEM guidance, and is included to put the new guidance law in terms of a familiar, well-studied problem. The orbital operations example includes a discussion of finding the optimal mission time, as well as an extension of the ZEM/ZEV law to command landing somewhere on a given surface, rather than at a point. The orbit transfer example shows ZEM/ZEV guidance for a highly nonlinear system. The orbit raising example shows an interesting case where the ZEV is used, but not the ZEM. This example also introduces the idea of using a set of waypoints as intermediate steps to improve performance.

Chapter 4 describes using the ZEM/ZEV feedback guidance algorithm for a pinpoint Mars landing. With constraints on the problem, including maximum engine thrust and a constraint to maintain positive altitude, the ZEM/ZEV algorithm is modified. Modifications discussed include adaptively changing the mission time to meet constraints, and using optimization software to find one waypoint for the spacecraft to reach, then using the closed-loop ZEM/ZEV algorithm to complete the landing. The waypoint-optimized ZEM/ZEV algorithm is shown to achieve near-optimal performance while maintaining the advantages of feedback control.

CHAPTER 2. Spacecraft Guidance Algorithms for Asteroid Intercept and Rendezvous Missions

This chapter presents a comprehensive review of spacecraft guidance algorithms for asteroid intercept and rendezvous missions. Classical proportional navigation (PN) guidance is reviewed first, followed by pulsed PN guidance, augmented PN guidance, predictive feedback guidance, Lambert guidance, and other guidance laws based on orbit perturbation theory. Optimal feedback guidance laws satisfying various terminal constraints are also discussed. Finally, the zero-effort-velocity (ZEV) error, analogous to the well-known zero-effort-miss (ZEM) distance, is introduced, leading to a generalized ZEM/ZEV guidance law. These various feedback guidance laws can be easily applied to real asteroid intercept and rendezvous missions. However, differing mission requirements and spacecraft capabilities will require continued research on terminal-phase guidance laws.

2.1 Introduction

Acknowledgement of the threat to planet Earth from the impact of an asteroid or comet has led to increased interest in studying asteroid intercept and rendezvous missions. In addition to responding to a threatening asteroid, scientific missions to asteroids and other small bodies are of interest. NASA's Deep Space 1 and Deep Impact missions, and JAXA's Hayabusa mission, are examples of flyby, intercept/impact, and rendezvous, respectively. The differing mission requirements and spacecraft capabilities will require continued study on terminal-phase guidance laws.

This chapter will review a number of different feedback guidance laws, starting with simple laws to enable intercept, then going through more sophisticated laws, including laws with specified terminal conditions, and optimal feedback guidance laws. Simulation results are shown, proving the practical effectiveness of such feedback guidance laws.

Proportional navigation (PN) guidance is one of the earliest known feedback guidance laws. Zarchan [1] describes PN guidance, as well as a method of augmenting it when acceleration characteristics of the target are known or can be assumed. These feedback guidance laws can be implemented with on-off thrusters using a simple Schmitt trigger or other pulse-modulation devices as described in Wie [5]. Zarchan [1] also describes predictive guidance for targets whose primary acceleration is due to a gravitational field. This is based on Lambert's theorem. Linearized predictive guidance is possible, based on the state-error transition matrix described by Vallado [6].

Gil-Fernández et al. [7, 8] describe various terminal-phase guidance laws of a kinetic impactor as applied to the European Space Agencys Don Quijote mission (now canceled). This work was continued in more detail by Hawkins et al. [9].

Much work has been done on the problem of commanding intercept at a specified impact angle. Some of the earliest work led to guidance laws with strict limits on initial conditions [10]. Since then, a number of different laws with different advantages have been proposed. Various guidance laws exist that do not require the time-to-go [11] that allow for significant target maneuvers [12] and that can be used during hypersonic flight [13]. Linear-quadratic control laws have been derived in [14], and guidance laws based on classical proportional navigation are described in [15]. Guidance laws have been derived to follow a circular path to the target [16] and to establish the desired end-of-mission geometry early on [17]. Hawkins and Wie [18] investigated modified PN guidance laws for asteroid intercept with terminal velocity directional constraints.

Bryson and Ho [19] discussed optimal feedback control laws for a simple rendezvous problem, considering both free terminal velocity and constrained-terminal velocity. They also discussed the relationship between optimal feedback control and proportional navigation guidance. Battin [3] also discussed an optimal terminal-state vector control for the orbit control problem, directly compensating for the known disturbing gravitational acceleration. D'Souza [20] further examined an optimal control algorithm in a uniform gravitational field, and developed a computational method to determine the optimal time-to-go.

Guo et al. [21] found three different optimal control laws for certain constraints on terminal conditions. All three laws require the terminal position to be specified, while the terminal velocity can be fully specified, free, or constrained along a particular direction. Hawkins et al. [22] compared these guidance laws with PN-based guidance laws.

Ebrahimi et al. [4] proposed a robust optimal sliding mode guidance law for an exoatmospheric interceptor, using fixed-interval propulsive maneuvers. In this paper, gravity was considered to be an explicit function of time. One major contribution of Ebrahimi et al. was the new concept of the zero-effort-velocity (ZEV) error, analogous to the well-known zero-effort-miss (ZEM) distance. The ZEV is the velocity error at the end of the mission if no further control accelerations are imparted. Furfaro et al. [23] later employed the ZEM/ZEV concept to construct two classes of non-linear guidance algorithms for a lunar precision landing mission.

Guo et al. [24, 25] showed that in a uniform gravitational field, the ZEM/ZEV logic is basically a generalized form of various well-known optimal feedback guidance solutions such as intercept or rendezvous, terminal guidance, and planetary landing. A Mars landing example, originally described by Açıkmeşe and Ploen [26] is investigated in [24, 25]. In some cases ZEM/ZEV guidance can be improved by introducing a number of waypoints. Details are given in [24, 25] on how to practically compute such waypoints in real time. Finally, the ZEM/ZEV concept was also generalized to other feedback guidance problems in [24, 25].

2.2 Mathematical modeling

The target asteroid can be modeled as a point mass in a standard heliocentric Keplerian orbit, as follows:

$$\begin{aligned} \dot{\mathbf{r}}_T &= \mathbf{v}_T \\ \dot{\mathbf{v}}_T &= \mathbf{g} \end{aligned} \tag{2.1}$$

where $\mathbf{r}_T$ and $\mathbf{v}_T$ are the position of the velocity vectors of the target and $\mathbf{g}$ is the gravitational acceleration due to the sun, expressed as

$$\mathbf{g} = -\frac{\mu_\odot \mathbf{r}}{r^3} \tag{2.2}$$

where $\mu_\odot$ is the solar gravitational parameter. Similarly, the motion of the spacecraft is described by

$$\begin{aligned} \dot{\mathbf{r}}_S &= \mathbf{v}_S \\ \dot{\mathbf{v}}_S &= \mathbf{g} + \mathbf{a} \end{aligned} \tag{2.3}$$

where $\mathbf{r}_S$ and $\mathbf{v}_S$ are the position of the velocity vectors of the spacecraft and $\mathbf{g}$ is again the gravitational acceleration due to the sun, and $\mathbf{a}$ is the control acceleration provided by the control thrusters. In this dissertation, a boldfaced symbol indicates a column matrix of a physical vector expressed in a chosen inertial reference frame.

In general, we have $\mathbf{g} = \mathbf{g}\left(\mathbf{r}, t\right)$. For some guidance problems the gravitational acceleration can be considered constant or negligible, but for asteroid terminal guidance missions, the gravitational acceleration must be considered a nonlinear function of position. There are some other disturbing accelerations that act on the spacecraft, such as radiation pressure and the gravitational acceleration due to the asteroid. However, intercept and rendezvous missions to small asteroids can neglect these.

The relative position of the spacecraft with respect to the target is then described by

$$\mathbf{r} = \mathbf{r}_S - \mathbf{r}_T \tag{2.4}$$

The equation of motion of the spacecraft with respect to the target becomes

$$\ddot{\mathbf{r}} = \mathbf{g} + \mathbf{a} \tag{2.5}$$

where $\mathbf{g}$ represents the sum of apparent gravitational accelerations on the target, as follows:

$$\begin{aligned} \mathbf{g} &= -\frac{\mu_\odot}{r_S^3}\mathbf{r}_S + \frac{\mu_\odot}{\mathbf{r}_T^3}\mathbf{r}_T \\ &= -\frac{\mu_\odot}{\left|\mathbf{r}_T + \mathbf{r}\right|^3}\left(\mathbf{r}_T + \mathbf{r}\right) + \frac{\mu_\odot}{r_T^3}\mathbf{r}_T \\ &\cong -\frac{\mu_\odot}{r_T^3}\mathbf{r} + \frac{3\mu_\odot}{r_T^5}\mathbf{r}_T\left(\mathbf{r}_T \cdot \mathbf{r}\right) \quad \text{for } r \ll r_T \end{aligned} \tag{2.6}$$

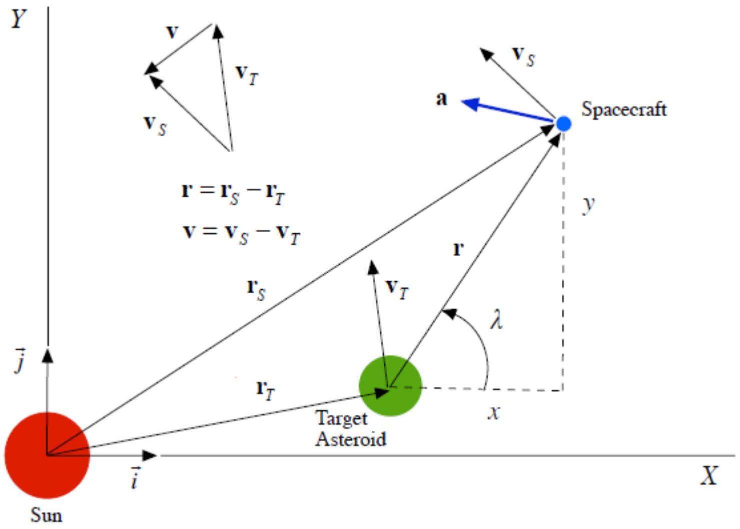


Figure 2.1 Coordinate system definition.

From Figure 2.1 it can be seen that

$$\lambda = \tan^{-1} \frac{y}{x} \tag{2.7}$$

where λ is the line-of-sight (LOS) angle and (x, y) are the components of the relative position vector along the inertial (X, Y) coordinates. Differentiating this with respect to time gives

$$\dot{\lambda} = \frac{x\dot{y} - y\dot{x}}{r^2} \tag{2.8}$$

where $\dot{\lambda}$ is the LOS rate and $r = \sqrt{x^2 + y^2}$. The rate of change of the distance between the target and the spacecraft is the closing velocity, found by differentiating r with respect to time as

$$V_c = -\dot{r} = \frac{-\left(x\dot{y} + y\dot{x}\right)}{r} \tag{2.9}$$

2.3 Guidance Laws

2.3.1 PN-based Feedback Guidance Laws

2.3.1.1 Classical Proportional Navigation (PN) Guidance

The first guidance law considered is the so-called proportional navigation (PN) guidance. The PN guidance attempts to drive the LOS rate to zero by applying accelerations perpendicular to the LOS direction. The PN guidance law is expressed as

$$a = nV_c\dot{\lambda} \tag{2.10}$$

where a is the acceleration command and n is the effective navigation ratio, a designer-tunable parameter. [1] The navigation ratio is typically chosen between 3 and 5. The optimal value, which will be derived and discussed in a later section, is 3. Larger values are chosen to provide more robustness against disturbances and errors. This can be seen by inspecting Eq. 2.10. Inaccurate estimates of the closing velocity or the LOS rate are equivalent to changing the navigation ratio and using accurate closing velocity and LOS rate information. Large navigation ratios will command unnecessarily large accelerations, while small navigation ratios risk commanding too little acceleration and missing the target. Thus a larger navigation ratio ensures that measurement errors will not make the accelerations too small to achieve impact.

Because the acceleration commands are always perpendicular to the LOS, the PN law gives a scalar value. The PN guidance acceleration command is then expressed in the inertial reference frame as

$$\mathbf{a} = nV_c\dot{\lambda}\begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{2.11}$$

The PN guidance law for steering the interceptor is also called constant-bearing guidance, as it steers the interceptor in such a way that the LOS does not rotate. An interceptor using PN guidance, on a perfect collision course, will maintain a constant bearing (i.e. $\dot{\lambda}$ is zero). When the interceptor is not on a collision course, the trajectory is not truly constant-bearing for $n < \infty$. As the effective navigation ratio becomes very large, the LOS rate approaches zero faster, at the expense of more commanded acceleration.

The PN guidance law does not require the target or interceptor velocities to be constant, nor does it require the external accelerations to be zero. For small deviations from constant velocity and small external accelerations, the PN guidance law will still achieve intercept in a feedback fashion. For the asteroid intercept scenario, the velocities are approximately constant, and the external acceleration is due almost entirely to the sun, and can be accounted for as described below.

2.3.1.2 Augmented PN Guidance

The basic PN guidance law can overcome target accelerations in a feedback fashion. As can be seen from the equations for LOS rate and closing velocity V_c, Eqs. 2.8 and 2.9, the PN guidance law uses only the position and velocity of the target, and is unable to take into account target accelerations (if they exist). A guidance law which incorporates terms to account for the target's acceleration should be able to perform better than the basic PN guidance law. Since the primary target accelerations are from the sun's gravity, the target's future accelerations are known. An augmented proportional navigation guidance (APNG) law will now be discussed. As with PNG, an easily tractable derivation will be given first, and optimality will be considered in a later section.

From Figure 2.1, with a small-angle approximation, we have

$$r \cong x \Rightarrow \lambda = \tan^{-1}\frac{y}{x} \cong \frac{y}{r} \tag{2.12}$$

where y is the spacecraft-target distance perpendicular to the reference line (X-axis). Define the mission time-to-go as

$$t_{go} = t_f - t \tag{2.13}$$

For a successful intercept mission, the separation at the end of the flight is zero, or $r\left(t_f\right) = 0$. Integrating Eq. 2.9 gives

$$r = V_c\left(t_f - t\right) = V_c t_{go} \tag{2.14}$$

Substituting Eq. 2.14 into Eq. 2.12 gives

$$\lambda \cong \frac{y}{V_c t_{go}} \tag{2.15}$$

Differentiating this expression gives

$$\dot{\lambda} \cong \frac{y + \dot{y}t_{go}}{V_c t_{go}^2} \tag{2.16}$$

Using this expression in the PNG law gives

$$a = nV_c\dot{\lambda} \cong \frac{n\left(y + \dot{y}t_{go}\right)}{t_{go}^2} \tag{2.17}$$

For PN guidance, we define the Zero-Effort-Miss (ZEM) distance as the separation between the target and the interceptor at the end of the flight, absent any further control accelerations. With no accelerations, the interceptor and target will continue on straight-line trajectories. The components of the ZEM can thus be given as

$$\begin{aligned} ZEM_x &= x + \dot{x}t_{go} \\ ZEM_y &= y + \dot{y}t_{go} \end{aligned} \tag{2.18}$$

The ZEM is a simplified prediction of future target-interceptor separation. PN guidance commands are always applied perpendicular to the LOS, therefore only the component of the ZEM that is perpendicular to the LOS can be accounted for in the augmented guidance law. From trigonometry, this component is

$$ZEM_{\perp} = -ZEM_x \sin\lambda + ZEM_y \cos\lambda \tag{2.19}$$

The PNG law, Eq. 2.10, is now rewritten as

$$a = n\frac{ZEM_{\perp}}{t_{go}^2} \tag{2.20}$$

A constant acceleration can be added to the ZEM term. The linearized ZEM equation is now

$$ZEM_{\perp} = y + \dot{y}t_{go} + \frac{1}{2}a_T t_{go}^2 \tag{2.21}$$

where a_T is the apparent target acceleration as seen by the spacecraft. Substituting this into Eq. 2.20, and using Eq. 2.17, gives

$$a = n\frac{y + \dot{y}t_{go}^2 + 0.5a_T t_{go}^2}{t_{go}^2} = nV_c\dot{\lambda} + \frac{n}{2}a_T \tag{2.22}$$

In general, for the asteroid terminal intercept scenario, the sun's gravity is the primary disturbing force that needs to be accounted for. The above augmented proportional navigation guidance law can easily incorporate the effect of gravity. The APNG law issues commands perpendicular to the LOS, and from the spacecraft's point of view. Thus the acceleration term needed is the relative solar acceleration perpendicular to the LOS. To begin, the components of the gravity term for the target and the spacecraft are

$$\begin{aligned} g_{T_x} &= \frac{-\mu_{\odot} r_{T_x}}{r_T^3} \\ g_{T_y} &= \frac{-\mu_{\odot} r_{T_y}}{r_T^3} \\ g_{S_x} &= \frac{-\mu_{\odot} r_{S_x}}{r_S^3} \\ g_{S_y} &= \frac{-\mu_{\odot} r_{S_y}}{r_S^3} \end{aligned} \tag{2.23}$$

The components perpendicular to the LOS are

$$\begin{aligned} g_{T_\perp} &= -g_{T_x} \sin\lambda + g_{T_y} \cos\lambda \\ g_{S_\perp} &= -g_{S_x} \sin\lambda + g_{S_y} \cos\lambda \end{aligned} \tag{2.24}$$

The target's apparent acceleration perpendicular to the LOS, as seen by the spacecraft, is

$$g_\perp = g_{T_\perp} - g_{S_\perp} \tag{2.25}$$

Substituting this equation into the APNG law, Eq. 2.22, gives

$$a = nV_c\dot{\lambda} + \frac{n}{2} g_\perp \tag{2.26}$$

In the inertial reference frame, the APNG law is expressed as

$$\mathbf{a} = nV_c\dot{\lambda}\begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} + \frac{n}{2} g_\perp \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{2.27}$$

2.3.1.3 Pulsed Guidance

For simple asteroid intercept, the terminal velocity is not specified, and is assumed to be the closing velocity for PNG and APNG. The PNG and APNG laws assume that continuously variable thrust is available. For thrusters with no throttling ability, a different approach to guidance laws is needed. Two approaches to formulating guidance laws for fixed-thrust-level (on-off) guidance laws are PN-based guidance laws and predictive guidance laws.

The PNG law continuously generates acceleration commands to achieve intercept. Due to its feedback nature, PNG will continue to generate guidance commands until intercept is achieved. A special case of PNG occurs when the interceptor is on a direct collision course. When this is true, the guidance commands will be zero. When using PNG logic, then, an acceleration command of zero means that the interceptor is instantaneously on a direct collision course. This fact can be exploited to use PNG logic for constant-thrust engines.

Pulsed PNG (PPNG) logic computes the required acceleration commands from PNG, but applies them in continuous-thrust pulses. PPNG will "overshoot" the amount of correction specified by PNG, until the PNG command is zero. At that point, the interceptor is instantaneously on a collision course, and the engines are turned off. If there were no external accelerations or disturbances, the interceptor would continue on an interception course. Because of the acceleration due to the sun, this will not be the case, and a further engine firing will be required later as the interceptor "drifts" further and further from the straight-line collision path.

Two approaches to determining when to fire engines are threshold methods and timed methods. Both methods will be described, as well as advantages and disadvantages associated with each.

The threshold method can employ the so-called Schmitt trigger or other pulse-modulation scheme. [5] Using a Schmitt trigger, acceleration commands are calculated by the PN guidance law as before. The trigger commands the divert thrusters to turn on once the commanded acceleration exceeds a certain magnitude, chosen by the designer, and off when the commanded acceleration reaches a designer-chosen cutoff. With traditional PN guidance the LOS rate must reach zero for a successful intercept. Therefore the second cutoff is typically selected as zero. The trigger control logic for pulsed proportional navigation guidance (PPNG) is shown in Figure 2.2. The Schmitt trigger can also be used for augmented PN guidance, giving an augmented pulsed proportional navigation guidance (APPNG).

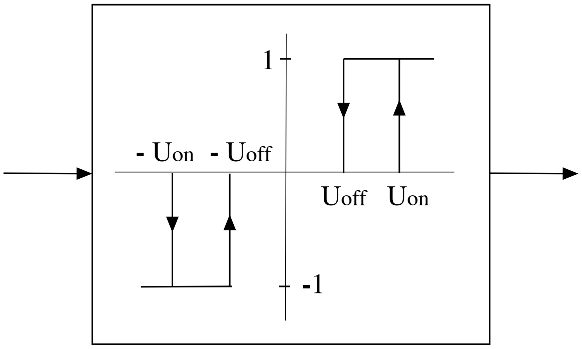


Figure 2.2 Scmitt trigger logic.

The timed method is similar to the threshold method in that the PNG commands are still calculated, and applied with constant thrust. As the name suggests, the difference is that the timed method uses predetermined firing times to turn on the thrusters. The designer must choose the firing schedule. Typically at least three firings are required. An early firing, near or at the beginning of the terminal mission phase, is used to overcome most of the orbit injection errors. The final firing comes shortly before impact, with enough lead time to allow the thrust command to complete (e.g. the commanded acceleration reaches zero), but close enough to impact that only minimal further errors accumulate. Additional intermediate firings provide robustness. As with the Schmitt trigger, the timed method turns off thrusters when the commanded thrust level is zero.

The advantage of these methods is the ability to use constant-thrust on-off engines. The Schmitt trigger has the same feedback advantages as PNG, in that it will issue commands when the spacecraft is not on an intercept course. A disadvantage of the Schmitt trigger is that the designer must select the magnitude to turn on the thrusters. Too small a magnitude risks excessive on-off cycles (chatter) for the engines, while too large a magnitude risks missing the target by failing to issue commands at all. An advantage of the timed method is that the total number of on-off cycles is known in advance and can be kept small. A disadvantage is that the timed method might not issue commands even when the calculated PNG commands are large.

The pulsed PN guidance law described in this section was applied to an asteroid intercept problem in [9]. A fictitious 300-m asteroid is assumed to be in a circular orbit with a radius of one astronomical unit. An interceptor with closing velocity of 10.4 km is displaced 350

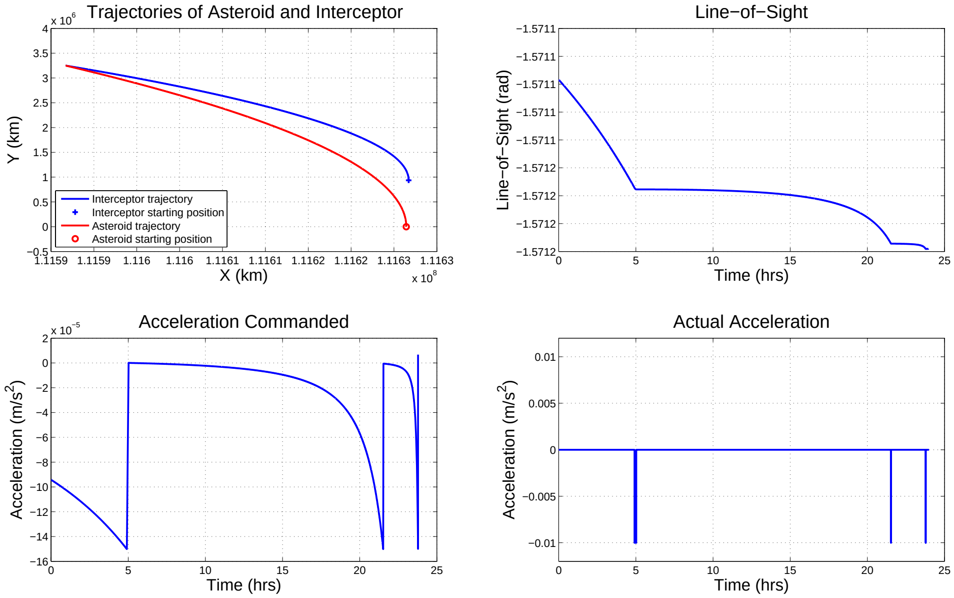


Figure 2.3 Trajectories, line-of-sight angle, commanded acceleration, and applied acceleration of the pulsed PN guidance law applied to an asteroid intercept problem [9].

kilometers out radially. It is displaced 24 hours travel time ahead of the target in the tangential direction. In the absence of guidance commands, the assumed initial conditions will result in a miss distance of 88.29 km. A 1000-kg interceptor with two 10-N divert thrusters is assumed. Simulation results summarized in Figures 2.3 and 2.4 indicate that the pulsed PN guidance system performs well. More detailed discussions of this asteroid intercept example problem can be found in [9].

2.3.2 Predictive Feedback Guidance Laws

A different class of guidance laws, which also use on-off pulses, are the predictive guidance schemes. Two types of predictive guidance laws are Lambert guidance and time-varying state transition matrix (STM) guidance. Both types of predictive guidance laws will command a required velocity, $\mathbf{v}_{req}$. Subtracting the current velocity $\mathbf{v}$ from this gives the velocity to be gained, or $\Delta\mathbf{v}$. The simplest way to generate acceleration commands is to align the thrust vector with the $\Delta\mathbf{v}$ vector. When the desired velocity is achieved, $\Delta\mathbf{v}$ is zero and the engine

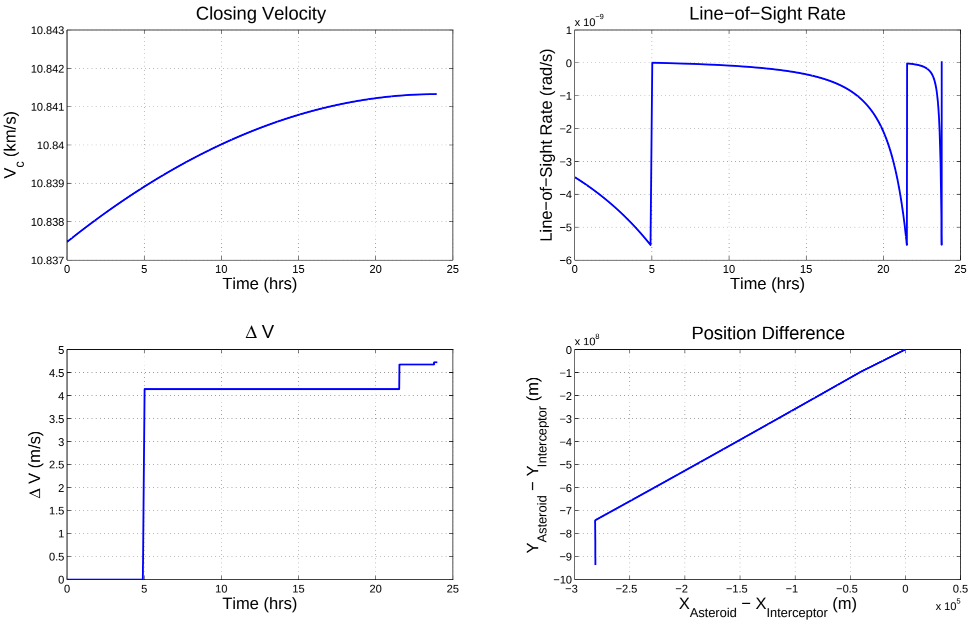


Figure 2.4 Closing velocity, line-of-site rate, Δv usage, and position error of the pulsed PN guidance law applied to an asteroid intercept problem [9].

is cut off. The velocity to be gained is

$$\Delta \mathbf{v} = \mathbf{v}_{req} - \mathbf{v} \tag{2.28}$$

For a given acceleration magnitude a, the direction of the thrust acceleration should be aligned with the velocity-to-be-gained vector, that is

$$\mathbf{a} = \frac{\Delta \mathbf{v}}{\|\Delta \mathbf{v}\|} \tag{2.29}$$

2.3.2.1 Lambert Guidance

The well-known Lambert's Theorem, a two-point boundary value problem (TPBVP), states that "the orbital transfer time depends only upon the semimajor axis, the sum of the distances of the initial and final points of the arc from the center of force, and the length of the chord joining these points." [3] Mathematically, Lambert's theorem is expressed as

$$t_2 - t_1 = f\left(r_1 + r_2,\, c,\, \bar{a}\right) \tag{2.30}$$

where $t_2 - t_1$ is the time of flight, $c = \|\mathbf{r}_2 - \mathbf{r}_1\|$, and $\bar{a}$ is the semi-major axis of the transfer orbit. The solution of Lambert's problem determines the required transfer orbit from position $\mathbf{r}_1$ to $\mathbf{r}_2$ in time $t_2 - t_1$. The current position of the spacecraft is known, and the desired time-of-flight can be calculated. Recall Eq. 2.14

$$r = V_c\left(t_f - t\right) = V_c t_{go}$$

The time-to-go is thus calculated as

$$t_{go} = \frac{r}{V_c} \tag{2.31}$$

The position of the target at the end of the flight time can be estimated by numerically integrating the target's current position and velocity over the time-to-go. Many different Lambert solvers have been developed that take the position vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, and the time of flight as inputs, and give parameters of the transfer orbit as output, including $\mathbf{v}_1$ and $\mathbf{v}_2$, the velocity of the transfer orbit at the initial and final times. The initial velocity is the required velocity from the Lambert solver routine, $\mathbf{v}_{Lambert}$. The required velocity is thus

$$\mathbf{v}_{req} = \mathbf{v}_{Lambert} = \mathbf{v}_1 \tag{2.32}$$

Comparing with Eq. 2.28 gives

$$\Delta\mathbf{v} = \mathbf{v}_{Lambert} - \mathbf{v} \tag{2.33}$$

The acceleration command is explicitly given as

$$\mathbf{a} = a\frac{\mathbf{v}_{Lambert} - \mathbf{v}}{\|\mathbf{v}_{Lambert} - \mathbf{v}\|} \tag{2.34}$$

The Lambert guidance routine requires a firing schedule to decide when to perform engine maneuvers. In principle, one Lambert guidance engine burn should suffice to achieve impact. In practice, multiple engine burns should be used to account for errors in calculating and applying velocity corrections. Similarly to PPNG with timed firings, a minimum of three burns should be used.

2.3.2.2 Time-varying State Transition Matrix (STM)

Impulsive guidance laws can also be formulated using the state transition matrix concept. In this approach, the target asteroid's orbit is considered to be a known, or reference, orbit. The interceptor's orbit is considered to be a perturbation from this reference orbit. The goal is then to issue guidance commands that will drive the position perturbation to zero.

Recall the dynamical equations of motion for the target asteroid

$$\begin{aligned} \dot{\mathbf{r}}_T &= \mathbf{v}_T \\ \dot{\mathbf{r}}_T &= \mathbf{v}_T \\ \mathbf{g}_T &= -\frac{\mu_{\odot}\mathbf{r}_T}{r_T^3} \end{aligned}$$

In this section, the standard notation of orbital perturbation theory will be followed. This notation differs from that used in the rest of the dissertation, but is used because it is more germane to the problem. The target state is the reference state, $\mathbf{x}^*$. The spacecraft state, $\mathbf{x}$, is the target state plus some deviation $\delta\mathbf{x}$, defined as

$$\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x} \tag{2.35}$$

where

$$\mathbf{x} \triangleq \begin{bmatrix} \mathbf{r}_S \\ \mathbf{v}_S \end{bmatrix}, \quad \mathbf{x}^* \triangleq \begin{bmatrix} \mathbf{r}_T \\ \mathbf{v}_T \end{bmatrix} \tag{2.36}$$

When components of the spacecraft state are needed, they will be given in orthogonal directions denoted with 1 and 2. The subscript S will be dropped for notational simplicity, but it is important to note that these are components of the spacecraft state, and not the spacecrafts relative state. The spacecraft state vector x is then defined as

$$\mathbf{x} = \begin{bmatrix} r_1 \\ r_2 \\ v_1 \\ v_2 \end{bmatrix} \tag{2.37}$$

Consider the target asteroid trajectory to be a known, or reference, trajectory. The equations of motion for the spacecraft trajectory are, in general, a function of both state and time as described by

$$\dot{\mathbf{x}}\left(t\right)=\mathbf{f}\left(\mathbf{x},t\right) \tag{2.38}$$

More explicitly, we have

$$\begin{bmatrix}\dot{r}_1\\ \dot{r}_2\\ \dot{v}_1\\ \dot{v}_2\end{bmatrix}=\begin{bmatrix}f_1\\ f_2\\ f_3\\ f_4\end{bmatrix}=\begin{bmatrix}v_1\\ v_2\\ -\frac{\mu_{\odot}r_1}{r^3}\\ -\frac{\mu_{\odot}r_2}{r^3}\end{bmatrix} \tag{2.39}$$

Substituting Eq. 2.35 into Eq. 2.38 gives

$$\dot{\mathbf{x}}=\mathbf{f}\left(\mathbf{x},t\right)=\mathbf{f}\left(\mathbf{x}^{*}+\delta\mathbf{x},t\right) \tag{2.40}$$

Equation 2.40 is nonlinear, so it can be expanded in a Taylor series about $\mathbf{x}^{*}$, as follows:

$$\dot{\mathbf{x}}\left(t\right)=\mathbf{f}\left(\mathbf{x}^{*},t\right)+\left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right]_{*}\delta\mathbf{x}\left(t\right)+\ldots \tag{2.41}$$

Substituting the time derivative of Eq. 2.35 into Eq. 2.41 gives

$$\dot{\mathbf{x}^{*}}\left(t\right)+\delta\dot{\mathbf{x}}\left(t\right)=\mathbf{f}\left(\mathbf{x}^{*},t\right)+\left[\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right]_{*}\delta\mathbf{x}\left(t\right)+\ldots \tag{2.42}$$

Recall that the target orbit is considered to be the known reference orbit. As such, the position and velocity are known for any given time. The function $\mathbf{f}$ and the state $\mathbf{x}$ can therefore be considered functions of time alone. The equation of motion for the state deviation is then simply expressed as

$$\delta\dot{\mathbf{x}}\left(t\right)=\mathbf{F}\left(t\right)\delta\mathbf{x}\left(t\right) \tag{2.43}$$

where $\mathbf{F}$ is the partial derivative term in Eq. 2.41, which is the Jacobian of the function $\mathbf{f}$ evaluated at $\mathbf{x}^{*}$ defined as

$$\mathbf{F}\left(t\right)=\left[\frac{\partial\mathbf{f}\left(t\right)}{\partial\mathbf{x}\left(t\right)}\right]_{*}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -\frac{\mu}{r^3}+\frac{3\mu r_1^2}{r^5} & \frac{3\mu r_1 r_2}{r^5} & 0 & 0\\ \frac{3\mu r_1 r_2}{r^5} & -\frac{\mu}{r^3}+\frac{3\mu r_2^2}{r^5} & 0 & 0\end{bmatrix} \tag{2.44}$$

where $\mu_\odot$ is replaced with μ for notational simplicity. Recall that the acceleration due to the sun's gravity, Eq. 2.2, is

$$\mathbf{g} = -\frac{\mu \mathbf{r}}{r^3}$$

The Jacobian of the gravitational force vector is often called the gravity-gradient matrix defined as follows:

$$\mathbf{G}(t) = \frac{\partial \mathbf{g}}{\partial \mathbf{r}} = \begin{bmatrix} -\frac{\mu}{r^3} + \frac{3\mu r_1^2}{r^5} & \frac{3\mu r_1 r_2}{r^5} \\ \frac{3\mu r_1 r_2}{r^5} & -\frac{\mu}{r^3} + \frac{3\mu r_2^2}{r^5} \end{bmatrix} \tag{2.45}$$

Observe that this corresponds to the bottom-left submatrix of $\mathbf{F}$, which can be written as a block matrix

$$\mathbf{F}(t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{G}(t) & \mathbf{0} \end{bmatrix} \tag{2.46}$$

where $\mathbf{I}$ and $\mathbf{0}$ are the identity matrix and the zero matrix of conformal (in this case 2x2) size.

The general solution to Eq. 2.43 above can be expressed as

$$\delta\mathbf{x}(t) = \Phi(t, t_0)\, \delta\mathbf{x}_0 \tag{2.47}$$

where $\delta\mathbf{x}_0 = \delta\mathbf{x}(t_0)$. Differentiating this gives

$$\delta\dot{\mathbf{x}}(t) = \dot{\Phi}(t, t_0)\, \delta\mathbf{x}_0 \tag{2.48}$$

Substituting Eq. 2.48 into Eq. 2.43, and using Eq. 2.47, we have

$$\dot{\Phi}(t, t_0)\, \delta\mathbf{x}_0 = \mathbf{F}(t)\, \Phi(t, t_0)\, \delta\mathbf{x}_0 \tag{2.49}$$

This must be true for any $\delta\mathbf{x}_0$, thus

$$\dot{\Phi}(t, t_0) = \mathbf{F}(t)\, \Phi(t, t_0) \tag{2.50}$$

with initial conditions given by

$$\Phi(t, t_0) = \mathbf{I} \tag{2.51}$$

Equation 2.50 can be numerically integrated, starting with the initial conditions given in Eq. 2.51, to find the current estimate of the state transition matrix. However, for use in generating a guidance law for asteroid intercept, it is desirable to avoid numerical integration.

As will be discussed in a later section, if a numerical integration is to be performed, it is better to simply integrate the equations of motion directly. It is desirable to have an analytical expression of Φ for use in guidance laws.

Consider the expression for the state error. Expanding this into a Taylor series gives

$$\delta\mathbf{x}\left(t\right) = \delta\mathbf{x}_0 + \delta\dot{\mathbf{x}}_0\Delta t + \frac{1}{2!}\delta\ddot{\mathbf{x}}_0\Delta t^2 + \ldots \tag{2.52}$$

Recalling Eq. 2.43, and taking its derivatives, gives

$$\begin{aligned} \delta\dot{\mathbf{x}}_0 &= \mathbf{F}\left(t\right)\delta\mathbf{x}_0 \\ \delta\ddot{\mathbf{x}}_0 &= \mathbf{F}\left(t\right)\delta\dot{\mathbf{x}}_0 = \mathbf{F}\left(t\right)^2\delta\mathbf{x}_0 \end{aligned} \tag{2.53}$$

Comparing Eqs. 2.47 and 2.52 gives

$$\delta\mathbf{x} = \left[\mathbf{I} + \mathbf{F}\left(t\right)\Delta t + \frac{1}{2}\mathbf{F}\left(t\right)^2\Delta t^2 + \ldots\right]\delta\mathbf{x}_0 = \Phi\delta\mathbf{x}_0 \tag{2.54}$$

Evaluating the expression in brackets gives the state-error transition matrix as

$$\Phi = \begin{bmatrix} 1 + \frac{3\mu\Delta t^2 r_1^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^5} & \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} & \Delta t & 0 \\ \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} & 1 + \frac{3\mu\Delta t^2 r_2^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^3} & 0 & \Delta t \\ \frac{3\mu\Delta t r_1^2}{r_0^5} - \frac{\mu\Delta t}{r_0^3} & \frac{3\mu\Delta t r_1 r_2}{r_0^5} & 1 + \frac{3\mu\Delta t^2 r_1^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^3} & \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} \\ \frac{3\mu\Delta t r_1 r_2}{r_0^5} & \frac{3\mu\Delta t r_2^2}{r_0^5} - \frac{\mu\Delta t}{r_0^3} & \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} & 1 + \frac{3\mu\Delta t^2 r_2^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^3} \end{bmatrix} \tag{2.55}$$

The state-error transition matrix can now be partitioned as

$$\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix} \tag{2.56}$$

where

$$\Phi_1 = \Phi_4 = \mathbf{I} + \frac{1}{2}\mathbf{G}\Delta t^2 = \begin{bmatrix} 1 + \frac{3\mu\Delta t^2 r_1^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^5} & \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} \\ \frac{3\mu\Delta t^2 r_1 r_2}{2r_0^5} & 1 + \frac{3\mu\Delta t^2 r_2^2}{2r_0^5} - \frac{\mu\Delta t^2}{2r_0^3} \end{bmatrix} \tag{2.57}$$

$$\Phi_2 = \mathbf{I}\Delta t = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{2.58}$$

$$\Phi_3 = \mathbf{G}\Delta t = \begin{bmatrix} \frac{3\mu\Delta t r_1^2}{r_0^5} - \frac{\mu\Delta t}{r_0^3} & \frac{3\mu\Delta t r_1 r_2}{r_0^5} \\ \frac{3\mu\Delta t r_1 r_2}{r_0^5} & \frac{3\mu\Delta t r_2^2}{r_0^5} - \frac{\mu\Delta t}{r_0^3} \end{bmatrix} \tag{2.59}$$

The simplest form of intercept guidance using the state transition matrix uses only the position information to generate a $\Delta\mathbf{v}$ command. The first-order approximation for the final miss vector, absent any further acceleration commands, is found as

$$\mathbf{r}\left(t_f\right) \cong \tilde{\mathbf{r}}_{tf} = \Phi_1\left(t\right)\mathbf{r}\left(t\right) + \Phi_2\left(t\right)\mathbf{v}\left(t\right) \tag{2.60}$$

The second term on the right-hand side is negligible for small changes in closing velocity, giving

$$\mathbf{r}\left(t_f\right) \cong \tilde{\mathbf{r}}_{tf} = \Phi_1\left(t\right)\mathbf{r}\left(t\right) \tag{2.61}$$

Consider driving the final relative position to zero. For linearized dynamics, there are no external accelerations, so we have

$$\begin{aligned} 0 &= \mathbf{r}\left(t_f\right) - \frac{\mathbf{r}\left(t_f\right)}{\Delta t} \\ &\cong \mathbf{r}\left(t_0\right) + \mathbf{v}\left(t_0\right) - \frac{\mathbf{r}\left(t_f\right)}{\Delta t}\Delta t \\ &= \mathbf{r}\left(t_0\right) + \mathbf{v}\left(t_0\right)\Delta t + \mathbf{v}_{STM}\Delta t \end{aligned} \tag{2.62}$$

where $\mathbf{v}_{STM}$ is seen to be the velocity to be gained to ensure impact.

For small changes in relative velocity, we have

$$\begin{aligned} \dot{r}\left(t_f\right) &\cong \dot{r}\left(t_0\right) \\ \Rightarrow \frac{\mathbf{r}\left(t_f\right)}{\Delta t} &\cong \hat{\mathbf{r}}\left(t_f\right)\dot{r}\left(t_0\right) \end{aligned} \tag{2.63}$$

The velocity change to be imparted becomes

$$\begin{aligned} \Delta\mathbf{v} &= \mathbf{v}_{STM} - \mathbf{v} \\ &= \hat{\mathbf{r}}\left(t_f\right)\dot{r}\left(t_0\right) - \mathbf{v} \end{aligned} \tag{2.64}$$

The acceleration command vector is given, similar to Lambert guidance, as

$$\mathbf{a} = a\frac{\mathbf{v}_{STM} - \mathbf{v}}{\left\|\mathbf{v}_{STM} - \mathbf{v}\right\|} \tag{2.65}$$

Using the gravity gradient matrix from Eq. 2.45, we can show that

$$\mathbf{r}\left(t_f\right) \approx \frac{\Delta t^2}{2}\mathbf{G}\left(t\right)\mathbf{r}\left(t\right) + \mathbf{r}\left(t\right) \tag{2.66}$$

2.3.2.3 Predictive Impulsive Guidance

For practical implementation, we adopt the following definitions

$$\begin{aligned} V_c\left(t\right) &= \dot{r}\left(t\right) \\ \mathbf{\Lambda}\left(t\right) &= \frac{\mathbf{r}\left(t\right)}{r\left(t\right)} \\ \mathbf{\Lambda}_c\left(t\right) &= \frac{\tilde{\mathbf{r}}\left(t_f\right)}{\tilde{r}\left(t_f\right)} \\ t_{go} &= \Delta t \end{aligned} \tag{2.67}$$

Substituting these into the guidance law in Eq. 2.64 gives the predictive impulsive guidance law

$$\Delta\mathbf{v} = V_c\mathbf{\Lambda}_c - \mathbf{v} \tag{2.68}$$

2.3.2.4 Kinematic Impulsive Guidance

In terms of the line-of-sight angle, we have

$$\begin{aligned} \mathbf{\Lambda}\left(t\right) &= \begin{bmatrix} \cos\lambda\left(t\right) \\ \sin\lambda\left(t\right) \end{bmatrix} \\ \dot{\mathbf{\Lambda}}\left(t\right) &= \begin{bmatrix} -\dot{\lambda}\sin\lambda\left(t\right) \\ \dot{\lambda}\cos\lambda\left(t\right) \end{bmatrix} \end{aligned} \tag{2.69}$$

The relative position can also be approximated as

$$\mathbf{r}\left(t\right) \approx V_c t_{go}\mathbf{\Lambda}\left(t\right) \tag{2.70}$$

and we obtain

$$\mathbf{r}\left(t_f\right) \approx \frac{V_c t_{go}^3}{2}\mathbf{G}\left(t\right)\mathbf{\Lambda}\left(t\right) + V_c t_{go}\mathbf{\Lambda}\left(t\right) \tag{2.71}$$

The relative velocity can be approximated as a component along the LOS and a component perpendicular to the LOS, described as

$$\mathbf{v}\left(t\right) \approx V_c t_{go}\dot{\mathbf{\Lambda}}\left(t\right) + V_c\mathbf{\Lambda}\left(t\right) \tag{2.72}$$

Substituting Eq. 2.72 into Eq. 2.68 results in the kinematic impulsive guidance law of the form

$$\Delta\mathbf{v} = V_c\left(\mathbf{\Lambda}_c - t_{go}\dot{\mathbf{\Lambda}} - \mathbf{\Lambda}\right) \tag{2.73}$$

2.4 Optimal Feedback Guidance Algorithms

For some applications it is desirable to specify terminal conditions on the interceptor. For intercept, the terminal position is by definition zero. The terminal velocity, though, may have direction or magnitude requirements, depending on the mission. Optimal feedback guidance laws can be used to achieve intercept, with the option of specifying the final velocity.

Three different optimal feedback guidance laws are considered for asteroid intercept and rendezvous. The various forms of proportional navigation and predictive guidance laws compute an estimated mission time-to-go based on relative position and velocity. This computed time-to-go is used as an input for the predictive laws, and is available as an output of the proportional navigation laws. In contrast, the optimal feedback guidance laws studied in this dissertation use a specified time-to-go as a mission parameter, and compute the acceleration commands needed to achieve intercept at this pre-determined time.

When the final impact velocity vector (both impact velocity and impact angle) is specified, the terminal velocity is constrained. This leads to the constrained-terminal-velocity guidance (CTVG) law. If the final velocity is free, the free-terminal-velocity guidance (FTVG) law results, as discussed in [24, 25]. When only the approach angle is commanded, the velocity vector component along the desired final direction is free, while the perpendicular components are constrained to be zero. A combination of FTVG along the impact direction and CTVG along the perpendicular directions allows pointing of the final velocity vector, referred to as intercept-angle-control guidance (IACG).

2.4.1 Constrained-Terminal-Velocity Guidance (CTVG)

Consider an optimal control problem for minimizing the integral of the acceleration squared, formulated as

$$J = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{a}^T \mathbf{a} \, \mathrm{d}t \tag{2.74}$$

subject to $\dot{\mathbf{r}} = \mathbf{v}$ and $\dot{\mathbf{v}} = \mathbf{g} + \mathbf{a}$ with the following boundary conditions

$$\begin{aligned} \mathbf{r}\left(t_0\right) = \mathbf{r}_0 \quad \mathbf{r}\left(t_f\right) = \mathbf{r}_f \\ \mathbf{v}\left(t_0\right) = \mathbf{v}_0 \quad \mathbf{v}\left(t_f\right) = \mathbf{v}_f \end{aligned} \tag{2.75}$$

The Hamiltonian function is given by

$$H = \frac{1}{2}\mathbf{a}^T\mathbf{a} + \mathbf{p}_r^T\mathbf{v} + \mathbf{p}_v^T\left(\mathbf{g} + \mathbf{a}\right) \tag{2.76}$$

where $\mathbf{p}_r$ and $\mathbf{p}_v$ are co-state vectors associated with the position and velocity vectors, respectively. In general, for a terminal-phase guidance problem gravity is a function of position and time, $\mathbf{g} = \mathbf{g}\left(\mathbf{r}, t\right)$. Using such a function will not allow a simple closed-form solution to the optimal control problem. For the class of terminal guidance problems considered here, the gravitational acceleration is approximately constant for the duration of the terminal phase. Therefore a constant gravitational acceleration is assumed to permit a closed-form solution.

The co-state equations and control equation imply that

$$\begin{align}
\dot{\mathbf{p}}_r &= -\frac{\partial H}{\partial \mathbf{r}} = 0 \tag{2.77}\\
\dot{\mathbf{p}}_v &= -\frac{\partial H}{\partial \mathbf{v}} = -\mathbf{p}_r \tag{2.78}\\
\frac{\partial H}{\partial \mathbf{a}} &= 0 \;\Rightarrow\; \mathbf{a} = -\mathbf{p}_v \tag{2.79}
\end{align}$$

For fixed terminal conditions, the co-states at t_f are non-zero. Define $t_{go} = t_f - t$ as the time-to-go before arrival at the terminal state, and let and $\mathbf{p}_r\left(t_f\right)$ and $\mathbf{p}_v\left(t_f\right)$ describe the values of $\mathbf{p}_r$ and $\mathbf{p}_v$ at t_f, respectively. Integrating the co-state equations yields

$$\mathbf{p}_r = \mathbf{p}_r\left(t_f\right) \tag{2.80}$$

$$\mathbf{p}_v = t_{go}\mathbf{p}_r\left(t_f\right) + \mathbf{p}_v\left(t_f\right) \tag{2.81}$$

Substituting Eq. 2.81 into Eq. 2.79 yields the optimal control solution as

$$\mathbf{a} = -t_{go}\mathbf{p}_r\left(t_f\right) - \mathbf{p}_v\left(t_f\right) \tag{2.82}$$

The states can thus be expressed as

$$\begin{align}
\mathbf{v} &= \frac{t_{go}^2}{2}\mathbf{p}_r\left(t_f\right) + t_{go}\mathbf{p}_v\left(t_f\right) - t_{go}\mathbf{g} + \mathbf{v}_f \tag{2.83}\\
\mathbf{r} &= -\frac{t_{go}^3}{6}\mathbf{p}_r\left(t_f\right) - \frac{t_{go}^2}{2}\mathbf{p}_v\left(t_f\right) + \frac{t_{go}^2}{2}\mathbf{g} - t_{go}\mathbf{v}_f + \mathbf{r}_f \tag{2.84}
\end{align}$$

Combining Eq. 2.83 and Eq. 2.84 leads to

$$\mathbf{p}_r\left(t_f\right) = \frac{6\left(\mathbf{v}+\mathbf{v}_f\right)}{t_{go}^2} + \frac{12\left(\mathbf{r}-\mathbf{r}_f\right)}{t_{go}^3} \tag{2.85}$$

$$\mathbf{p}_v\left(t_f\right) = -\frac{2\left(\mathbf{v}+2\mathbf{v}_f\right)}{t_{go}} - \frac{6\left(\mathbf{r}-\mathbf{r}_f\right)}{t_{go}^2} + \mathbf{g} \tag{2.86}$$

Finally, the optimal feedback control law with specified $\mathbf{r}_f$, $\mathbf{v}_f$, and t_{go}, the CTVG law, is obtained as

$$\mathbf{a} = \frac{6\left[\mathbf{r}_f - \left(\mathbf{r}+t_{go}\mathbf{v}\right)\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f-\mathbf{v}\right)}{t_{go}} - \mathbf{g} \tag{2.87}$$

or

$$\mathbf{a} = \frac{6\left[\mathbf{r}_f - \left(\mathbf{r}+t_{go}\mathbf{v}_f\right)\right]}{t_{go}^2} + \frac{4\left(\mathbf{v}_f-\mathbf{v}\right)}{t_{go}} - \mathbf{g} \tag{2.88}$$

2.4.2 Free-Terminal-Velocity Guidance (FTVG)

For the case when the terminal velocity is free, the boundary conditions for unconstrained final velocity give $\mathbf{p}_v\left(t_f\right) = 0$, thus Eq. 2.81, $\mathbf{p}_v$ becomes

$$\mathbf{p}_v = t_{go}\mathbf{p}_r\left(t_f\right) \tag{2.89}$$

The acceleration command is then given as

$$\mathbf{a} = -\mathbf{p}_v = -t_{go}\mathbf{p}_r\left(t_f\right) \tag{2.90}$$

Accordingly we have

$$\mathbf{v} = \frac{t_{go}^2}{2}\mathbf{p}_r\left(t_f\right) - t_{go}\mathbf{g} + \mathbf{v}_f \tag{2.91}$$

$$\mathbf{r} = -\frac{t_{go}^3}{6}\mathbf{p}_r\left(t_f\right) + \frac{t_{go}^2}{2}\mathbf{g} - t_{go}\mathbf{v}_f + \mathbf{r}_f \tag{2.92}$$

Solving the above equations results in

$$\mathbf{p}_r\left(t_f\right) = -\frac{3}{t_{go}^3}\left(\mathbf{r}_f - \mathbf{r} - \mathbf{v}t_{go} - \frac{t_{go}^2}{2}\mathbf{g}\right) \tag{2.93}$$

The terminal velocity can be expressed in terms of time-to-go and system states after substituting Eq. 2.93 into Eq. 2.91, as

$$\mathbf{v}_f = \frac{3}{2t_{go}}\left(\mathbf{r}_f - \mathbf{r}\right) - \frac{1}{2}\mathbf{v} + \frac{1}{4}t_{go}\mathbf{g} \tag{2.94}$$

which is a function of time with given initial position and velocity vectors.

Substituting Eq. 2.93 into Eq. 2.90, the FTVG law is obtained as follows

$$\mathbf{a} = \frac{3}{t_{go}^2}\left(\mathbf{r}_f - \mathbf{r}\right) - \frac{3}{t_{go}}\mathbf{v} - \frac{3}{2}\mathbf{g} \tag{2.95}$$

2.4.3 Intercept-Angle-Control Guidance (IACG)

Both CTVG and FTVG command the final position. The terminal velocity vector can be commanded, as in CTVG, or free as in FTVG. Consider an orthogonal coordinate system with the first component ($\mathbf{e}_1$) in the direction of the desired terminal velocity, and the other two components ($\mathbf{e}_2$ and $\mathbf{e}_3$) perpendicular to this direction. The acceleration vector can be expressed as

$$\mathbf{a} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3 \tag{2.96}$$

The performance index then becomes simply

$$J = \frac{1}{2}\int_{t_0}^{t_f}\left(a_1^2 + a_2^2 + a_3^2\right)\,\mathrm{d}t \tag{2.97}$$

The position, velocity, and gravity vectors can also be expressed as

$$\begin{aligned} \mathbf{r} &= a_1\mathbf{e}_1 + r_2\mathbf{e}_2 + r_3\mathbf{e}_3 \\ \mathbf{v} &= a_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3 \\ \mathbf{g} &= a_1\mathbf{e}_1 + g_2\mathbf{e}_2 + g_3\mathbf{e}_3 \end{aligned} \tag{2.98}$$

In order to achieve impact along the $\mathbf{e}_1$-direction, it is required that v_1 is free, and v_2 and v_3 are both zero. The IACG algorithm combines the CTVG and FTVG algorithms as follows:

$$\begin{aligned} \mathbf{a} = &\left(\frac{3\left(r_{f1} - r_1\right)}{t_{go}^2} - \frac{3v_1}{t_{go}} - \frac{3g_1}{2}\right)\mathbf{e}_1 + \left(\frac{3\left(r_{f2} - r_2\right)}{t_{go}^2} - \frac{3v_2}{t_{go}} - \frac{3g_2}{2}\right)\mathbf{e}_2 \\ &+ \left(\frac{6\left(r_{f3} - r_3\right)}{t_{go}^2} - \frac{4v_3}{t_{go}} - 3g_3\right)\mathbf{e}_3 \end{aligned} \tag{2.99}$$

The algorithm can also be expressed in terms of vectors $\mathbf{r}$, $\mathbf{r}_f$, $\mathbf{v}$, and $\mathbf{g}$ as

$$\begin{aligned} \mathbf{a} = &\left(3\mathbf{e}_1\mathbf{e}_1^T + 6\mathbf{e}_2\mathbf{e}_2^T + 6\mathbf{e}_3\mathbf{e}_3^T\right)\frac{\mathbf{r}_f - \mathbf{r}}{t_{go}^2} - \left(3\mathbf{e}_1\mathbf{e}_1^T + 4\mathbf{e}_2\mathbf{e}_2^T + 3\mathbf{e}_3\mathbf{e}_3^T\right)\frac{\mathbf{v}}{t_{go}} \\ &- \left(\frac{3}{2}\mathbf{e}_1\mathbf{e}_1^T + 4\mathbf{e}_2\mathbf{e}_2^T + 4\mathbf{e}_3\mathbf{e}_3^T\right)\mathbf{g} \end{aligned} \tag{2.100}$$

It is important to note that the IACG guidance law does not impose a unique direction on the final velocity. Ultimately the velocity is only constrained to be parallel to the specified direction. The final velocity direction will depend on the initial conditions. If the spacecraft is initially moving toward the target in the $\mathbf{e}_1$ direction, that is $(r_{f1} - r_1)\, v_1 > 0$, then the intercept will be in the specified direction. Otherwise, when $(r_{f1} - r_1)\, v_1 \leq 0$, the spacecraft will intercept opposite the specified direction.

2.4.4 Relationship Between PNG and Optimal Feedback Guidance

As mentioned in the section on PNG, the optimal value for the navigation constant is 3. To show this, first consider a two-dimensional problem with

$$\begin{aligned} \mathbf{r}_f - \mathbf{r} &= [x\ y]^T \\ -\mathbf{v} &= [\dot{x}\ \dot{y}]^T \qquad (2.101) \end{aligned}$$

$$\lambda = \tan^{-1}\frac{y}{x} \qquad (2.102)$$

where λ is the LOS angle as before. Define $R = |\mathbf{r}_f - \mathbf{r}|$ as the distance from the spacecraft to the target along the LOS. The closing velocity and time-to-go are given by

$$V_c = -\dot{R} \qquad (2.103)$$

$$t_{go} = \frac{R}{V_c} \qquad (2.104)$$

The optimal FTVG algorithm thus becomes

$$\mathbf{a} = 3V_c\dot{\lambda}\begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} - \frac{3}{2}\mathbf{g} \qquad (2.105)$$

This is the augmented PNG logic, with an effective navigation ratio of 3.

Controlling the direction of the final velocity is equivalent to controlling the final impact angle, but not the velocity. [19] Since the PNG laws only command control perpendicular to the LOS, the velocity along the LOS is free. For the case with small LOS angles, it can be shown that IACG becomes PNG with impact angle control

$$a = 4V_c\dot{\lambda} + \frac{2V_c\,(\lambda - \lambda_f)}{t_{go}} - g \qquad (2.106)$$

where λ_f is the desired final impact angle.

2.4.5 Calculation of Time-To-Go

The basic PNG laws do not specify the time-to-go, and can estimate it based on current conditions. In contrast, the optimal feedback guidance laws are derived for a fixed flight time. The time-to-go appears in the acceleration commands. For some missions it may be desirable to specify the mission time. However, for asteroid intercept it is often not necessary to achieve impact or rendezvous at a particular time, and a difference of a few seconds or minutes is not significant to the overall mission.

Consider, for example, an interceptor that is already on a collision course with the target. Proportional navigation will not issue any commands, and intercept will occur based on the interceptor's velocity relative to the target. If one of the optimal guidance laws is used with a different time-to-go, intercept will still occur, but the spacecraft will spend unnecessary fuel either speeding up or slowing down the spacecraft along the LOS direction. It is natural to ask, then, if there is an optimal choice for time-to-go.

For both the CTVG and FTVG laws, under certain conditions a local minimum for the performance index with respect to mission time is possible. This condition will be derived next. It is important to note that the optimal feedback control laws minimize the integral of acceleration squared, and not Δv or fuel use.

2.4.5.1 CTVG

As was the case when deriving the CTVG law, the gravitational acceleration is assumed to be constant to permit a closed form solution. The transversality condition is given by

$$\begin{aligned}\frac{\partial J^*}{\partial t_{go}} = \frac{\partial J^*}{\partial t_f} &= H = \text{constant on the optimal trajectory} \\ &= 0 \text{ when } t_f \text{ is free} \qquad (2.107)\end{aligned}$$

The Hamiltonian function of the form

$$H = -\frac{1}{2}\mathbf{a}^T\mathbf{a}\left(\mathbf{a} + 2\mathbf{g}\right) + \mathbf{p}_r^T\mathbf{v} \qquad (2.108)$$

becomes

$$H = \frac{1}{t_{go}^4}\left[t_{go}^4\left(\mathbf{g}^T\mathbf{g}+\Gamma\right) - 2t_{go}^2\left(\mathbf{v}^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}_f\right) + 12t_{go}\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{v}+\mathbf{v}_f\right) \right. \\ \left. - 18\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right)\right] \tag{2.109}$$

Equation 2.107 then gives

$$t_{go}^4\left(\mathbf{g}^T\mathbf{g}+\Gamma\right) - 2t_{go}^2\left(\mathbf{v}^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}_f\right) + 12t_{go}\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{v}+\mathbf{v}_f\right) - 18\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right) = 0 \tag{2.110}$$

which can be solved for the time-to-go. Note that the first term in the above equation contains the fourth power of the time-to-go, and the unknown parameter Γ. This first term is also the only term where the gravitational acceleration appears. By assuming that $\mathbf{g}$ is negligible, and setting Γ to zero, Eq. 2.110 simplifies to

$$t_{go}^2\left(\mathbf{v}^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}_f\right) - 6t_{go}\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{v}+\mathbf{v}_f\right) + 9\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right) = 0 \tag{2.111}$$

The time-to-go can be calculated as follows:

$$t_{go} = \begin{cases} \tau & B^2 - 4AC > 0 \text{ and } B < 0 \\ \text{no solution} & \text{otherwise} \end{cases} \tag{2.112}$$

where

$$\begin{aligned} \tau &= \frac{-B-\sqrt{B^2-4AC}}{2A} \\ A &= \left(\mathbf{v}^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}+\mathbf{v}_f^T\mathbf{v}_f\right) \geq 0 \\ B &= 6\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{v}+\mathbf{v}_f\right) \\ C &= 9\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right) \end{aligned}$$

and τ is the smaller positive solution of Eq. 2.111, leading to a local minimum of J^*. The larger solution corresponds to a local maximum of J^*. J^* decreases monotonically with respect to t_{go} for values beyond the larger solution. When there is no solution, the partial derivative of J^* is always negative, thus increasing t_{go} leads to decreasing J^*.

2.4.5.2 FTVG

A similar procedure can be used to compute the time-to-go for the FTVG law. The condition corresponding to Eq. 2.110 is

$$t_{go}^4\left(\mathbf{g}^T\mathbf{g}+\Gamma\right)-4t_{go}^2\left[\mathbf{v}^T\mathbf{v}-\left(\mathbf{r}_f-\mathbf{r}\right)^T\mathbf{g}\right]+16t_{go}^2\left(\mathbf{r}_f-\mathbf{r}\right)^T\mathbf{v}-12\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right)=0 \quad (2.113)$$

When $\mathbf{g}=0$ and $\Gamma=0$, this becomes

$$t_{go}^2\mathbf{v}^T\mathbf{v}-4t_{go}^2\left(\mathbf{r}_f-\mathbf{r}\right)^T\mathbf{v}+3\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right)=0 \quad (2.114)$$

Let θ be the angle between the vectors $\mathbf{v}$ and $(\mathbf{r}_f-\mathbf{r})$. The solution of Eq. 2.113 is obtained as

$$\tau=\frac{2|\mathbf{r}_f-\mathbf{r}|}{|\mathbf{v}|}\left(\cos\theta-\sqrt{\cos^2\theta-{}^3\!/_4}\right)\;;\quad\theta\in\left(-30^\circ,30^\circ\right) \quad (2.115)$$

There is a finite solution only when θ lies in the range (−30°, 30°).

There is no general way to derive an optimal time-to-go for IACG. Recall that IACG consists of FTVG along the final impact velocity direction, and CTVG perpendicular to this. Typically, then, there will be two different optimal times, and the true local minimum will be dependent on the particular mission geometry.

2.4.6 ZEM/ZEV Feedback Guidance

In the preceding section, the optimal feedback guidance laws were discussed assuming a uniform gravitational field ($\mathbf{g}$ = constant). If $\mathbf{g}$ is an explicit function of time, it is also possible to derive the optimal CTVG and FTVG algorithms.

Let the zero-effort-miss (ZEM) be the position offset at the end of the mission if no more acceleration is applied. Also let the zero-effort velocity (ZEV) be the end-of-mission velocity offset with no acceleration applied. The dynamic equations of motion with no control acceleration are

$$\begin{aligned}\dot{\mathbf{r}}&=\mathbf{v}\\\dot{\mathbf{v}}&=\mathbf{g}\left(t\right)\end{aligned} \quad (2.116)$$

These equations can be integrated to find the ZEV and ZEM as

$$\mathbf{ZEV} = \mathbf{v}_f - \left[\mathbf{v} + \int_t^{t_f} \mathbf{g}\left(t\right)\, \mathrm{d}\tau\right] \tag{2.117}$$

$$\mathbf{ZEM} = \mathbf{r}_f - \left[\mathbf{v}t_{go} + \int_t^{t_f} \left(t_f - \tau\right) \mathbf{g}\left(\tau\right)\, \mathrm{d}\tau\right] \tag{2.118}$$

With the ZEM and ZEV defined as above, the ZEM/ZEV version of the CTVG law is expressed as

$$\mathbf{a} = \frac{6}{t_{go}^2}\mathbf{ZEM} - \frac{2}{t_{go}}\mathbf{ZEV} \tag{2.119}$$

The FTVG law is expressed as

$$\mathbf{a} = \frac{3}{t_{go}}\mathbf{ZEM} \tag{2.120}$$

As was the case for the optimal laws, the ZEM/ZEV laws are optimal for a specified flight time. When a particular flight time is needed as a mission requirement, ZEM/ZEV laws can be applied using that flight time. If the exact flight time is not important, some additional analysis can be applied to find the optimal flight time. The optimal flight times found above for CTVG and FTVG, for example, can be used as a starting point for the optimal flight time.

2.4.7 Estimating the ZEM and ZEV

For the case when gravity is not constant, the ZEM and ZEV must be found somehow. There are three basic options, of varying complexity. The most complex option is simple numerical integration of the equations of motion. This method is computationally intensive, but will result in the most accurate estimates of ZEM and ZEV.

The second option employs the time-varying STM. Recall that the ZEM is the difference between the desired final position and the final position in the absence of corrective maneuvers. For the asteroid intercept/rendezvous problem, this can be expressed as

$$\mathbf{ZEM}\left(t\right) = \left[\mathbf{r}\left(t_f\right) - \mathbf{r}_f\right]_{\mathbf{a}=0} \tag{2.121}$$

where $\mathbf{r}\left(t_f\right)$ is the predicted spacecraft position at $t = t_f$ and $\mathbf{r}_f$ is the desired final position. Then the ZEM estimate using the time-varying STM becomes

$$\mathbf{ZEM}_{STM}\left(t\right) \approx \delta\mathbf{r}\left(t_f\right) = \Phi_1 \delta\mathbf{r}\left(t\right) + \Phi_2 \delta\mathbf{v}\left(t\right) \tag{2.122}$$

Similarly, the ZEV can be estimated as

$$\mathbf{ZEV}\left(t\right) = \left[\mathbf{v}\left(t_f\right) - \mathbf{v}_f\right]_{\mathbf{a}=0} \tag{2.123}$$

$$\mathbf{ZEV}_{STM}\left(t\right) \approx \delta\mathbf{r}\left(t_f\right) = \Phi_3\delta\mathbf{r}\left(t\right) + \Phi_4\delta\mathbf{v}\left(t\right) \tag{2.124}$$

Finally, for cases when the gravitational force is not significant, the ZEM and ZEV can be estimated by direct linearization of the relative states. Ignoring any external accelerations, we can estimate the ZEV as the current relative velocity and the ZEM as the current relative position plus the relative velocity times time-to-go, described as

$$\mathbf{ZEM}\left(t\right) \approx \delta\mathbf{r}\left(t\right) + \delta\mathbf{v}\left(t\right)\Delta t \tag{2.125}$$

$$\mathbf{ZEV}\left(t\right) \approx \delta\mathbf{v}\left(t\right) \tag{2.126}$$

2.4.8 Optimal Feedback Guidance Algorithms for a Special Case of $\mathbf{g} = \mathbf{g}\left(t\right)$

The gravitational acceleration is, in general, a function of position, which will not lead to an analytical solution of the optimal control problem. However, if the gravitational acceleration is assumed to be an explicit function of only time, then the analytical optimal solution can be found.

For a mission from time t_0 to t_f, the optimal control acceleration needs to be determined by minimizing the classical performance index of the form

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \tag{2.127}$$

subject to Eq. 2.116 and the following given boundary conditions:

$$\begin{aligned} \mathbf{r}\left(t_0\right) = \mathbf{r}_0 \quad \mathbf{r}\left(t_f\right) = \mathbf{r}_f \\ \mathbf{v}\left(t_0\right) = \mathbf{v}_0 \quad \mathbf{v}\left(t_f\right) = \mathbf{v}_f \end{aligned} \tag{2.128}$$

The Hamiltonian function for this problem is then defined as

$$H = \frac{1}{2}\mathbf{a}^T\mathbf{a} + \mathbf{p}_r^T\mathbf{v} + \mathbf{p}_v^T\left(\mathbf{g}\left(t\right) + \mathbf{a}\right) \tag{2.129}$$

where $\mathbf{p}_r$ and $\mathbf{p}_v$ are the co-state vectors associated with the position and velocity vectors, respectively.

The co-state equations provide the optimal control solution expressed as a linear combination of the terminal values of the co-state vectors. Defining the time-to-go, t_{go}, as $t_{go} = t_f - t$, the optimal acceleration at any time t is expressed as

$$\mathbf{a} = -t_{go}\mathbf{p}_r\left(t_f\right) - \mathbf{p}_v\left(t_f\right) \tag{2.130}$$

By substituting the above expression into the dynamic equations to solve for $\mathbf{p}_r\left(t_f\right)$ and $\mathbf{p}_v\left(t_f\right)$, the optimal control solution with the specified $\mathbf{r}_f$, $\mathbf{v}_f$, and t_{go} is finally obtained as

$$\mathbf{a} = \frac{\left[\mathbf{r}_f - \left(\mathbf{r} - t_{go}\mathbf{v}\right)\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f - \mathbf{v}\right)}{t_{go}} + \frac{6\int_t^{t_f}\left(\tau - t\right)\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau}{t_{go}^2} - \frac{4\int_t^{t_f}\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau}{t_{go}} \tag{2.131}$$

The zero-effort-miss (ZEM) distance and zero-effort-velocity (ZEV) error denote, respectively, the differences between the desired final position and velocity and the projected final position and velocity if no additional control is commanded after the current time. For the assumed gravitational acceleration $\mathbf{g}\left(t\right)$, the ZEM and ZEV have the following expressions:

$$\mathbf{ZEM} = \mathbf{r}_f - \left[\mathbf{r} + t_{go}\mathbf{v} + \int_t^{t_f}\left(t_f - \tau\right)\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau\right] \tag{2.132}$$

$$\mathbf{ZEV} = \mathbf{v}_f - \left[\mathbf{v} + \int_t^{t_f}\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau\right] \tag{2.133}$$

2.4.9 Generalized ZEM/ZEV Feedback Guidance

In addition to the terminal-phase guidance problem of asteroid missions discussed so far, the ZEM/ZEV concept can also be applied to a wide variety of orbital guidance problems with $\mathbf{g} = \mathbf{g}\left(\mathbf{x}, t\right)$. In certain applications, such as planetary landing, gravity can be simply assumed to be constant. In other applications, such as orbital transfer, nonlinearities are strong enough that the position-dependent gravitational acceleration is often dealt with by directly compensating (canceling) for it, rather than predicting its future effects. However, in some cases a generalized ZEM/ZEV feedback guidance law can be employed by introducing a number of waypoints. Details can be found in [24, 25] on how to practically compute such waypoints in real time for the generalized ZEM/ZEV feedback guidance applications.

2.5 Conclusions

The guidance algorithms described in this chapter are applicable to a wide variety of asteroid missions. The PN-based methods require only line-of-sight measurements, while more advanced methods require knowledge of the state of both the spacecraft and the target asteroid. Predictive guidance laws use only information about the current state to form guidance commands for future intercept. Optimal feedback guidance laws were also described, which can be used to deliver the spacecraft to a specified final position, or they can offer full final state control. When the mission time is not specified, conditions for an optimal mission time were found. Finally, these optimal guidance laws were shown to be part of a class of generalized ZEM/ZEV laws. Examples of these laws, and some practical considerations for implementation, were given. This dissertation establishes a firm foundation for continued research into asteroid missions, including topics such as waypoint guidance, and strategies for dealing with irregular gravitational fields of larger asteroids.

CHAPTER 3. Applications of Generalized Zero-Effort-Miss/Zero-Effort-Velocity Feedback Guidance Algorithm

The performance of the zero-effort-miss/zero-effort-velocity (ZEM/ZEV) feedback guidance algorithm is evaluated through practical space application examples. The ZEM/ZEV feedback guidance algorithm is in general not an optimal solution; however, it is an optimal solution in a uniform gravitational environment. It is also conceptually simple and easy to implement, and thus has great potential for autonomous on-board implementation. It is shown that, for some classic ballistic missile intercept and asteroid intercept scenarios, the ZEM/ZEV algorithm can even compete with corresponding open-loop optimal solutions, while its feedback characteristics make it more suitable to deal with uncertainties and perturbations. By employing the ZEM/ZEV algorithm in the highly nonlinear orbital transfer and raising problems and comparing with corresponding open-loop optimal solutions, its simplicity and near-optimality are further verified.

3.1 Introduction

Optimal control theory has been widely used for decades in many different applications; examples include spacecraft orbit control, missile guidance control, robot control, and flight vehicle trajectory control. [18, 27] The problem of controlling the trajectory of aerospace vehicles from an arbitrary initial position and initial velocity to a desired target position with constrained, free or pointed terminal velocity in a specific time, or within a predefined time range, is of fundamental interest as an optimal control problem.

Bryson and Ho discussed optimal control laws for a simple rendezvous problem, considering both free terminal velocity and constrained terminal velocity. [18] They also discussed the relationship between optimal control and proportional navigation guidance. Battin also discussed an optimal terminal state vector control for the orbit control problem, directly compensating for the known disturbing gravitational acceleration. [3] D'Souza further examined an optimal control algorithm in a uniform gravitational field, and developed a computational method to determine the optimal time-to-go. [20] Ebrahimi et al. proposed a robust optimal sliding mode guidance law for an exoatmospheric interceptor, using fixed-interval propulsive maneuvers. [4] In this paper, gravity was considered to be an explicit function of time. One major contribution of Ebrahimi et al. was the new concept of the zero-effort-velocity (ZEV) error, analogous to the well-known zero-effort-miss (ZEM) distance. The ZEV is the velocity error at the end of the mission if no further control accelerations are imparted. Furfaro et al. later employed the ZEM/ZEV concept to construct two classes of non-linear guidance algorithms for a lunar precision landing mission. [23] Guo et al. [21] showed that in a uniform gravitational field, the ZEM/ZEV logic is basically a generalized form of various well-known optimal feedback guidance solutions such as intercept or rendezvous, [18] terminal guidance, [3] and planetary landing. [20] The performance of the ZEM/ZEV logic for an asteroid intercept mission with precision targeting requirements was evaluated by Hawkins et al., [22] and compared with the performances of classical missile guidance methods like proportional navigation guidance (PNG) and augmented proportional navigation guidance (APNG).

For many practical missions, the gravitational acceleration is not constant nor an explicit function of time, but is instead a function of position. *The ZEM/ZEV algorithm is not an optimal solution when the gravitational acceleration is a function of position.* However, the ZEM and ZEV terms can be obtained by numerically propagating the dynamic equations, and the ZEM/ZEV algorithm can accomplish the control mission in a near-optimal manner. The objective of this chapter is to show how to use the generalized ZEM/ZEV algorithm for a variety of practical applications. For highly nonlinear systems, numerical propagation of the states for the entire remaining mission time is not sufficient, as nonlinearities during the actual mission violate the assumptions of the ZEM/ZEV algorithm. For these highly nonlinear cases, a general way to improve the performance of the ZEM/ZEV feedback algorithm is to divide the total flight time into one or more segments and somehow determine optimal or near-optimal waypoints to connect the different segments. Such a waypoint concept was considered by Sharma et al., [28] and the computational method was provided to solve nonlinear optimal control problems with terminal constraints.

In the last decade, pseudospectral optimization methods have been used for a variety of optimal control applications. [29, 30, 31] NASA's Transition Region and Corona Explorer (TRACE) spacecraft successfully flight-tested time-optimal slews in the presence of various constraints, [32] ushering in a new era of employing optimization techniques for spacecraft attitude control. A number of optimization software packages are now on the market, including SNOPT, DIDO, TOMLAB [33] and others. GPOPS (General Pseudospectral Optimal Control Software) is one of the most versatile open-source multi-phase optimizers, and is used in this dissertation. [34] GPOPS is used in this study to generate the open-loop optimal solution to compare the ZEM/ZEV algorithm against, and to obtain optimal waypoints to improve the performance of the generalized ZEM/ZEV algorithm for missions with highly nonlinear dynamics.

In this chapter, the generalized optimal control problem is first briefly reviewed. Three different types of ZEM/ZEV optimal feedback control algorithms are obtained, for different terminal requirements. Each algorithm is then investigated through an illustrative example, and the characteristics of each algorithm are discussed.

3.2 Optimal Feedback Guidance Algorithms

3.2.1 General Equations of Motion

The equations of motion of a space vehicle in a gravitational field are given by

$$\dot{\mathbf{r}} = \mathbf{v} \tag{3.1}$$

$$\dot{\mathbf{v}} = \mathbf{g}\left(\mathbf{r}\right) + \mathbf{a} \tag{3.2}$$

$$\mathbf{a} = \frac{\mathbf{T}}{m} \tag{3.3}$$

where $\mathbf{r}$ and $\mathbf{v}$ represent the position and velocity vectors, respectively; $\mathbf{a}$ is the control acceleration provided by the thrusting force $\mathbf{T}$; m is the vehicle mass; and $\mathbf{g}\left(\mathbf{r}\right)$ denotes the gravitational acceleration acting on the vehicle, which is generally a function of position. In this dissertation, these vectors denote 3x1 column vectors expressed in a non-rotating inertial reference frame.

3.2.2 Optimal Feedback Guidance Algorithms for a Special Case of $\mathbf{g} = \mathbf{g}\left(t\right)$

The gravitational acceleration is, in general, a function of position, which will not lead to an analytical solution of the optimal control problem. However, if the gravitational acceleration is assumed to be an explicit function of only time, then the analytical optimal solution can be found.

For a mission from time t_0 to t_f, the optimal control acceleration needs to be determined by minimizing the classical performance index of the form

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \tag{3.4}$$

subject to Eqs. 4.1 through 3.3 and the following given boundary conditions:

$$\begin{aligned} \mathbf{r}\left(t_0\right) &= \mathbf{r}_0 \quad \mathbf{r}\left(t_f\right) = \mathbf{r}_f \\ \mathbf{v}\left(t_0\right) &= \mathbf{v}_0 \quad \mathbf{v}\left(t_f\right) = \mathbf{v}_f \end{aligned} \tag{3.5}$$

The Hamiltonian function for this problem is then defined as

$$H = \frac{1}{2}\mathbf{a}^T\mathbf{a} + \mathbf{p}_r^T\mathbf{v} + \mathbf{p}_v^T\left(\mathbf{g}\left(t\right) + \mathbf{a}\right) \tag{3.6}$$

where $\mathbf{p}_r$ and $\mathbf{p}_v$ are the co-state vectors associated with the position and velocity vectors, respectively.

The co-state equations provide the optimal control solution expressed as a linear combination of the terminal values of the co-state vectors. Defining the time-to-go, t_{go}, as $t_{go} = t_f - t$, The optimal acceleration at any time t is expressed as

$$\mathbf{a} = -t_{go}\mathbf{p}_r\left(t_f\right) - \mathbf{p}_v\left(t_f\right) \tag{3.7}$$

By substituting the above expression into the dynamic equations to solve for $\mathbf{p}_r\left(t_f\right)$ and $\mathbf{p}_v\left(t_f\right)$, the optimal control solution with the specified $\mathbf{r}_f$, $\mathbf{v}_f$, and t_{go} is finally obtained as

$$\mathbf{a} = \frac{\left[\mathbf{r}_f - \left(\mathbf{r} - t_{go}\mathbf{v}\right)\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f - \mathbf{v}\right)}{t_{go}} + \frac{6\int_t^{t_f}\left(\tau - t\right)\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau}{t_{go}^2} - \frac{4\int_t^{t_f}\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau}{t_{go}} \tag{3.8}$$

The zero-effort-miss (ZEM) distance and zero-effort-velocity (ZEV) error denote, respectively, the differences between the desired final position and velocity and the projected final position and velocity if no additional control is commanded after the current time. For the assumed gravitational acceleration $\mathbf{g}\left(t\right)$, the ZEM and ZEV have the following expressions [22, 28]:

$$\mathbf{ZEM} = \mathbf{r}_f - \left[\mathbf{r} + t_{go}\mathbf{v} + \int_t^{t_f}\left(t_f - \tau\right)\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau\right] \tag{3.9}$$

$$\mathbf{ZEV} = \mathbf{v}_f - \left[\mathbf{v} + \int_t^{t_f}\mathbf{g}\left(\tau\right)\,\mathrm{d}\tau\right] \tag{3.10}$$

Then, the optimal control law, Eq. 3.8, can be equivalently expressed as

$$\mathbf{a} = \frac{6}{t_{go}^2}\mathbf{ZEM} - \frac{2}{t_{go}}\mathbf{ZEV} \tag{3.11}$$

For certain missions where the terminal velocity is not specified, the optimal control law, in terms of ZEM only, can be obtained as

$$\mathbf{a} = \frac{3}{t_{go}^2}\mathbf{ZEM} \tag{3.12}$$

Though of limited interest for most intercept and rendezvous missions, the optimal control to regulate only the terminal velocity, in terms of ZEV only, can also be obtained as

$$\mathbf{a} = \frac{1}{t_{go}}\mathbf{ZEV} \tag{3.13}$$

A special case of uniform gravitational environment can be assumed for many planetary landing and asteroid terminal guidance problems. The three optimal algorithms described by Eqs. 3.11 through 3.13 then become the exact optimal solutions that achieve the optimal feedback performance and maintain robustness against uncertainties. For other missions, though, the gravitational acceleration cannot be simply modeled as a constant (or as a pure function of time), but must be considered a function of position. For this case, the ZEM and ZEV can be found by numerically integrating the dynamic equations. The same control expressions are used, with the numerically propagated values of ZEM and ZEV. The predictions and controls are updated in real-time, finally accomplishing the control mission at near-optimal levels with acceptable computational complexity.

For highly nonlinear systems, predicting the future states is prone to errors. Another alternative form of the ZEM/ZEV algorithm can be adopted for this situation. Rather than predicting the effect of the nonlinear terms, the effects of these terms are directly compensated for at all times. The algorithm thus approaches feedback linearization behavior. The control algorithm, Eq. 3.8, then simply becomes the following form suggested by Battin in [3]:

$$\mathbf{a} = \frac{6\left[\mathbf{r}_f - (\mathbf{r} + t_{go}\mathbf{v})\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f - \mathbf{v}\right)}{t_{go}} - \mathbf{g}\left(\mathbf{r}\right) \tag{3.14}$$

3.3 Optimization with GPOPS Software

GPOPS uses hp-adaptive pseudospectral methods to solve optimal control problems of one or more phases. The user supplies the governing dynamic equations, cost function, and various constraints, and, if feasible, GPOPS returns the optimal control histories. The theoretical basis and implementation of GPOPS are now briefly described.

Pseudospectral numerical optimization methods approximate the states, co-states, and controls using Lagrange interpolating polynomials. Since Lagrange polynomials and their derivatives can be evaluated for any time, the states and controls can be represented by a discrete set of points. If there is an integral in the cost functional, this is approximated with Gaussian quadrature. An hp-adaptive method can automatically adjust the mesh points (h-adaptivity) and the order of the approximating polynomial (p-adaptivity). The complete problem is thus posed as a nonlinear program (NLP), which can be solved using one of a variety of well-known NLP solvers.

The GPOPS user must supply the governing differential equations, minimum and maximum values for all of the states and controls, as well as minimum and maximum values for the initial and final times. Guesses for the initial and final values of all of the states, controls, and times are also supplied. Finally, the cost functional, a combination of the standard Mayer (endpoint) and Lagrange (running) costs, is specified.

The initial time is typically fixed, and the final time can be fixed or free. In a multi-phase problem, each phase can have a minimum or maximum duration. For example, a three-stage rocket could be designed with the first two phases lasting a specified length of time, and the third phase and, thus, the final time, free.

In addition to these inputs, some optional constraints may be specified. Event constraints are equality or inequality constraints that are expressed as functions of initial and final time, as well as initial and final values of the states. A typical event constraint might be ensuring that the final position falls on some specified curve. Path constraints are equality or inequality constraints expressed as functions of the time, states, and controls. A typical path constraint might be ensuring that the norm of the controls always equals control magnitude for a constant-thrust mission. Parameter constraints are equality or inequality constraints on parameters, variables that are constant during the mission but not known ahead of time. A typical parameter constraint might be the mass ratio of a staged rocket.

Because the controls and states are approximated with polynomials, the controls and states (and co-states) must be smooth within a given phase. The examples considered in this chapter, and many other guidance problems with continuous thrust, are smooth in this sense. Known or suspected discontinuities and singularities, such as ”bang-bang” control or problems with singular arcs, can be handled by breaking the problem into multiple phases. Even in these cases, GPOPS can often find a good approximation to the discontinuous solution in a single phase, which can then be refined into a multi-phase problem. GPOPS can find numerical derivatives of the various constraints, but if available, analytical expressions should be used to enhance speed and accuracy. Certain constraints, such as control saturation, do not have continuous derivatives. For such cases, an approximation with continuous derivatives can be used.

3.4 Ballistic Missile Intercept Example

Before introducing the full ZEM/ZEV algorithm, it is instructive to consider ZEM control for a well-known example problem. The ballistic missile intercept problem is considered, to put considerations of the ZEM and the mission time in familiar terms. With this background, the full ZEM/ZEV algorithm will be introduced in the next example problem.

The objective of an example problem here is to examine the performance of guidance algorithms for a tactical missile to intercept a ballistic missile target. The dynamic models for the missile and target are represented as [1]

$$\begin{aligned} \ddot{x}_M &= \frac{-\mu x_M}{\left(x_M^2 + y_M^2\right)^{1.5}} + a_{xM} = g_{xM} + a_{xM} \\ \ddot{y}_M &= \frac{-\mu y_M}{\left(x_M^2 + y_M^2\right)^{1.5}} + a_{yM} = g_{xM} + a_{yM} \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} \ddot{x}_T &= \frac{-\mu x_T}{\left(x_M^2 + y_M^2\right)^{1.5}} = g_{xM} \\ \ddot{y}_T &= \frac{-\mu y_T}{\left(x_M^2 + y_M^2\right)^{1.5}} = g_{yM} \end{aligned} \tag{3.16}$$

where the subscript M is used for the missile, the subscript T is used for the target ballistic missile, (x, y) are position vector components in an Earth-centered inertial coordinate system, μ is the gravitational parameter of the Earth (3.986x10ˆ14 mˆ3/sˆ2), and a is the control acceleration.

The line-of-sight (LOS) is defined as the line from the missile to the target. The LOS angle, λ, is the angle between this line and a constant reference line in the inertial frame, and it is expressed as

$$\lambda = \tan^{-1} \frac{y_{TM}}{x_{TM}} \tag{3.17}$$

where (x_{TM}, y_{TM}) are the relative position components in the inertial frame defined as

$$\begin{aligned} x_{TM} &= x_T - x_M \\ y_{TM} &= y_T - y_M \end{aligned} \tag{3.18}$$

The relative velocity components, (v_x, v_y), are also defined as

$$\begin{aligned} v_x &= \dot{x}_{TM} = \dot{x}_T - \dot{x}_M \\ v_y &= \dot{y}_{TM} = \dot{y}_T - \dot{y}_M \end{aligned} \tag{3.19}$$

The distance from the missile to the target is

$$r = \sqrt{x_{TM}^2 + y_{TM}^2} \tag{3.20}$$

3.4.1 Classical Guidance Algorithms

Before discussing an application of the ZEM/ZEV algorithms, first the classical proportional navigation guidance (PNG) law will be described. The classical PNG law is simply given by

$$\begin{bmatrix} a_{xM} \\ a_{yM} \end{bmatrix} = N V_c \dot{\lambda} \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{3.21}$$

where N is the effective navigation ratio, a user-adjustable parameter, and the closing velocity, V_c, and LOS rate, $\dot{\lambda}$, are determined as

$$V_c = -\dot{r} = \frac{-\left(x_{TM} v_x + y_{TM} v_y\right)}{r} \tag{3.22}$$

$$\dot{\lambda} = \frac{x_{TM} v_y - y_{TM} v_x}{r^2} \tag{3.23}$$

The classical PNG algorithm commands acceleration perpendicular to the instantaneous LOS direction. For a missile with aerodynamic actuators, the actual acceleration would be perpendicular to the missile's velocity vector. For small turn rates, the missile velocity vector and LOS direction are approximately aligned.

When the gravitational environment is known, the augmented PNG (APNG) algorithm can be used to improve the performance of the PNG. The APNG law is

$$\begin{bmatrix} a_{xM} \\ a_{yM} \end{bmatrix} = N \left(V_c \dot{\lambda} + \frac{1}{2} \left(g_{T\perp} - g_{M\perp} \right) \right) \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{3.24}$$

where $g_{T\perp}$ and $g_{M\perp}$ are components gravity acting on the missile and target, perpendicular to the LOS direction. The best PNG law, described in [1], is the so-called predictive guidance algorithm.

When accurate models of the missile and target dynamics are known, the positions of the missile and target at the planned intercept time can be found by numerical integration. The ZEM vector is determined as

$$\mathbf{ZEM} = \begin{bmatrix} ZEM_x \\ ZEM_y \end{bmatrix} = \begin{bmatrix} \tilde{x}_{TF} - \tilde{x}_{MF} \\ \tilde{y}_{TF} - \tilde{y}_{MF} \end{bmatrix} \tag{3.25}$$

where $(\tilde{x}_{TF}, \tilde{y}_{TF})$ are the predicted final position components of the target and $(\tilde{x}_{MF}, \tilde{y}_{MF})$ are the predicted final position components of the missile if no further accelerations are imparted.

The predictive guidance algorithm is based on PNG, so only components normal to the LOS are considered. The control acceleration is then given as

$$\begin{bmatrix} a_{xM} \\ a_{yM} \end{bmatrix} N \frac{ZEM_{\perp}}{t_{go}^2} \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{3.26}$$

where

$$ZEM_{\perp} = -ZEM_x \sin\lambda + ZEM_y \cos\lambda$$

3.4.2 Feedback Guidance Using Generalized ZEM Algorithm

The optimal ZEM feedback algorithm, given by Eq. 3.12, is then expressed as

$$\begin{bmatrix} a_{xM} \\ a_{yM} \end{bmatrix} = \frac{3}{t_{go}^2} \begin{bmatrix} ZEM_x \\ ZEM_y \end{bmatrix} \tag{3.27}$$

The predictive guidance algorithm, Eq. 3.26, differs from the generalized ZEM algorithm [Eq. 3.27] in two main ways. First, the predictive algorithm restricts accelerations to be perpendicular to the LOS, while the ZEM algorithm can command accelerations in any direction. Second, the predictive algorithm leaves the navigation constant N as an adjustable parameter, while the ZEM algorithm has a fixed gain to ensure optimality.

Both algorithms are derived assuming that the total flight time is specified. There are some cases where enforcing a particular mission time may be desirable, such as intercepting a target missile before multiple payloads can be deployed, or ensuring that intercept does not occur above a particular location on Earth. For most missions, though, the particular time of intercept is not crucial, and is not specified beforehand. Although in principle a broad range of intercept times could be specified, missions that are too short or too long can experience problems. More control energy will be consumed, which may exceed the capacity of the actuators and result in a failure to intercept.

Recall that the ZEM is a function of the time-to-go, in addition to being a function of the current missile and target states. The simplest way to determine the time-to-go is to choose the time-to-go that corresponds to the minimum norm of the ZEM. The ZEM is found by numerically integrating the current states without control accelerations, so the ZEM is found at every time step in the integration. Finding the minimum norm of the ZEM is equivalent to simply predicting the time of closest approach. The goal of the control law is to reduce the ZEM to zero, so choosing the time of minimum ZEM corresponds to exploiting the dynamics to give the control algorithm the smallest error to overcome.

A simple line search can be used to find the best time-to-go. As the dynamic equations are integrated, there will be a ZEM associated with each time step. Due to the assumptions for PN guidance, the missile must be headed ”toward” the target. When this is true, the predicted ZEM will monotonically decrease with increasing t_f, reach a minimum, then monotonically increase. Finding the time-to-go is just a matter of integrating until the predicted ZEM starts to increase. The smaller of the last two ZEM predictions corresponds to the best time-to-go.

3.4.3 Numerical Simulation Example

A ballistic intercept scenario, examined in [1], was used to evaluate both the predictive and ZEM algorithms. With “perfect” initial conditions, the predictive and ZEM laws will both achieve impact with no control accelerations (an ideal situation), while the PNG and APNG laws will still require some control accelerations. To be able to compare the predictive and ZEM laws, the example was modified appropriately. The initial position and initial velocity magnitude are the same, but the initial velocity direction is changed slightly. The initial conditions are $\mathbf{r}_{T0} = (0, 6378.245)\,\mathrm{km}$, $\mathbf{v}_{T0} = (6785, 2880)\,\mathrm{m/s}$, $\mathbf{r}_{M0} = (4510.1, 4510.1)\,\mathrm{km}$, $\mathbf{v}_{M0} = (2006, 5954)\,\mathrm{m/s}$.

Solving the optimal control problem with GPOPS is straightforward for the ballistic missile intercept. There are no limits on the control inputs, and generous limits on the states can be used. For example, the x and y positions can be constrained to be from negative two to positive two earth radii. GPOPS will converge more quickly if there are finite bounds. The flight time can be left free for the overall optimal solution, or specified to find the optimal intercept for a given mission time.

Figure 3.1 shows comparisons between the generalized ZEM algorithm with adaptive t_{go} and the open-loop optimal solution, as generated by GPOPS. The trajectories, control histories, and performance index histories are shown. The adaptive flight time algorithm finds the best flight time to be 687 s, a little less than the optimal time of 700 s. The control histories, however, are similar, and the performance index value is only 2% larger.

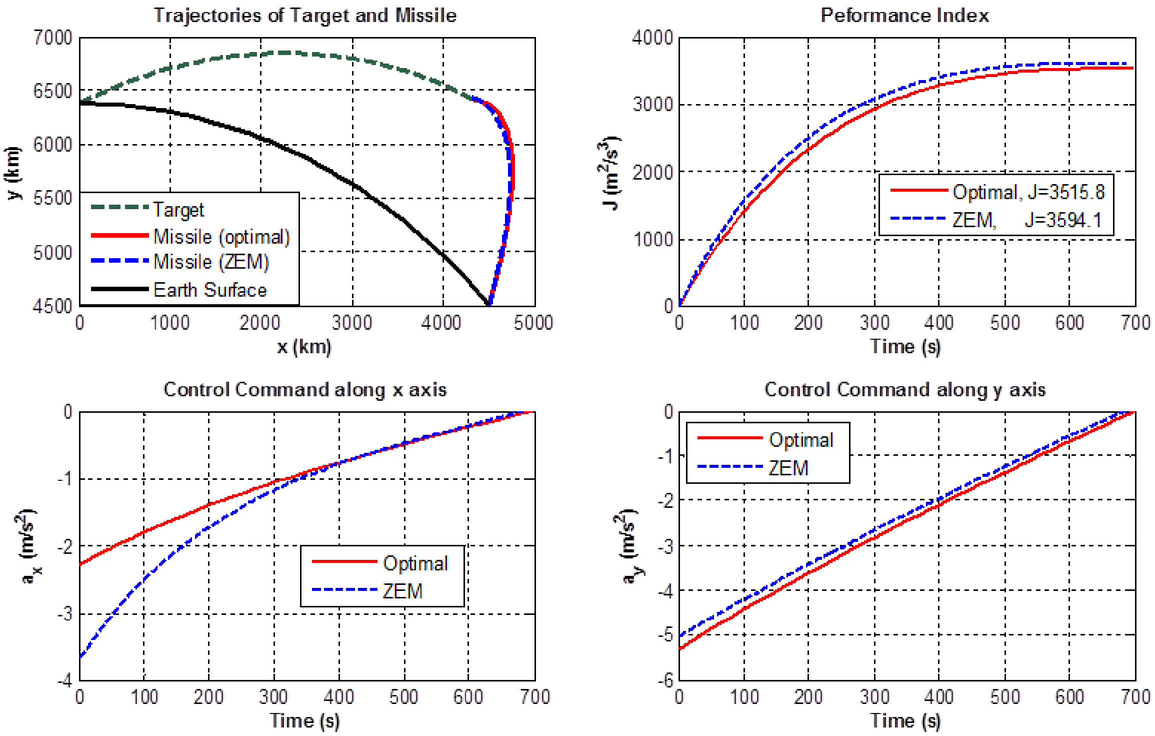


Figure 3.1 Missile intercept using the generalized ZEM algorithm and an open-loop optimal approach.

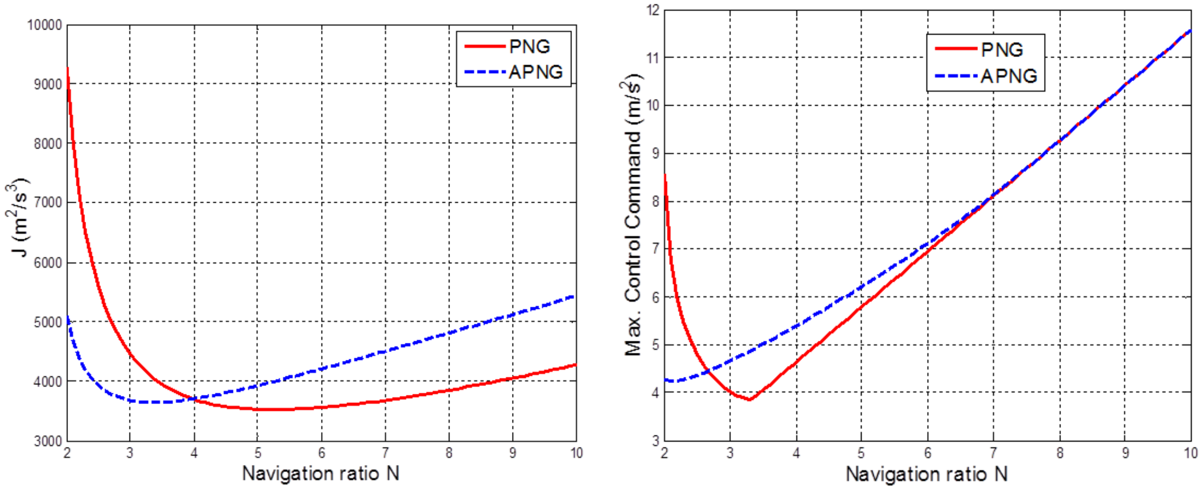
The ZEM algorithm has a fixed gain to ensure optimality, while the gain is an adjustable parameter for the PNG-based methods. The optimal value for the navigation constant turns out to be 3, but it is usually chosen in the range from 3 to 5. Larger values cause the vehicle to turn onto a collision course more quickly, at the expense of more control effort. This increased control effort is manifest in two different ways. First, the performance index for the mission increases. Second, the maximum level of commanded acceleration increases, which can become important if there is an upper limit on available accelerations.

To study the effect of changing the navigation constant, the intercept mission example was simulated for both the PNG law and the APNG law, for a variety of N ranging from 2 to 10. Figure 3.2 shows the performance index with varying N, as well as the maximum control magnitude required. For the test case shown, the optimal N for PNG is 5.3, while the optimal

N for APNG is 3.4. Table 3.1 gives a detailed comparison of the different guidance laws. The ZEM law with adaptive flight time is shown, as well as the ZEM law using the known optimal flight time of 700 s.

Table 3.1 Performance of various guidance methods for missile intercept

	PNG	APNG	ZEM	ZEM	Open-Loop
	($N = 5.3$)	($N = 3.4$)	(adaptive t_f)	($t_f = 700$s)	Optimal
Performance Index J	3526.5	3648.2	3594.1	3515.9	3515.8
Max Control, m/s^2	6.14	4.94	6.25	5.79	5.79
Flight time t_f, s	701	702	687	700	700



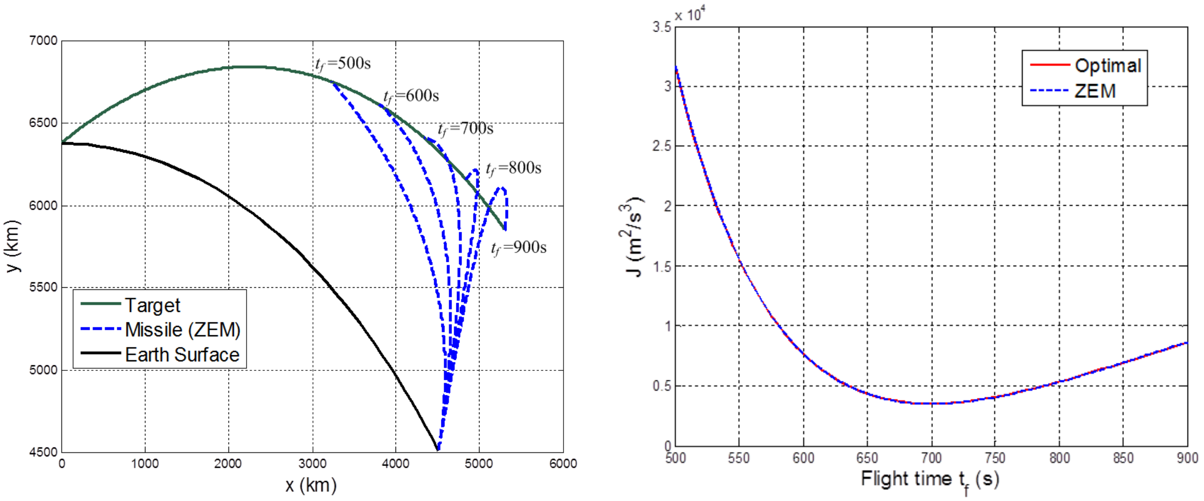
(a) Performance index.

(b) Maximum magnitude of control.

Figure 3.2 Missile intercept using PNG and APNG with different navigation ratios.

The best PNG law performs better than the ZEM law with adaptive t_f, which in turn is better than the APNG law. The ZEM law with the known optimal flight time performs best of all, effectively identical to the optimal. The specific performance of the different algorithms depends on the setup of the problem. The optimal values for N are only found by simulating many cases, and the optimal flight time for the ZEM algorithm comes from first finding the open-loop optimal solution. If the maximum control magnitude is a concern, there is a limit to how large a navigation ratio is acceptable. And although the best case for the APNG does not perform as well as the best-case PNG law, other tradeoffs for J and control magnitude must be considered when the optimal N cannot be determined beforehand.

One advantage the ZEM law has over the PNG-based laws is the ability to control flight time. A mission with the same initial conditions was simulated, with flight times ranging from 500 s to 900 s. Figure 3.3 shows the trajectories for various cases, as well as the performance index J for both the ZEM law and the open-loop optimal solution. Not only is the ZEM law able to intercept at a variety of times, it does so with barely any discernible difference from the optimal.



(a) Vehicle trajectories using ZEM.

(b) Performance Comparison with the optimal.

Figure 3.3 ZEM algorithm with various specified flight times.

3.5 Asteroid Proximity Operation Example

The problem of detecting a possibly threatening near-Earth object (NEO) and responding to that threat has been given much consideration in recent years. One major task of the planetary defense community is to find an optimal approach to avert a potential NEO-Earth collision. [35] Although there is no universally accepted definition of optimal mitigation for this problem, three broad categories of deflection missions have emerged. The first is a slow-push scheme to gradually change the NEO's orbit. The second is a high-speed intercept mission by a massive spacecraft, changing the NEO's orbit via a kinetic impact. [9, 19] The third approach is a nuclear detonation for large NEOs, or when there is little mission lead time. A number of different guidance algorithms can be employed for such asteroid intercept or rendezvous missions.

3.5.1 Terminal Guidance for Asteroid Intercept

Consider both the target asteroid and the interceptor spacecraft as point masses in a heliocentric Keplerian orbit, with the equations of motion described by

$$\ddot{\mathbf{r}}_T = -\frac{\mu}{r_T^3}\mathbf{r}_T \tag{3.28}$$

$$\ddot{\mathbf{r}}_S = -\frac{\mu}{r_s^3}\mathbf{r}_S - \frac{\mu_\otimes}{r^3}\mathbf{r} + \mathbf{a} \tag{3.29}$$

where $\mathbf{r}_T$ and $\mathbf{r}_S$ are the position vectors of the target and the interceptor, respectively, with magnitudes r_T and r_S, μ is the gravitational parameter of the sun (1.32715x10ˆ20 mˆ3/sˆ2), $\mu_\otimes$ is the estimated gravitational parameter of the asteroid, $\mathbf{a}$ is the applied control acceleration, and $\mathbf{r}$ is the relative position vector of the interceptor (with magnitude r) with respect to the asteroid, defined as

$$\mathbf{r} = \mathbf{r}_S - \mathbf{r}_T \tag{3.30}$$

Neglecting the asteroid's gravitational parameter, the equations of motion are essentially the same as for missile intercept. The ZEM algorithm can be obtained as

$$\mathbf{a} = -\frac{3}{t_{go}^2}\tilde{\mathbf{r}} = \frac{3}{t_{go}^2}\left(\tilde{\mathbf{r}}_{TF} - \tilde{\mathbf{r}}_{SF}\right) \tag{3.31}$$

where $\tilde{\mathbf{r}}_{TF}$ and $\tilde{\mathbf{r}}_{SF}$ are the predicted final positions of the asteroid and interceptor if no further control accelerations are applied, and t_{go} can be based on a specified final time or adjusted as described in the previous section.

For the case where the gravitational effect is negligible, the optimal time-to-go can be calculated based on the current relative position $\mathbf{r}$ and relative velocity $\mathbf{v}$, as follows: [21]

$$t_{go} = \frac{2r}{v}\left(\cos\theta - \sqrt{\cos^2\theta - \frac{3}{4}}\right), \quad \theta \in (-30°, 30°) \tag{3.32}$$

where θ is the angle between $\mathbf{r}$ and $-\mathbf{v}$. There exists a solution for t_{go} that locally minimizes the optimal performance index only when θ is in the above range. Outside that range, the performance index decreases monotonically with increasing t_{go}, and some upper bound on the flight time must be selected.

3.5.2 Terminal Guidance for Asteroid Landing

For an asteroid landing problem, we assume that the asteroid is a sphere with radius $R_\otimes$. For convenience, we also ignore the negligible gravitational acceleration of the asteroid, so the equations of motion become

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v}\\ \dot{\mathbf{v}} &= \mathbf{a}\end{aligned} \tag{3.33}$$

The terminal velocity is by definition zero for the landing mission. If the landing site is specified, one of the algorithms from [8] can be used. When the landing site is not specified, then the final position must meet the following constraint

$$\mathbf{r}_f^T\mathbf{r}_f = R_\otimes^2 \tag{3.34}$$

Incorporating the terminal constraint, the performance index can be modified as follows:

$$J = \frac{1}{2}\sigma\left(\mathbf{r}_f^T\mathbf{r}_f - R_\otimes^2\right) + \int_{t_0}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \tag{3.35}$$

where σ is the scalar multiplier for the terminal constraint. The co-state vector has the following terminal constraint

$$\mathbf{p}_r\left(t_f\right) = \frac{\partial J}{\partial \mathbf{r}_f} = \sigma\mathbf{r}_f \tag{3.36}$$

The optimal landing site $\mathbf{r}_f^*$ can be found as

$$\mathbf{r}_f^* = R_\otimes\frac{t_{go}\left(\mathbf{v}+\mathbf{v}_f\right)+2\mathbf{r}}{\|t_{go}\left(\mathbf{v}+\mathbf{v}_f\right)+2\mathbf{r}\|} = R_\otimes\frac{t_{go}\mathbf{v}+2\mathbf{r}}{\|t_{go}\mathbf{v}+2\mathbf{r}\|} \tag{3.37}$$

Finally, the guidance algorithm for optimal asteroid soft landing is obtained as

$$\mathbf{a} = \frac{6\left[\mathbf{r}_f^* - \left(\mathbf{r}+t_{go}\mathbf{v}\right)\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f-\mathbf{v}\right)}{t_{go}} \tag{3.38}$$

Similar to the intercept problem, an optimal time-to-go exists for certain initial conditions. [21] The optimal time-to-go is found by solving the final equation

$$t_{go}^2\mathbf{v}^T\mathbf{v} - t_{go}6\mathbf{v}^T\left(\mathbf{r}_f-\mathbf{r}\right) + 9\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right) = 0 \tag{3.39}$$

where the condition

$$\left[6\mathbf{v}^T\left(\mathbf{r}_f-\mathbf{r}\right)\right]^2-\left(4\mathbf{v}^T\mathbf{v}\right)9\left(\mathbf{r}_f-\mathbf{r}\right)^T\left(\mathbf{r}_f-\mathbf{r}\right)\leq 0 \tag{3.40}$$

is required for a local minimum of t_{go}. When this minimum does not exist, t_{go} needs to be as large as possible. However, there is no guarantee that any particular t_{go} will not intercept the surface of the asteroid, so a numerical simulation is required to evaluate feasibility, using the chosen t_{max} as the initial guess.

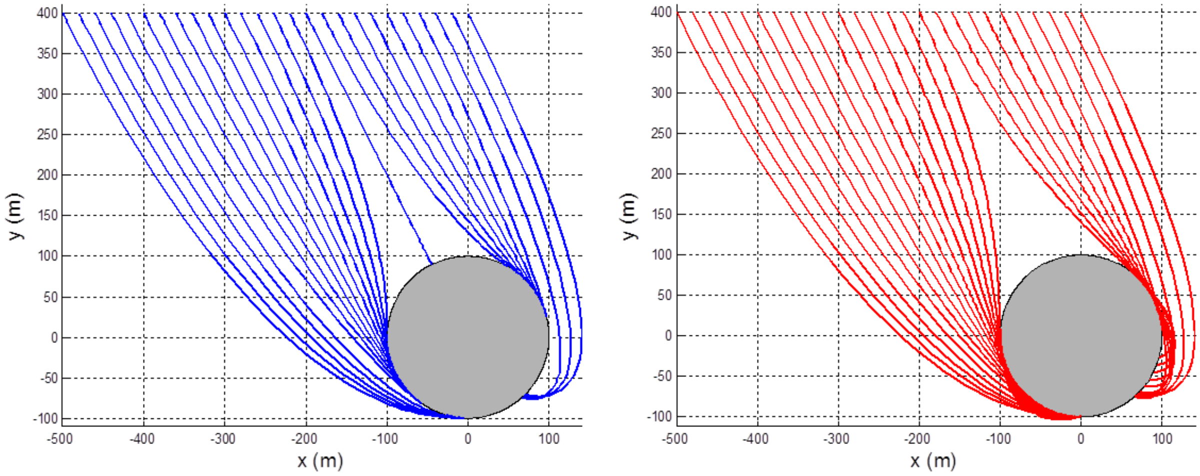
3.5.3 Numerical Simulation Example

An asteroid landing problem using the guidance law expressed by Eq. 3.38 was numerically simulated for a variety of initial conditions. The target asteroid has a radius of 100 m, and is assumed to be centered at the origin of the coordinate system. The lander has an initial velocity of (20, -40, 0) m/s. Various initial positions are tested to illustrate the performance of the ZEM/ZEV algorithm with adaptive t_f and landing site. For initial conditions without a local minimum for the performance index, the ultimate upper bound of the flight time is 40 s.

Solving the optimal asteroid landing problem with GPOPS requires only a couple of constraints. There is a terminal event constraint requiring that the final position lies on a circle or radius 100 m. The final time is free, with the designer-chosen upper bound of 40 s. In this case, the optimal solution always takes the maximum amount of time. As in the previous example, generous bounds on the states and controls are provided to make the search space finite.

Figure 3.4 shows the trajectories using the ZEM/ZEV approach and the open-loop optimal trajectories obtained from GPOPS. Figure 3.5 compares the calculated flight time and performance index for the ZEM/ZEV law and the open-loop optimal solution.

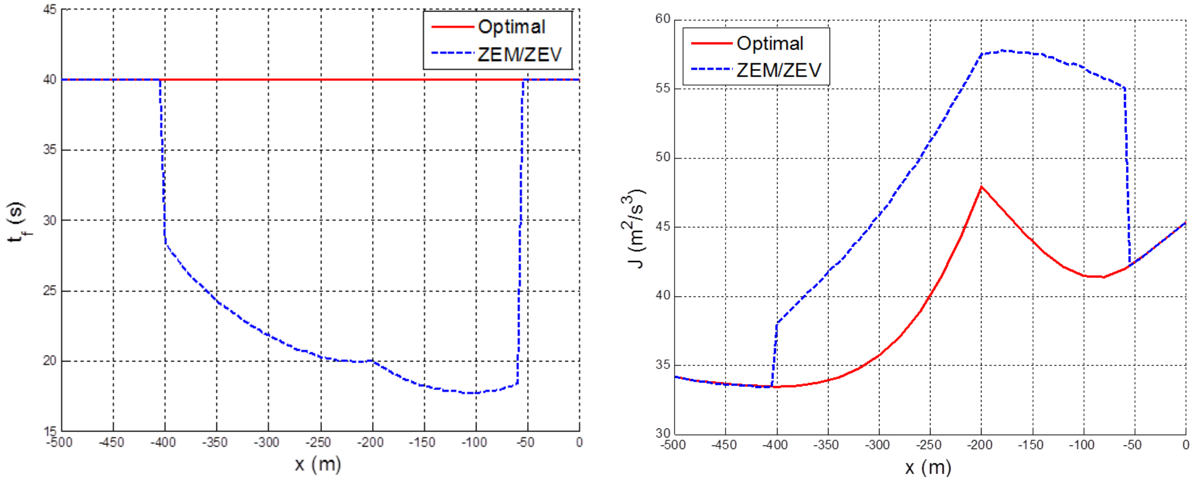
The landing mission can be successfully completed for all of the initial conditions considered. The initial conditions roughly fall into two different types. The first type includes initial conditions that will collide with, or come close to colliding with, the asteroid if the lander continues on a straight-line trajectory. The second type has initial conditions that will travel well outside the asteroid's footprint. For the second type of initial conditions, the ZEM/ZEV algorithm performs identically to the open-loop optimal solution, which verifies the optimal



(a) ZEM/ZEV with adaptive t_f and landing site.

(b) Open-loop optimal.

Figure 3.4 Vehicle trajectories for asteroid landing ($t_f = 40$s).



(a) Flight time comparison.

(b) Performance comparison.

Figure 3.5 Performance comparison between ZEM/ZEV and open-loop optimal ($t_f = 40$s).

feature of autonomous landing site calculation approach. For initial conditions on a collision course (or nearly so), a collision hazard is detected and the flight time is adjusted downward until the mission is safe. The reduced mission time leads to more acceleration commanded.

One trajectory of particular interest is the case when the spacecraft starts on a collision course through the center of the asteroid. The ZEM/ZEV algorithm will simply reduce the mission time and arrest the spacecraft's forward velocity, while the optimal solution is to turn the spacecraft to fly around the asteroid while taking the full 40 s.

3.6 Orbit Transfer Example with Continuous Thrust

The objective of an orbital transfer/raising problem is to optimally transfer a spacecraft from a lower orbit to a higher orbit, with a specified injection point and velocity, at a given time. The spacecraft can also be brought from a higher orbit to a lower one. For high-thrust engines, the well-known impulsive Hohmann transfer is the minimum-energy transfer, however for continuous low-thrust engines, other methods must be used.

For simplicity, consider the following spacecraft dynamic equations:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \end{aligned} \tag{3.41}$$

$$\begin{aligned} \dot{v}_x &= -\mu \frac{x}{\left(x^2+y^2\right)^{1.5}} + a_x \\ \dot{v}_y &= -\mu \frac{y}{\left(x^2+y^2\right)^{1.5}} + a_y \end{aligned} \tag{3.42}$$

where (x, y) and (v_x, v_y) denote the position components and velocity components in heliocentric inertial orbit plane, μ is the gravitational parameter of the sun, and (a_x, a_y) are control accelerations along (x, y) axes.

The only requirement for the control system is to ensure that the spacecraft satisfies the following terminal conditions at the final time t_f

$$\begin{aligned} x\left(t_f\right) &= x_c \quad & y\left(t_f\right) &= y_c \\ v_x\left(t_f\right) &= v_{xc} \quad & v_y\left(t_f\right) &= v_{yc} \end{aligned} \tag{3.43}$$

3.6.1 Application of ZEM/ZEV Feedback Guidance Algorithm

The equations of motion are strongly coupled, and an analytic optimal control algorithm does not exist. The ZEM/ZEV algorithm, Eq. 3.11, can control the terminal position and velocity, at a specified final time. These encompass all of the requirements of the orbit transfer mission, making it a good candidate for this problem. Expressed in the $x-$ and $y-$coordinates, the proposed ZEM/ZEV law becomes

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \frac{6}{t_{go}^2} \begin{bmatrix} ZEM_x \\ ZEM_y \end{bmatrix} - \frac{2}{t_{go}} \begin{bmatrix} ZEV_x \\ ZEV_y \end{bmatrix} \tag{3.44}$$

where the ZEM and ZEV are obtained by subtracting the predicted terminal states (with no further control accelerations) from the required terminal states, as follows:

$$\begin{aligned} \begin{bmatrix} ZEM_x \\ ZEM_y \end{bmatrix} &= \begin{bmatrix} x_c - \tilde{x}_F \\ y_c - \tilde{y}_F \end{bmatrix} \\ \begin{bmatrix} ZEV_x \\ ZEV_y \end{bmatrix} &= \begin{bmatrix} v_{xc} - \tilde{v}_{xF} \\ v_{yc} - \tilde{v}_{yF} \end{bmatrix} \end{aligned} \tag{3.45}$$

3.6.2 Numerical Simulation Example

An orbit transfer example from Earth to Mars is considered here to evaluate the performance of the generalized ZEM/ZEV algorithm. Feedback guidance control is not generally needed for such an orbital transfer mission, because an open-loop optimal trajectory can easily be generated during the long mission time. We examine this case here only as an illustrative example to demonstrate the applicability of the ZEM/ZEV feedback guidance concept.

For ease of analysis, canonical (or normalized) units will be used. For Earth's orbit, the mean distance to the sun is 1 AU (astronomical unit), 1.4959965x10ˆ11 m. Defining 1 TU (time unit) as 58.132821 days gives the Earth a circular orbital velocity of 1 AU/TU. Mars orbit is at a radius of 1.54 AU, with a velocity of 0.8059 AU/TU. In these units, the gravitational parameter μ is 1 AUˆ3/TUˆ2.

For this mission, the spacecraft starts at (1, 0) AU with velocity of (0, 1) AU/TU. The terminal position is (-0.3986, 1.4875) AU, the terminal velocity is (-0.7784, -0.2086) AU/TU.

The flight time is 144 days, or 2.4771 TU.

Finding the optimal control with GPOPS is straightforward in this case, as the initial and final position and velocity are fully specified, as is the mission time. Again, generous limits on the states and controls are given to maintain a finite search space.

Figure 3.6 shows the position, velocity, and acceleration histories for both the ZEM/ZEV algorithm and the optimal open-loop solution determined by GPOPS. Figure 3.7 shows the transfer orbits for both cases. The direction and normalized magnitude of the acceleration commands are also shown every 1/10 of the mission.

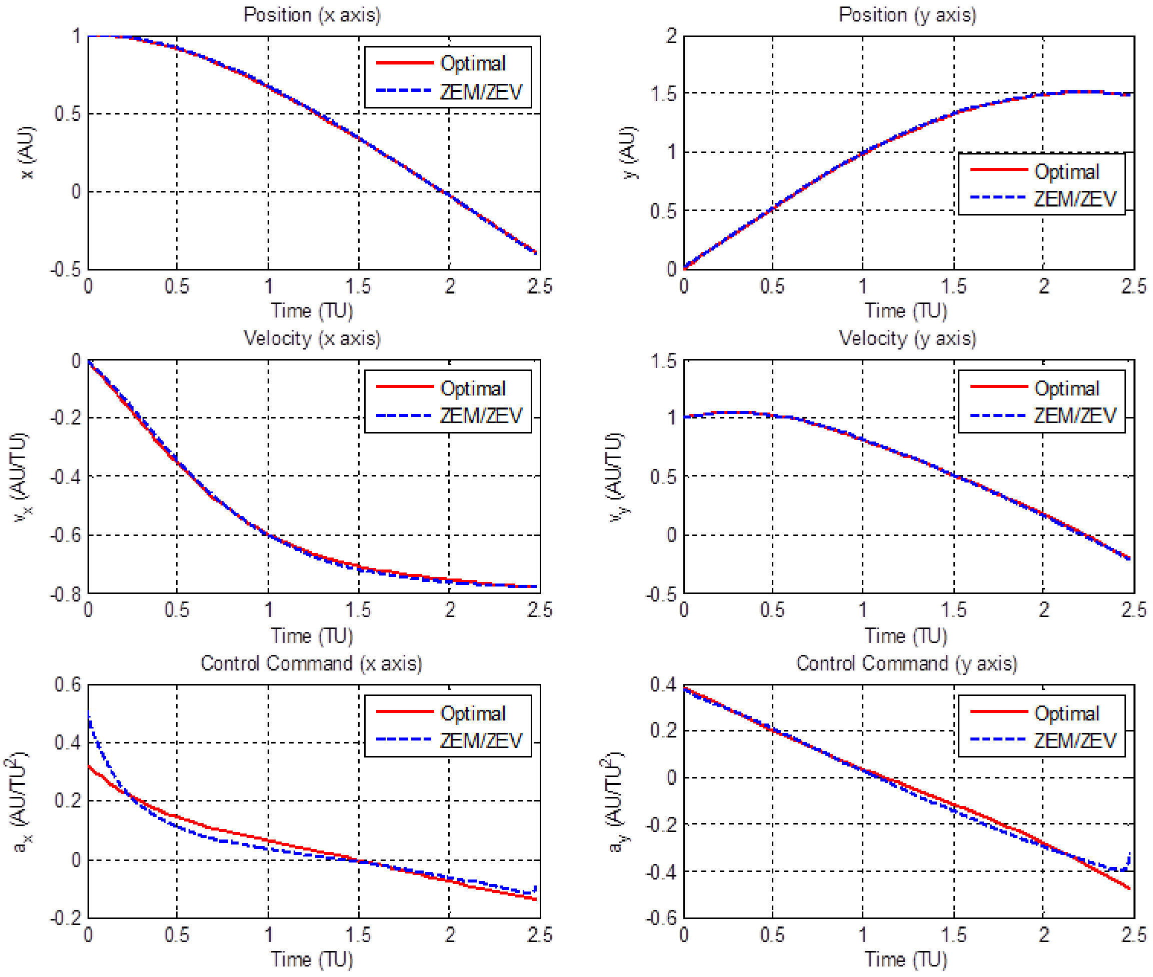


Figure 3.6 144-Days orbit transfer from Earth to Mars.

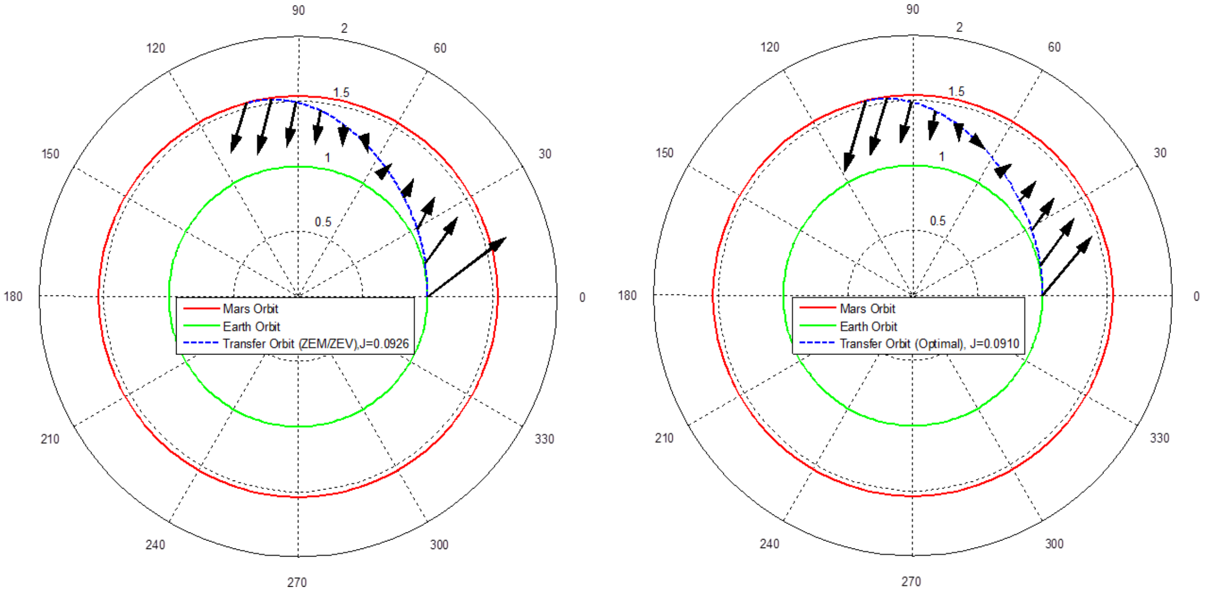


Figure 3.7 144-Days orbit transfer from Earth to Mars.

3.7 Orbit Raising Problem

The objective of an orbit raising problem is to transfer a spacecraft from one orbit to another orbit. For an orbital transfer to a circular orbit, the terminal constraints are that the spacecraft should be placed at a specified distance from the sun with circular orbital velocity. The final radial velocity is zero, while the true anomaly (or any equivalent angular position) is free. Due to the nature of the constraints, polar coordinates are used. The standard dynamical models for this type of orbit raising problem are described by

$$\begin{aligned} \dot{r} &= u \\ \dot{u} &= \frac{v^2}{r} - \frac{\mu}{r^2} + a_r \\ \dot{v} &= -\frac{uv}{r} + a_t \end{aligned} \tag{3.46}$$

where r, u, and v represent the distance of the spacecraft from the sun, the radial velocity, and transverse velocity, respectively; and a_r and a_t are control accelerations in the radial and transverse directions, respectively. The required terminal states, as described above, are

$$\begin{aligned} r\left(t_f\right) &= r_f \\ u\left(t_f\right) &= 0 \\ v\left(t_f\right) &= \sqrt{\frac{\mu}{r_f}} \end{aligned} \tag{3.47}$$

3.7.1 Application of ZEM/ZEV Feedback Guidance

The orbit raising problem is somewhat unusual in that the control requirements are different along the radial and tangential axes. In the radial direction, there are position and velocity requirements as usual. In the tangential direction, we have the rare case where only the velocity is specified. The feedback algorithm is a combination of Eqs. 3.11 and 3.13, as follows:

$$\begin{aligned} a_r &= \frac{6}{t_{go}^2} ZEM_r - \frac{2}{t_{go}} ZEV_r \\ a_t &= \frac{1}{t_{go}} ZEV_t \end{aligned} \tag{3.48}$$

where the ZEM and ZEV are, as before, the difference between the required and predicted terminal states defined as

$$\begin{aligned} ZEM_r &= r_f - \tilde{r}_f \\ ZEV_r &= -\tilde{u}_f \\ ZEV_t &= \sqrt{\frac{\mu}{r_f}} - \tilde{v}_f \end{aligned} \tag{3.49}$$

Due to nonlinearities in the system, the ZEM/ZEV algorithm with direct gravity compensation, Eq. 3.14, can be used as

$$\begin{aligned} a_r &= \frac{6}{t_{go}^2}\left(r_f - \left(r + t_{go} u\right)\right) - \frac{2}{t_{go}}\left(u_f - u\right) - \left(\frac{v^2}{r} - \frac{\mu}{r^2}\right) \\ a_t &= \frac{1}{t_{go}}\left(v_f - v\right) - \left(\frac{uv}{r}\right) \end{aligned} \tag{3.50}$$

The first ZEM/ZEV algorithm, Eq. 3.48, works by numerically predicting the final states, so it is called the ZEM/ZEV predicting algorithm, or ZEM/ZEV-p. The second algorithm, Eq. 3.50, works by directly compensating for the gravitational acceleration terms, so it is called the ZEM/ZEV compensating algorithm, or ZEM/ZEV-c.

3.7.2 Numerical Simulation Example

The same initial conditions as the Mars orbit transfer problem are considered, with the same flight time. In the polar coordinate system, we have $r(t_0) = 1\text{AU}$, $u(t_0) = 0\text{AU/TU}$, $v(t_0) = 1\text{AU/TU}$, and $t_f = 2.4771\text{TU}$. Both proposed ZEM/ZEV algorithms, described by Eqs. 3.48 and 3.50, are used, called ZEM/ZEV-p and ZEM/ZEV-c, respectively. The optimal open-loop solution, generated by GPOPS, is also shown.

Implementing GPOPS for the basic orbit raising problem is relatively simple. The flight time is fixed. The final radial and tangential velocities are specified, as is the final radius. The final rotation angle is free. Later, a more interesting implementation of a multiphase optimal control problem with GPOPS will be described.

Figure 3.8 shows the position, velocity, and acceleration histories in the radial and tangential directions for all three cases. The different algorithms perform similarly in the radial direction, but are noticeably different in the tangential direction.

Sharma et al. describes a waypoint method for the same orbit raising problem. [28] The problem was broken up into several segments, where the terminal state for each segment is a waypoint in the overall problem. A series solution method (SSM) was then used to connect the waypoints. The ZEM/ZEV algorithm can be used as part of a waypoint method to improve performance. For this problem, the total flight time is divided into equal-length segments, and GPOPS is used to generate waypoints, enforcing the control law during the flight. These waypoints are then used as intermediate points for control law simulation. The gravity compensation form, ZEM/ZEV-c, is used for the waypoint method.

Optimal waypoints for the orbit raising problem can be found with GPOPS. The principle of optimality ensures that the optimal solution in multiple phases is equivalent to the optimal solution as one phase. To find waypoints, then, the problem is broken up into any number of segments of equal duration. The six-dimensional waypoint (three position components and three velocity components) is left as a set of six parameters to be found by GPOPS. Parameter variables are constant during each phase, but not known ahead of time. When GPOPS solves the optimal control problem, the optimal parameters (the optimal waypoints) are found.

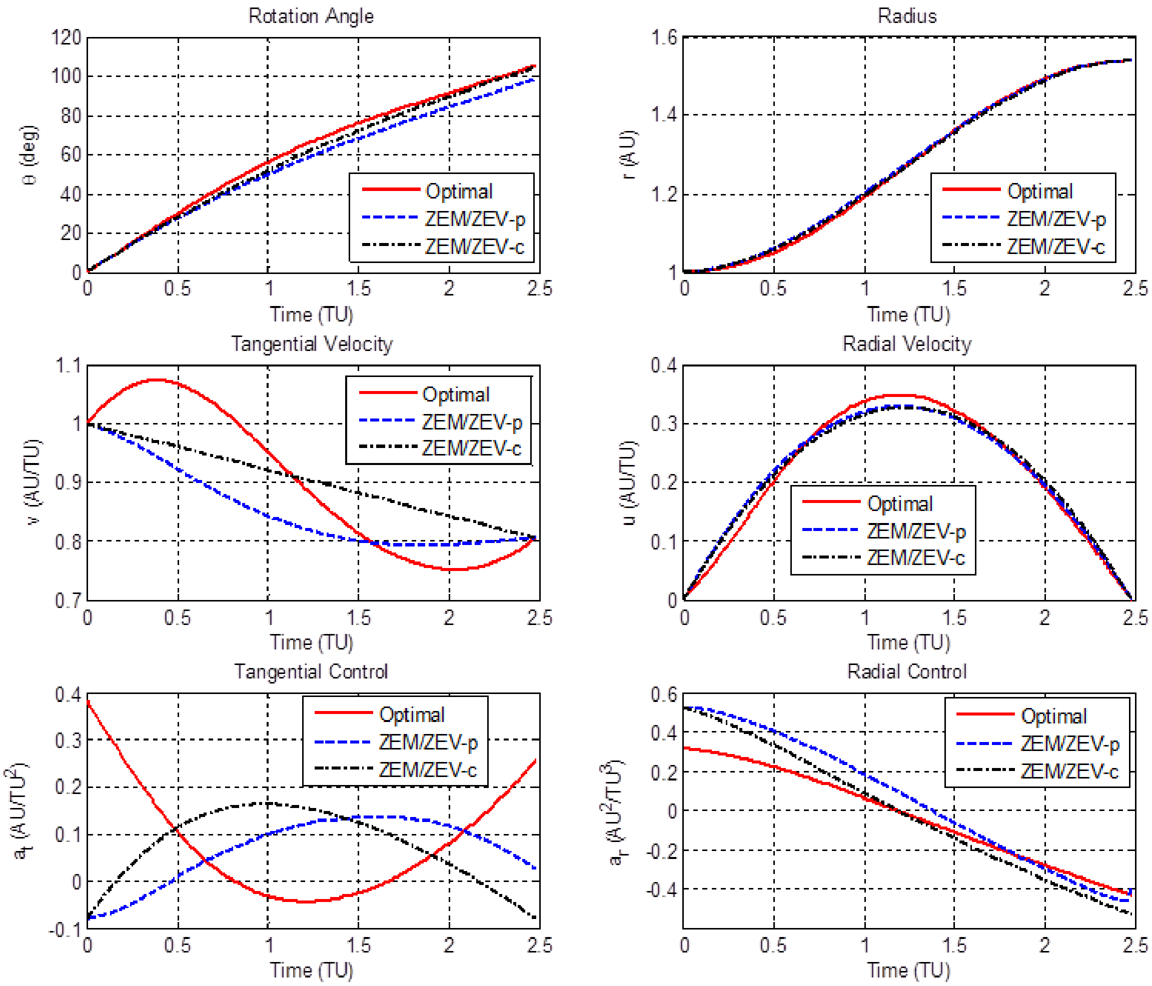


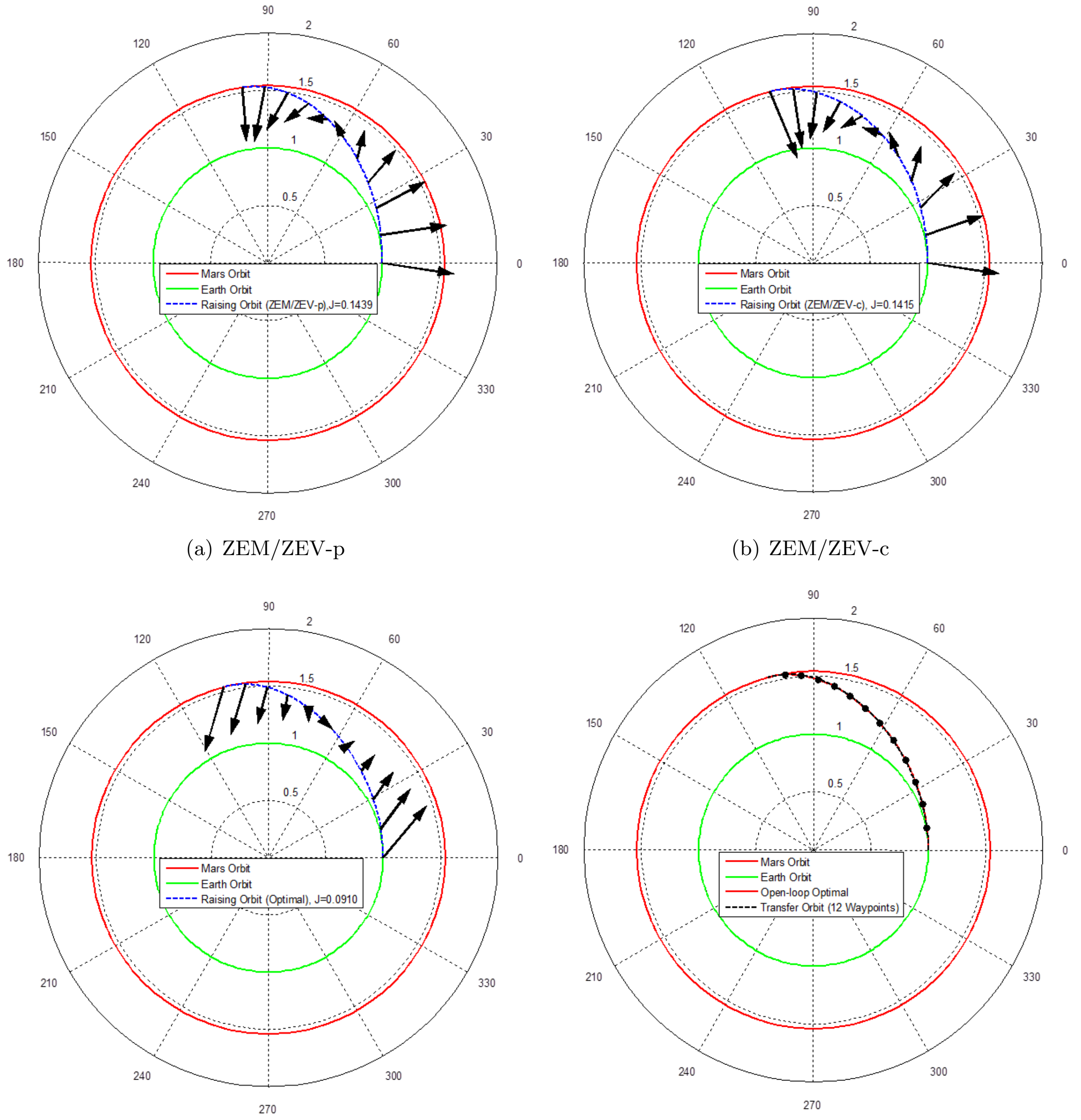
Figure 3.8 144-Days orbit raising from Earth orbit to Mars orbit.

ZEM/ZEV control can then be used, with each waypoint used as the final position and final velocity for a given phase. A much more in-depth discussion of optimal waypoint determination, applied to a Mars landing problem, is given in [24].

Figure 3.9 shows the orbit raising trajectories for all four methods discussed, ZEM/ZEV-p, ZEM/ZEV-c, optimal open-loop, and the waypoint method with 12 waypoints. The first three of these plots show the normalized accelerations every 1/10 of the mission time, while the last shows the locations of the waypoints. Table 3.2 compares the performance of the ZEM/ZEV waypoint method and the SSM waypoint method. Four cases are considered, using 1, 2, 4, and 12 total waypoints. The case with no waypoints, as well as the open-loop optimal solution, are also shown for comparison.

The two ZEM/ZEV algorithms perform similarly, with a performance index 50-60% larger than the open-loop optimal. The compensating algorithm performs better than the predicting algorithm due to the nonlinear coupled terms. The predicting algorithm does not command as large of an angular change, as can be seen in the rotation angle plot in Fig. 3.8. This plot shows that control effort is wasted trying to overcome misleading terms in the dynamic equations. The acceleration vectors in Fig. 3.9 show that the ZEM/ZEV algorithms spend too much effort in the radial direction, which must be made up for later. The ZEM/ZEV algorithms also show the opposite trends in tangential control from the optimal. Because the control directions are separated, the tangential control channel cannot account for the effects of radial acceleration on tangential velocity. One way to overcome these problems is to simply specify a terminal position and velocity, changing the orbit raising problem to an orbital transfer problem. For such case of orbital transfer problem formulation, the ZEM/ZEV comes within 2% increase of the open-loop optimal performance index value.

The ZEM/ZEV algorithm can also be improved by implementing a waypoint scheme. The adverse effects from nonlinear terms and coupled dynamics are reduced for shorter mission times. By breaking the mission up into many shorter segments, the feedback properties of the ZEM/ZEV algorithm can be preserved, while approaching optimal performance. The ZEM/ZEV algorithm compares favorably compared to the SSM method as the number of waypoints is increased, as can be seen in Table 3.2.



(a) ZEM/ZEV-p

(b) ZEM/ZEV-c

(c) Open-loop optimal

(d) Waypoint-optimized ZEM/ZEV and open-loop optimal

Figure 3.9 Orbit-raising trajectories using ZEM/ZEV and open-loop optimal methods.

Table 3.2 Performance Comparisons between ZEM/ZEV Algorithms and SSM

					Open-Loop
	No waypoint	2 waypoints	4 waypoints	12 waypoints	Optimal
SSM	0.1314	0.1051	0.0982	0.0942	
ZEM/ZEV-c	0.1415	0.1152	0.0996	0.0922	0.0910

3.8 Conclusions

Four different applications of the generalized zero-effort-miss/zero-effort-velocity (ZEM/ZEV) feedback guidance algorithm have been investigated in this chapter. The application examples were the ballistic missile intercept problem, the asteroid intercept and landing problem, and the orbital transfer/raising problems. For cases when the gravitational acceleration can be assumed to be independent of the vehicle's state, three different feedback-optimal ZEM/ZEV algorithms are considered. For many practical missions, the gravitational acceleration is a function of the vehicle's state. By numerically propagating the system state, corresponding generalized ZEM/ZEV algorithms can be obtained.

Numerical simulations demonstrated that the generalized ZEM/ZEV guidance algorithm can achieve intercept at a specified time. When the mission time is not specified, performance can be improved with a flight-time adaptive approach. ZEM/ZEV feedback guidance is conceptually simple, and is easy to implement. It works for many different cases, as the gains are pre-defined, and do not need to be adjusted based on experience or on the specific problem.

The ZEM/ZEV algorithm can be used for a case, such as asteroid landing, where the only the magnitude of the terminal position is specified. Results of numerical simulations show the feasibility of the approach, including autonomous landing site selection.

For highly nonlinear systems with coupled dynamics, such as the orbital transfer/raising problem, numerical simulations have confirmed the effectiveness of the generalized ZEM/ZEV algorithm. For some missions, the performance of the ZEM/ZEV algorithm is significantly worse than the open-loop optimal solution. In these cases, a series of waypoints can be found using commercially available optimization software. The ZEM/ZEV algorithm is then used between each waypoint. The ZEM/ZEV waypoint algorithm approaches the performance of the open-loop optimal solution, while maintaining the robustness of a closed-loop feedback algorithm.

CHAPTER 4. Waypoint-Optimized Zero-Effort-Miss/Zero-Effort-Velocity Feedback Guidance for Mars Landing

This chapter investigates the optimization approach to generate waypoints for the Mars landing problem in the context of employing the zero-effort-miss/zero-effort-velocity (ZEM/ZEV) feedback guidance algorithm. For a power-limited engine, the waypoint optimization problem in the presence of state constraints is converted to an equivalent standard quadratic programming problem, which can be solved efficiently. In the case with a thrust-limited engine, by introducing a continuously differentiable function to approximate the standard saturation function, the optimal waypoint can be determined using open-source optimization software. This novel idea exploits parameter optimization techniques for feedback control implementation, thus it can combine the advantages of open-loop and closed-loop methods to achieve near-optimal performance with acceptable robustness, while meeting various practical constraints and requirements.

4.1 Introduction

The Mars pinpoint landing problem continues to be an active area of research. A considerable amount of effort has been devoted to it, with many new issues being investigated. The general Mars entry, descent, and landing (EDL) mission is divided into four phases: the hypersonic entry phase, the subsonic parachute entry phase, the propulsive terminal descent (or powered descent) phase, and the touchdown phase. [36, 37] Due to the difficulty of accounting for accumulated uncertainties from the first two phases, past landing missions, including the Viking series (launched in 1975), Mars Pathfinder (1996), and Mars Exploration Rover (2003), focused only on a safe landing. Landing accuracies larger than 10 km were acceptable for these missions. Even the recent Phoenix lander (2007) used a gravity turn descent, since precision landing is not required. [38] The Mars Science Laboratory (launched on Nov. 26, 2011) is expected to use a precision EDL system, to reach a more scientifically interesting area with hazardous terrain nearby, and to demonstrate advanced landing capabilities.

Açıkmeşe and Ploen presented an off-line convex optimization approach for fuel-optimal Mars pinpoint landing. [26] This approach takes many complex constraints into account, especially the non-convex lower bound on thruster levels. The approach was extended to a minimum landing error problem for the case where no feasible pinpoint landing trajectories exist. [39] These approaches, however, are not robust against disturbances, and require precise mathematical modeling. Their open-loop natures make them inapplicable for autonomous onboard implementation, given the wide range of potential initial conditions and the short propulsive terminal descent phase.

Ebrahimi et al. proposed a robust optimal sliding mode guidance law for an exoatmospheric interceptor, using fixed-interval propulsive maneuvers. [4] In this paper gravity was considered to be a pure function of time. One major contribution of Ebrahimi et al. was the new concept of the zero-effort-velocity (ZEV) error, analogous to the well-known zero-effort-miss (ZEM) distance. The ZEV is the velocity error at the end of the mission if no further control accelerations are imparted. Furfaro et al. later employed the ZEM/ZEV concept to construct two classes of non-linear guidance algorithms for a lunar precision landing mission. [23] The performance of the ZEM/ZEV algorithm for an asteroid intercept mission with precision targeting requirements was evaluated by Hawkins et al., [22] and compared with the performances of classical missile guidance methods like proportional navigation guidance (PNG) and augmented PNG (APNG). Reference [21] showed that in a uniform gravitational field, the ZEM/ZEV algorithm is equivalent to well-known classical optimal feedback controllers for problems such as intercept or rendezvous, [18] terminal guidance, [3] and planetary landing. [28, 40]

All of the optimal feedback guidance algorithms discussed above are concerned only with the specified boundary conditions. For the Mars landing problem, this is not sufficient. For example, the basic ZEM/ZEV algorithm does not ensure that a lander will stay above the surface of Mars for the duration of the mission. The ZEM/ZEV algorithm, then, must be adapted to avoid collision with the surface of Mars, or other constraints.

Steinfeldt et al. describe a method of adjusting the flight time to meet constraints on control magnitude and altitude. [40] Changing only the flight time, however, is of limited utility when dealing with landing mission constraints. Control saturation limits can be alleviated by increasing the mission time, so that the same total amount of control effort can be distributed over a longer period. Collision avoidance, especially avoiding collision with the surface, can usually be achieved by reducing the mission time, driving the lander closer to a “straight shot” trajectory. These contradictory demands cannot both be satisfied by changing the mission flight time.

In general, a Mars lander should be able to predict its trajectory and evaluate the possibility of impact during the flight. If a hazard is detected, effective measures can be taken. Yanning et al. [21] include an exploratory study of waypoint guidance, choosing a waypoint as an intermediate target to meet mission constraints. Near-optimal performance can be achieved with only one waypoint. In this study, however, the waypoint was chosen by trial and error, and a more rigorous, autonomous method is needed. The waypoint method is also studied by Sharma et al. [28] as a means to solve nonlinear optimal control problems with terminal constraints.

In the last decade pseudospectral optimization methods have been used for a variety of optimal control applications. [29, 30, 31] NASA’s Transition Region and Corona Explorer (TRACE) spacecraft successfully flight-tested time-optimal slews in the presence of various constraints, ushering in a new era of employing optimization techniques for advanced space missions. [32] A number of optimization software packages are now on the market, including SNOPT, DIDO, TOMLAB, [33] and others. GPOPS (General Pseudospectral Optimal Control Software) is one of the most versatile open-source multi-phase optimizers using pseudospectral methods, and is used in this dissertation. Its latest version offers a mesh refinement algorithm to accurately distribute collocation points. [34]

This chapter begins with a brief summary of optimal ZEM/ZEV algorithms. Key results are given, including a method for determining time-to-go. Two types of spacecraft engine are considered. First is a power-limited engine. The total flight time is divided into two parts, with the acceleration assumed to be a linear function of time in each segment. The state constraint is considered as an inequality constraint, and the waypoint optimization problem becomes a standard quadratic programming problem. Although the power-limited engine may be impractical to be employed for Mars landing, it is used here to illustrate the applicability of the ZEM/ZEV method for different thruster systems, but not necessarily as a practical mission example. For the thrust-limited engine, GPOPS is used to determine optimal waypoints. GPOPS is suitable for constraining the control commands to be a function the current and terminal states. To improve the efficiency and accuracy of the optimization, a continuously differentiable function is developed that approaches the classical saturation function, so that analytical derivatives can be passed to GPOPS. The novelty of this waypoint research lies in the combination of the optimal properties of the optimization approach and the robustness of ZEM/ZEV feedback control. Numerical simulations confirm the effectiveness of this approach.

4.2 Equations of Motion

Neglecting aerodynamic drag, the trajectory equations of motion during the powered descent phase for a Mars lander are described by

$$\dot{\mathbf{r}} = \mathbf{v} \tag{4.1}$$

$$\dot{\mathbf{v}} = \mathbf{g} + \mathbf{a} \tag{4.2}$$

$$\mathbf{a} = \frac{\mathbf{T}}{m} \tag{4.3}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the position and velocity vectors, $\mathbf{a}$ is the control acceleration provided by the thrusters, $\mathbf{T}$ is the thrusting force vector, m is the spacecraft mass, and $\mathbf{g}$ is the gravitational acceleration vector acting on the vehicle. During the powered descent phase, $\mathbf{g}$ is considered to be constant. These vectors are 3x1 column vectors expressed in a non-rotating inertial reference frame with its origin at the landing site of Mars.

The magnitude of the thrusting force, T, is modeled as the two-norm of a vector $\mathbf{T}$ as

$$T = |\mathbf{T}| = -\dot{m}c \tag{4.4}$$

where $\dot{m}$ is the negative mass flow rate, and $c = I_{sp}g_0$ is the engine exhaust velocity.

4.2.1 Optimal Feedback Guidance Algorithms

Consider a classical optimization problem with the following performance index

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \tag{4.5}$$

subject to Eqs. 4.1 through 4.4 and the following boundary conditions:

$$\begin{aligned} \mathbf{r}\left(t_0\right) = \mathbf{r}_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \qquad \mathbf{r}\left(t_f\right) = \mathbf{r}_f = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} \\ \mathbf{v}\left(t_0\right) = \mathbf{v}_0 = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \qquad \mathbf{v}\left(t_f\right) = \mathbf{v}_f = \begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{z}_f \end{bmatrix} \end{aligned} \tag{4.6}$$

The Hamiltonian function is defined as

$$H = \frac{1}{2}\mathbf{a}^T\mathbf{a} + \mathbf{p}_r^T\mathbf{v} + \mathbf{p}_v^T\left(\mathbf{g} + \mathbf{a}\right) \tag{4.7}$$

where $\mathbf{p}_r$ and $\mathbf{p}_v$ are the co-state vectors associated with the position and velocity vectors, respectively.

From the co-state equations, the optimal control solution can be expressed as a linear combination of the terminal value of co-state vectors as

$$\mathbf{a} = -t_{go}\mathbf{p}_r\left(t_f\right) - \mathbf{p}_v\left(t_f\right) \tag{4.8}$$

where the time-to-go, t_{go}, is defined as: $t_{go} = t_f - t$.

By substituting the above expression into the dynamic equations and solving for $\mathbf{p}_r\left(t_f\right)$ and $\mathbf{p}_v\left(t_f\right)$, the optimal control law with the specified $\mathbf{r}_f$, $\mathbf{v}_f$, and t_f is obtained as

$$\mathbf{a} = \frac{6\left[\mathbf{r}_f - \left(\mathbf{r} + t_{go}\mathbf{v}\right)\right]}{t_{go}^2} - \frac{2\left(\mathbf{v}_f - \mathbf{v}\right)}{t_{go}} - \mathbf{g} \tag{4.9}$$

The zero-effort-miss (ZEM) distance and zero-effort-velocity (ZEV) error denote the differences between the desired final position and velocity and the projected final position and velocity if no additional control is commanded after the current time. For the assumed constant gravitational acceleration, the ZEM and ZEV have the following expressions: [4, 21, 22, 23]

$$\mathbf{ZEM} = \mathbf{r}_f - \left(\mathbf{r} + t_{go}\mathbf{v} + \frac{1}{2}t_{go}^2\mathbf{g}\right) \tag{4.10}$$

$$\mathbf{ZEV} = \mathbf{v}_f - \left(\mathbf{v} + t_{go}\mathbf{g}\right) \tag{4.11}$$

Using these expressions, Eq. 4.9 can be rewritten as

$$\mathbf{a} = \frac{6}{t_{go}^2}\mathbf{ZEM} - \frac{2}{t_{go}}\mathbf{ZEV} \tag{4.12}$$

For the Mars soft landing problem, the landing site is typically chosen as the origin of the reference frame, so $\mathbf{r}_f = 0$, $\mathbf{v}_f = 0$, the optimal guidance law has the well-known simple form: [21, 36]

$$\mathbf{a} = -\frac{6\mathbf{r}}{t_{go}^2} - \frac{4\mathbf{v}}{t_{go}} - \mathbf{g} \tag{4.13}$$

4.3 Determination of Time-To-Go

The optimal mission time-to-go can be determined as the minimum real positive solution of the following equation: [21]

$$\mathbf{g}^T\mathbf{g}t_{go}^4 - 4\left(\mathbf{v}^T\mathbf{v} + \mathbf{v}_f^T\mathbf{v} + \mathbf{v}_f^T\mathbf{v}_f\right)t_{go}^2 + 24\left(\mathbf{r}_f - \mathbf{r}\right)^T\left(\mathbf{v} + \mathbf{v}_f\right)t_{go} - 36\left(\mathbf{r}_f - \mathbf{r}\right)^T\left(\mathbf{r}_f - \mathbf{r}\right) = 0 \tag{4.14}$$

The above equation can be simplified for the Mars soft landing problem due to the simple terminal state requirements, as follows:

$$\mathbf{g}^T\mathbf{g}t_{go}^4 - 4\mathbf{v}^T\mathbf{v}t_{go}^2 - 24\mathbf{r}^T\mathbf{v}t_{go} - 36\mathbf{r}^T\mathbf{r} = 0 \tag{4.15}$$

The altitude component of the vehicle's position can thus be expressed as

$$y\left(t\right) = -\frac{\left(t_f - t\right)^3}{6}p_{ry} + \frac{\left(t_f - t\right)^2}{2}\left(g - p_{vy}\right) \tag{4.16}$$

where y is the altitude of the vehicle, g is the gravitational acceleration, and p_{ry} and p_{vy} are the components of the co-state vectors along the altitude direction, given by

$$p_{ry} = \frac{6\dot{y}_0}{t_f^2} + \frac{12y_0}{t_f^3} \tag{4.17}$$

$$p_{vy} = -\frac{2\dot{y}_0}{t_f} - \frac{6y_0}{t_f^2} + g \tag{4.18}$$

The upper limit on flight time, which ensures a positive altitude in the time period $(0, t_f)$, can be found by analyzing the distribution of the roots of Eq. 4.16, which gives

$$t_{max} = -\frac{3y_0}{\dot{y}_0} \tag{4.19}$$

If the optimal t_f obtained from Eq. 4.15 exceeds the upper limit t_{max} from Eq. 4.19, the lander will collide with Mars's surface before the scheduled landing at t_f. When a collision is predicted, the final time can be adjusted to t_{max} to complete the landing mission without collision, at the expense of a higher performance index value.

When there is a constraint on either the control force or the control acceleration, adjusting the flight time is usually not sufficient. To overcome this defect, the waypoint-optimized ZEM/ZEV feedback scheme is proposed in the next section.

4.4 Waypoint Optimization for Power-Limited Engine

For a power-limited (or variable specific impulse) engine, the power for a given thrust is expressed as [41, 42, 43]

$$P = -\frac{1}{2}\dot{m}c^2 = \frac{1}{2}cT = -\frac{1}{2}\frac{m^2}{\dot{m}}|\mathbf{a}|^2 \leq P_{max} \tag{4.20}$$

where P_{max} is the maximum power available from the engine.

The performance index can be expressed in terms of the final mass of the spacecraft by integrating the above equation, as follows:

$$J = \frac{1}{2}\int_{t_0}^{t_f} |\mathbf{a}|^2 \,\mathrm{d}t \leq -P_{max}\int_{t_0}^{t_f} \frac{\dot{m}}{m^2}\,\mathrm{d}t = \frac{P_{max}}{m_f} - \frac{P_{max}}{m_0} \tag{4.21}$$

Equation 4.20 shows that, for a given acceleration command, $\mathbf{a}$, and current mass, m, a larger power P results in a smaller mass consumption rate $|\dot{m}|$. If the thruster always operates at P_{max}, all of the terms in Eq. 4.21 are equal, and minimizing J is equivalent to maximizing the final mass of the spacecraft m_f. The ZEM/ZEV algorithm discussed in this chapter is thus optimal in terms of mass consumption.

The ZEM/ZEV feedback guidance command for constant gravitational acceleration is a linear function of time. By dividing the mission time into two parts-from the beginning to the waypoint time t_m, and from the waypoint time to the end-the optimal accelerations can be expressed as

$$\mathbf{a} = \begin{cases} (t - t_0)\,\boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1 - \mathbf{g} & t \in (t_0, t_m) \\ (t_f - t)\,\boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1 - \mathbf{g} & t \in (t_m, t_f) \end{cases} \tag{4.22}$$

where $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, $\boldsymbol{\beta}_1$, and $\boldsymbol{\beta}_2$ are constant vectors to be found.

Substituting Eq. 4.22 into the equations of motion, the waypoint velocity and position can be found from successive integrations from t_0 to t_m as

$$\mathbf{v}_m = \frac{1}{2}t_1^2\boldsymbol{\alpha}_1 + t_1\boldsymbol{\beta}_1 + \mathbf{v}_0 \tag{4.23}$$

$$\mathbf{r}_m = \frac{1}{6}t_1^3\boldsymbol{\alpha}_1 + \frac{1}{2}t_1^2\boldsymbol{\beta}_1 + \mathbf{v}_0 t_1 + \mathbf{r}_0 \tag{4.24}$$

where time duration of the segment is defined as $t_1 = t_m - t_0$.

The waypoint velocity and position can be found similarly for the second segment as

$$\mathbf{v}_m = \frac{1}{2}t_2^2\boldsymbol{\alpha}_2 + t_2\boldsymbol{\beta}_2 + \mathbf{v}_f \tag{4.25}$$

$$\mathbf{r}_m = \frac{1}{6}t_2^3\boldsymbol{\alpha}_2 + \frac{1}{2}t_2^2\boldsymbol{\beta}_2 + \mathbf{v}_f t_2 + \mathbf{r}_f \tag{4.26}$$

where $t_2 = t_f - t_m$.

Combining Eqs. 4.23 through 4.26 leads to the following boundary conditions

$$\mathbf{v}_f - \mathbf{v}_0 = \frac{1}{2}t_1^2\boldsymbol{\alpha}_1 + \frac{1}{2}t_2^2\boldsymbol{\alpha}_2 + t_1\boldsymbol{\beta}_1 + t_2\boldsymbol{\beta}_t \tag{4.27}$$

$$\mathbf{v}_0 t_1 + \mathbf{v}_f t_2 - \mathbf{r}_f + \mathbf{r}_0 = -\frac{1}{6}t_1^3\boldsymbol{\alpha}_1 + \frac{1}{6}t_2^3\boldsymbol{\alpha}_2 - \frac{1}{2}t_1^2\boldsymbol{\beta}_1 + \frac{1}{2}t_2^2\boldsymbol{\beta}_2 \tag{4.28}$$

The performance index, defined as Eq. 4.5, can now be expressed in terms of the unknown parameters $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, $\boldsymbol{\beta}_1$, and $\boldsymbol{\beta}_2$ as

$$\begin{aligned} J &= \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \\ &= \frac{1}{2}\int_{t_0}^{t_m} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t + \frac{1}{2}\int_{t_m}^{t_f} \mathbf{a}^T\mathbf{a}\,\mathrm{d}t \\ &= \frac{1}{2}\Bigg[\frac{1}{3}\boldsymbol{\alpha}_1^T\boldsymbol{\alpha}_1 t_1^3 + \boldsymbol{\alpha}_1^T\left(\boldsymbol{\beta}_1 - \mathbf{g}\right)t_1^3 + \frac{1}{3}\boldsymbol{\alpha}_2^T\boldsymbol{\alpha}_2 t_2^3 + \boldsymbol{\alpha}_2^T\left(\boldsymbol{\beta}_2 - \mathbf{g}\right)t_2^3 + \left(\boldsymbol{\beta}_1 - \mathbf{g}\right)^T\left(\boldsymbol{\beta}_1 - \mathbf{g}\right)t_1 \\ &\qquad + \left(\boldsymbol{\beta}_2 - \mathbf{g}\right)^T\left(\boldsymbol{\beta}_2 - \mathbf{g}\right)t_2\Bigg] \end{aligned} \tag{4.29}$$

Define $\left[\boldsymbol{\alpha}_1^T\,\boldsymbol{\alpha}_2^T\,\boldsymbol{\beta}_1^T\,\boldsymbol{\beta}_2^T\right]^T$ as the vector of to-be-determined constants. The waypoint optimization problem can now be described as a standard quadratic programming problem, where the goal is to determine a vector $\mathbf{x}$ which minimizes the following function

$$J = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x} \tag{4.30}$$

where

$$\mathbf{H} = \begin{bmatrix} \frac{1}{3}t_1^3\mathbf{I}_3 & 0_3 & \frac{1}{2}t_1^2\mathbf{I}_3 & 0_3 \\ 0_3 & \frac{1}{3}t_2^3\mathbf{I}_3 & 0_3 & \frac{1}{2}t_2^2\mathbf{I}_3 \\ \frac{1}{2}t_1^2\mathbf{I}_3 & 0_3 & t_1\mathbf{I}_3 & 0_3 \\ 0_3 & \frac{1}{2}t_2^2\mathbf{I}_3 & 0_3 & t_2\mathbf{I}_3 \end{bmatrix}, \quad \mathbf{c} = -\begin{bmatrix} \frac{1}{2}t_1^2\mathbf{g} \\ \frac{1}{2}t_2^2\mathbf{g} \\ t_1\mathbf{g} \\ t_2\mathbf{g} \end{bmatrix}$$

subject to one or more constraints of the form

$$\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \tag{4.31}$$

$$\mathbf{A}_{ineq}\mathbf{x} \leq \mathbf{b}_{ineq} \tag{4.32}$$

where $\mathbf{I}_3$ is the 3x3 identity matrix, and 0_3 is the 3x3 zero matrix.

The boundary condition given by Eqs. 4.27 and 4.28 is an equality constraint, which can be cast in the standard form of Eq. 4.31 as

$$\mathbf{A}_{eq} = \begin{bmatrix} \frac{1}{2}t_1^2\mathbf{I}_3 & \frac{1}{2}t_2^2\mathbf{I}_3 & t_1\mathbf{I}_3 & t_2\mathbf{I}_3 \\ -\frac{1}{6}t_1^3\mathbf{I}_3 & \frac{1}{6}t_2^3\mathbf{I}_3 & -\frac{1}{2}t_1^2\mathbf{I}_3 & \frac{1}{2}t_2^2\mathbf{I}_3 \end{bmatrix}$$

$$\mathbf{b}_{eq} = \begin{bmatrix} \mathbf{v}_f - \mathbf{v}_0 \\ \mathbf{v}_0 t_1 + \mathbf{v}_f t_2 - \mathbf{r}_f + \mathbf{r}_0 \end{bmatrix}$$

Altitude constraints, such as the no-subsurface flight constraint, or a glide-slope constraint, are inequality constraints. For brevity, in this example we look in detail at only the no-subsurface flight constraint. Other inequality constraints can be handled similarly.

Let us divide the two mission segments into N_1 and N_2 equal intervals, respectively. Let $a_i = (i \cdot t_1)/N_1$, $(i = 1, 2, \ldots, N_1)$ denote the duration of each interval in the first mission segment, and $b_i = (t_f - j \cdot t_2)/N_2$, $(j = 1, 2, \ldots, N_2 - 1)$ denote the duration of each interval in the second mission segment. The no-subsurface flight constraint is thus described by

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \left(\frac{1}{6} a_i^3 \boldsymbol{\alpha}_1 + \frac{1}{2} a_i^2 \boldsymbol{\beta}_1 + a_i \mathbf{v}_0 + \mathbf{r}_0 \right) \geq \quad i = 1, 2, \ldots, N_1 \tag{4.33}$$

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \left(\frac{1}{6} b_j^3 \boldsymbol{\alpha}_2 + \frac{1}{2} b_j^2 \boldsymbol{\beta}_2 - b_i \mathbf{v}_f + \mathbf{r}_f \right) \geq \quad j = 1, 2, \ldots, N_2 - 1 \tag{4.34}$$

Eqs. 4.33 and 4.34 describe a total of $N_1 + N_2 - 1$ inequality constraints. These constraints can be cast in the standard form of Eq. 4.32 as

$$\mathbf{A}_{ineq} = \begin{bmatrix} 0 & \frac{1}{6}a_1^3 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}a_1^2 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \frac{1}{6}a_{N_1}^3 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}a_{N_1}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{6}b_1^3 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}b_1^2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \frac{1}{6}b_{N_2-1}^3 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}b_{N_2-1}^2 & 0 \end{bmatrix}_{(N_1+N_2-1)\cdot 12}$$

$$\mathbf{b}_{ineq} = \begin{bmatrix} y_0 + a_1\dot{y}_0 & \ldots & y_0 + a_{N_1}\dot{y}_0 & y_f - \dot{y}_f b_1 & \ldots & y_f - \dot{y}_f b_{N_2} \end{bmatrix}^T_{(N_1+N_2-1)}$$

The minimum altitude constraints, as well as glide slope constraint, etc., can similarly be expressed in the form of Eq. 4.32. Details are omitted here.

A number of commercially available programs can solve this quadratic programming problem. Once the minimizing vector is found, the waypoint velocity and position can be found from either Eqs. 4.23 and 4.24, or Eqs. 4.25 and 4.26. This approach can be extended to include multiple waypoints, with each additional waypoint increasing the size of the unknown vector.

4.5 Waypoint Optimization for Thrust-Limited Engine

For a spacecraft equipped with a thrust-limited engine, the engine exhaust velocity is fixed at a constant value. The thrust magnitude is constrained as

$$0 \leq T \leq T_{max} \tag{4.35}$$

where T_{max} is the maximum available thrust.

Considering control saturation, the ZEM/ZEV algorithm, given by Eq. 4.12, becomes

$$\mathbf{a} = \underset{T_{max}/m}{\text{sat}} \left(\frac{6}{t_{go}^2} \mathbf{ZEM} - \frac{2}{t_{go}} \mathbf{ZEV} \right) \tag{4.36}$$

where the normalized saturation function of a vector $\mathbf{q}$ is defined as

$$\underset{U}{\text{sat}} \left(\mathbf{q} \right) = \begin{cases} \mathbf{q} & \text{if } |\mathbf{q}| \leq U \\ \mathbf{q} \frac{U}{|\mathbf{q}|} & \text{if } |\mathbf{q}| > U \end{cases} \tag{4.37}$$

The normalized saturation function, given by Eq. 4.37, is non-differentiable at the critical saturation points. Typical optimization packages, such as GPOPS, can make use of analytical partial derivatives of the constraints with respect to the states and controls. Supplying these derivatives can greatly improve the computational efficiency and accuracy of the optimization problem. The following continuously differentiable function approximates the saturation function

$$\underset{U}{\text{sat}} \left(\mathbf{q} \right) = \begin{cases} \mathbf{q} & \text{if } \frac{U}{|\mathbf{q}|} > n_U \\ \Phi \left(\mathbf{q}, U \right) \cdot \mathbf{q} & \text{if } \frac{U}{|\mathbf{q}|} \in [n_L, n_U] \\ \frac{U}{|\mathbf{q}|} \cdot \mathbf{q} & \text{if } \frac{U}{|\mathbf{q}|} < n_L \end{cases} \tag{4.38}$$

where the function Φ is a second-order spline function, whose coefficients depend on the endpoints of the interval. The modified saturation function becomes the normalized saturation function when the interval (n_L, n_U) shrinks to zero. For this study, the interval is chosen as $(0.9, 1.1)$. Then we have

$$\Phi \left(\mathbf{q}, U \right) = -2.5 \left(\frac{U}{|\mathbf{q}|} \right)^2 + 5.5 \frac{U}{|\mathbf{q}|} - 2.025 \tag{4.39}$$

$$\frac{\partial \underset{U}{\text{sat}}\left(\mathbf{q}\right)}{\partial \mathbf{q}} = \begin{cases} \mathbf{I}_3 & \text{if } \frac{U}{|\mathbf{q}|} > 1.1 \\ \Phi\left(\mathbf{q}, U\right) \cdot \mathbf{I}_3 + \frac{U}{|\mathbf{q}|}\left(5\frac{U}{\mathbf{q}|} - 5.5\right) \cdot \frac{\mathbf{q}\mathbf{q}^T}{|\mathbf{q}|^2} & \text{if } \frac{U}{|\mathbf{q}|} \in [0.9, 1.1] \\ \frac{U}{|\mathbf{q}|}\left(\mathbf{I}_3 - \frac{\mathbf{q}\mathbf{q}^T}{|\mathbf{q}|^2}\right) & \text{if } \frac{U}{|\mathbf{q}|} < 0.9 \end{cases} \tag{4.40}$$

where the continuity of the above equation can be easily verified.

The waypoint optimization problem for a thrust-limited engine becomes:

Determine the optimal waypoint ($\mathbf{r}_m$ and $\mathbf{v}_m$) to connect the following two phases

$$\begin{aligned} &\text{Phase 1} \quad (t_0 \le t \le t_m) \begin{cases} \mathbf{r}\left(t_0\right) = \mathbf{r}_0 & \mathbf{r}\left(t_m\right) = \mathbf{r}_m = \text{free} \\ \mathbf{v}\left(t_0\right) = \mathbf{v}_0 & \mathbf{v}\left(t_m\right) = \mathbf{v}_m = \text{free} \end{cases} \\ &\text{Phase 2} \quad (t_m \le t \le t_f) \begin{cases} \mathbf{r}\left(t_m\right) = \mathbf{r}_m = \text{free} & \mathbf{r}\left(t_f\right) = \mathbf{r}_f \\ \mathbf{v}\left(t_m\right) = \mathbf{v}_m = \text{free} & \mathbf{v}\left(t_f\right) = \mathbf{v}_f \end{cases} \end{aligned} \tag{4.41}$$

where each phase is governed by Eqs. 4.1 through 4.4 , and the acceleration command is assumed to have the following form:

$$\begin{aligned} &\text{Phase 1} \quad (t_0 \le t \le t_m) \quad \mathbf{a} = \underset{T_{max}/m}{\text{sat}}\left(\mathbf{a}_{c1}\right) \\ &\text{Phase 2} \quad (t_m \le t \le t_f) \quad \mathbf{a} = \underset{T_{max}/m}{\text{sat}}\left(\mathbf{a}_{c2}\right) \end{aligned} \tag{4.42}$$

where

$$\begin{aligned} \mathbf{a}_{c1} &= \frac{6\left(\mathbf{r}_m - \mathbf{r}\right)}{\left(t_m - t\right)^2} - \frac{2\mathbf{v}_m + 4\mathbf{v}}{t_m - t} - \mathbf{g} \\ \mathbf{a}_{c2} &= -\frac{6\mathbf{r}}{\left(t_f - t\right)^2} - \frac{4\mathbf{v}}{t_f - t} - \mathbf{g} \end{aligned}$$

and the following path constraints are satisfied during the two phases

$$\begin{aligned} &\text{Phase 1} \quad (t_0 \le t \le t_m) \quad \boldsymbol{\sigma}_{c1} \equiv \mathbf{a} - \underset{T_{max}/m}{\text{sat}}\left(\mathbf{a}_{c1}\right) \\ &\text{Phase 2} \quad (t_m \le t \le t_f) \quad \boldsymbol{\sigma}_{c2} \equiv \mathbf{a} - \underset{T_{max}/m}{\text{sat}}\left(\mathbf{a}_{c2}\right) \end{aligned} \tag{4.43}$$

where the path constraints, $\boldsymbol{\sigma}_{c1}$ and $\boldsymbol{\sigma}c2$, must be zero during the entire phase.

The derivatives of the dynamic equations and the cost function are straightforward to find. Note that the path constraints in the two phases are the same. The partial derivatives of the path constraints will be given for phase 1. The partial derivatives for phase 2 proceed similarly.

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial \mathbf{a}} = \mathbf{I}_3 \tag{4.44}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial \mathbf{r}} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial \mathbf{r}} = \frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{6}{(t_m - t)^2}\mathbf{I}_3 \tag{4.45}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial \mathbf{r}_m} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial \mathbf{r}_m} = \frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{-6}{(t_m - t)^2}\mathbf{I}_3 \tag{4.46}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial \mathbf{v}} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial \mathbf{v}} = \frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{4}{t_m - t}\mathbf{I}_3 \tag{4.47}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial \mathbf{v}_m} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial \mathbf{v}_m} = \frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{2}{t_m - t}\mathbf{I}_3 \tag{4.48}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial t} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial t} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \left(\frac{12(\mathbf{r}_m - \mathbf{r})}{(t_m - t)^3} - \frac{2\mathbf{v}_m + 4\mathbf{v}}{(t_m - t)^2}\right) \tag{4.49}$$

$$\frac{\partial \boldsymbol{\sigma}_{c1}}{\partial t_m} = -\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \frac{\partial \mathbf{a}_1}{\partial t_m} =-\frac{\partial \underset{T_{max}/m}{\mathrm{sat}}(\mathbf{a}_1)}{\partial \mathbf{a}_1} \cdot \left(-\frac{12(\mathbf{r}_m - \mathbf{r})}{(t_m - t)^3} + \frac{2\mathbf{v}_m + 4\mathbf{v}}{(t_m - t)^2}\right) \tag{4.50}$$

The partial derivative of the path constraint with respect to mass can be obtained by referring to Eq. 4.40. It is omitted here due to the lengthiness of the expression.

More waypoints can be obtained by considering more phases in GPOPS, and increasing the number of waypoints would improve the optimality and decrease the difference with the open-loop optimal solution, but more processing time would be required.

4.6 Waypoint-Optimized ZEM/ZEV Scheme

The complete procedure for conducting a Mars pinpoint landing mission using the waypoint-optimized ZEM/ZEV feedback control scheme is proposed as follows:

1. At the end of the subsonic parachute entry phase, determine a suitable landing site.

2. Using the position and velocity of the lander, solve Eq. 4.15 to obtain the optimal flight time t_f.

3. For a power-limited engine, t_f is the optimal flight time, which minimizes both J and Δm. Numerically propagate the dynamic system with the ZEM/ZEV algorithm to generate the optimal trajectory. For a thrust-limited engine, t_f can be considered a reference time. A line search can be implemented to determine the optimal t_f for either J or Δm.

4. If any part of the optimal trajectory violates an altitude constraint, go to step 5. Otherwise, go to step 6.

5. Use the optimization approach discussed in this chapter to determine a waypoint to satisfy the altitude constraint. Consider the optimized waypoint as an intermediate target, using the ZEV/ZEM algorithm to reach this waypoint.

6. Finally, use the ZEM/ZEV algorithm to land the vehicle at the designated landing site.

The waypoint optimization step 5 requires a certain amount of computational time if t_m and/or t_f is free. Because this guidance scheme is intended for real-time use, it is not advisable to leave the times as completely free parameters. The process can be made much more efficient if both of them are restricted to narrower ranges. The major challenge in a practical mission is how to quickly find these times for step 5. Based on our experiences with a wide variety of numerical simulations, a reasonable method is to simply use the same t_f found in step 3, and choosing the flight time corresponding to the lowest altitude on the optimal trajectory for t_m.

Even though there is no strict guarantee of optimality, this selection has provided good performance in our simulations, close to the performance with free waypoint and final times. In a practical mission, the spacecraft is descending after the end of the parachute entry phase, so a quick solution is desirable. The waypoint and final times are easily found from previous steps, requiring minimal computation.

Both off-line and on-line waypoint optimization schemes have potential for practical implementation. The conventional off-line optimal solution approach stores the entire profile on board, and loses optimality if the lander deviates from the intended path. The waypoint method only requires a single point to be stored, and due to the feedback nature of the ZEM/ZEV algorithm, it is robust against disturbances. For online autonomous application, the vehicle is controlled by the ZEM/ZEV algorithm with t_f from step 3. If a waypoint is needed, the ZEM/ZEV algorithm switches over to the waypoint-optimized ZEM/ZEV algorithm.

4.7 Numerical Simulation Examples

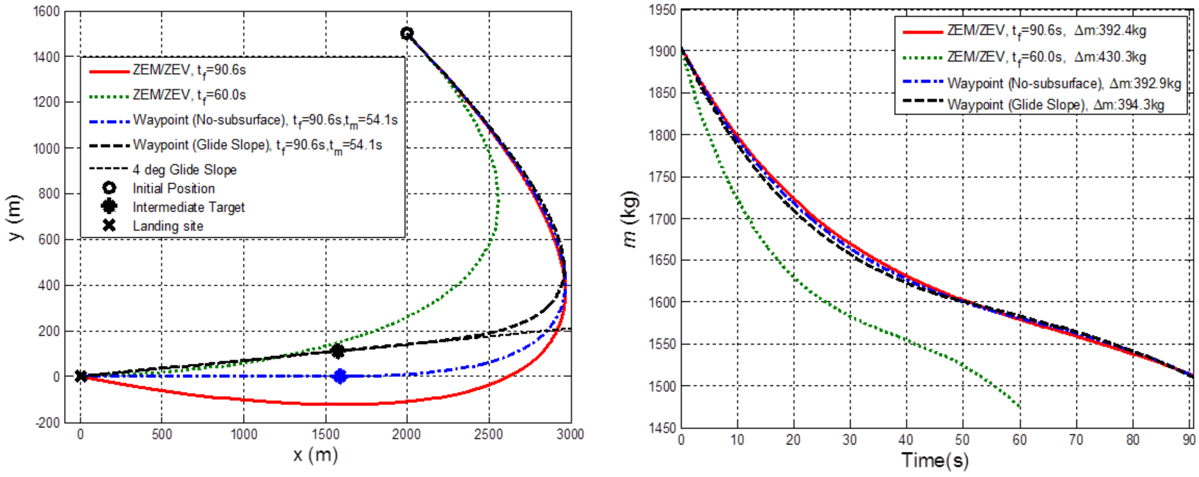
A Mars landing example of [26] was examined for both a power-limited engine and a thrust-limited engine. In addition to the ZEM/ZEV feedback guidance algorithms, open-loop optimal solutions were generated with GPOPS for comparison. A spacecraft similar to the one described in [26] is used, with an initial mass of 1905 kg. The initial conditions are given as $\mathbf{r}_0 = (2, 1.5, 0)\,\mathrm{km}$, $\mathbf{v}_0 = (100, -75, 0)\,\mathrm{m/s}$. The landing site is at the origin of the reference frame, and for a soft landing the final velocity is zero. The final conditions are thus $\mathbf{r}_f = (0, 0, 0)\,\mathrm{km}$, $\mathbf{v}_f = (0, 0, 0)\,\mathrm{m/s}$. The constant gravitational acceleration is assumed as $\mathbf{g} = (0, -3.7114, 0)\,\mathrm{m/s\hat{}2}$. It is assumed that all system states are available without any measurement error.

4.7.1 Mars Landing with Power-Limited Engine

The power-limited engine may not be available for an actual Mars landing mission; however, it is considered here for a couple of reasons. First, our main research interest is in planetary landing, so this is a way to look at different engine dynamics within our motivating example. Second, as described previously, fuel usage can be directly minimized, unlike with the thrust-limited engine. This happens when the engine operates at the maximum power level, therefore for these simulations the engine is assumed to always operate at the maximum power level. The quadratic programming problem for waypoint optimization is solved using the MATLAB *quadprog* function with default options. In all test cases the required processing time is less than 0.1 s.

Figure 4.1 shows the optimal trajectories and corresponding fuel usage histories for the standard ZEM/ZEV algorithm, and for the ZEM/ZEV waypoint method with altitude constraints. From step 2 of the landing procedure, the optimal t_f is found to be 90.6 s. Checking Eq. 4.19 reveals that the upper bound on flight time is 60 s. The solid-line trajectory in Fig. 4.1 shows that the optimal trajectory for a 90.6 s flight indeed violates the no-subsurface constraint. By adjusting the t_f to 60 s, the ZEM/ZEV algorithm produces the dotted-line trajectory in Fig. 4.1, which does not violate the constraint. However, the mass usage comparison in Fig. 4.1

shows that the mission with adjusted flight time uses significantly more fuel.



(a) Vehicle Trajectory.

(b) Vehicle Mass.

Figure 4.1 Comparisons of various forms of the ZEM/ZEV algorithm (power-limited engine).

The waypoint-optimized ZEM/ZEV algorithm can be employed to improve the performance of the mission, while still satisfying the altitude constraints. Following the suggestion for selecting proper waypoint time and final time, the mission time t_f is kept as 90.6 s. From the simulation of the trajectory for the basic ZEM/ZEV algorithm with a 90.6 s mission time, the time corresponding to the minimum altitude is 54.1 s. This is chosen for the waypoint time t_m.

Two types of altitude constraint are considered for the waypoint-optimized ZEM/ZEV method. First is the simple no-subsurface flight constraint, shown as the dash-dot line in Fig. 4.1. A slightly more sophisticated constraint, keeping the glide slope above 4 degrees, is shown by the dashed line. Note that such a constraint automatically satisfies the no-subsurface flight constraint. Figure 4.1 shows that the fuel usage for both waypoint schemes is nearly identical to the basic optimal ZEM/ZEV algorithm.

Figure 4.2 shows engine performance histories for the case with one waypoint and a no-subsurface flight constraint, shown as the dotted blue line in Fig. 4.1. The dotted line in this figure shows the power-limited engine case; the thrust-limited case will be discussed later. The optimal constant power level can be seen, as well as gently changing curves for control magnitude and mass consumption rate. The changing exhaust velocity required for the power-limited engine is also shown.

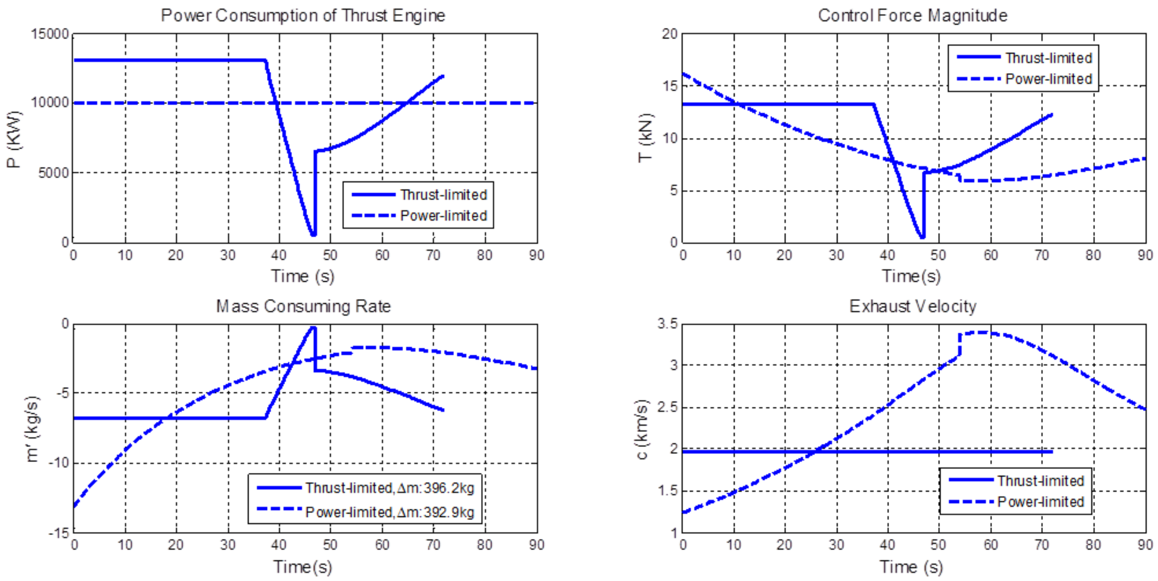


Figure 4.2 Engine performance histories for waypoint-optimized Mars landing.

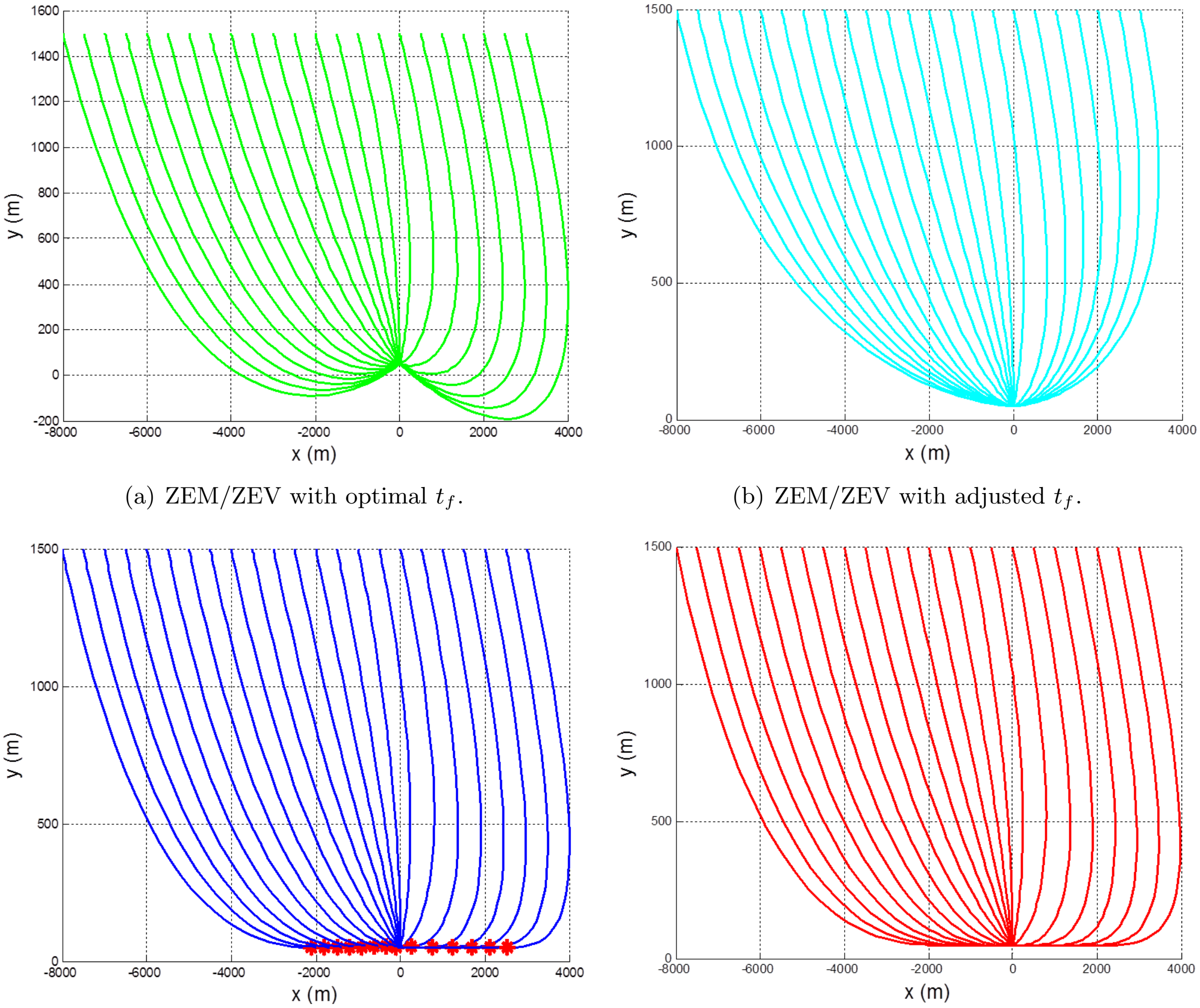
The waypoint-optimized ZEM/ZEV scheme is intended to be used on-line, finding the optimal waypoint in real time for a range of possible initial conditions. To demonstrate the applicability of the method for different initial conditions, a range of initial conditions are simulated. The initial velocity and altitude are held constant, while the initial horizontal position ranges from -8 km to 3 km. A 50-m minimum altitude is imposed to avoid collision with surface hazards, making the final position $\mathbf{r}_f = (0, 50, 0)\,\mathrm{m}$. From here the lander can descend straight down onto the landing site. Figure 4.3 shows the optimal trajectories for a variety of initial conditions. Three different ZEM/ZEV methods are shown, as well as the open-loop optimal solution. The four control schemes are:

- *ZEM/ZEV with optimal* t_f (Fig. 4.7.1): the standard ZEM/ZEV algorithm is employed, using the calculated optimal time, with no constraints. For many of the initial positions considered, the optimal trajectory passes through the surface, even with the 50 m "cushion."

- *ZEM/ZEV with adjusted* t_f (Fig. 4.7.1): the standard ZEM/ZEV algorithm is again used, but the upper bound on flight time is enforced for cases that would violate the no-subsurface flight constraint. All of the trajectories for this case stay above the surface.

- *Waypoint-optimized ZEM/ZEV* (Fig. 4.7.1): the quadratic programming problem is solved to determine the optimal waypoint, with t_f and t_m determined following the suggested procedure. The waypoints for each trajectory are shown as asterisks. All trajectories again stay above the surface. Other altitude constraints can be added by modifying the inequality constraints.

- *Open-loop optimal* (Fig. 4.7.1): The optimal solution, found by GPOPS, and satisfying the altitude constraint. These optimal trajectories look nearly identical to the waypoint-optimized ZEM/ZEV trajectories.



(a) ZEM/ZEV with optimal t_f.

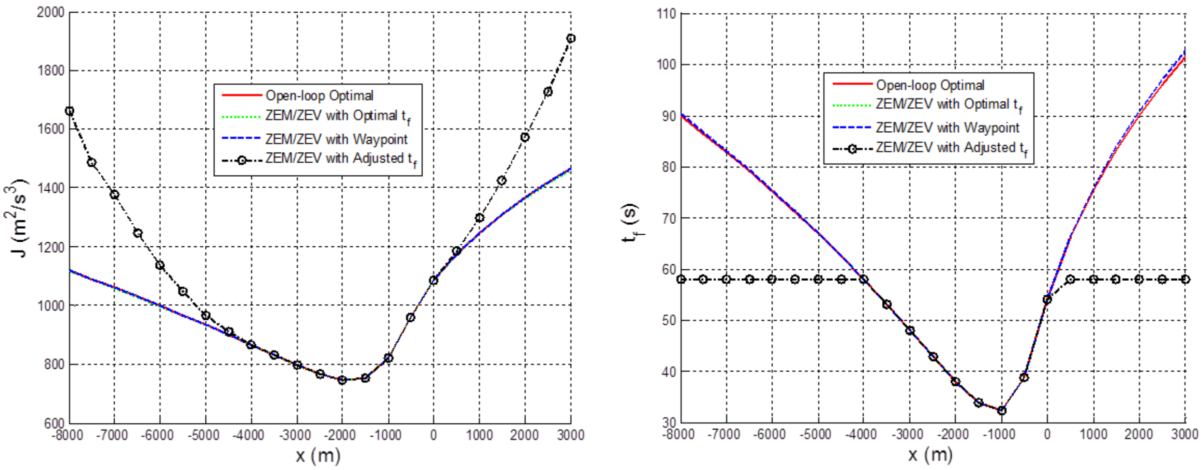
(b) ZEM/ZEV with adjusted t_f.

(c) Waypoint optimized ZEM/ZEV.

(d) Open-loop optimal.

Figure 4.3 Vehicle trajectories using various methods (power-limited engine).

For some initial conditions, the basic ZEM/ZEV algorithm works without violating the altitude constraint. For other cases, either the mission time needs to be adjusted, or a waypoint must be added. The performance of the various control schemes must ultimately be evaluated by comparing either the performance index J or the fuel usage Δm. Figure 4.4 shows the performance of each of the four algorithms in terms of J. The trends in Δm are similar and are not shown here. The mission time t_f is also shown. For initial conditions with horizontal range approximately between -4.5 km and 0.5 km, the lander starts out "headed in the right direction," and all four schemes show almost identical performance. Outside of this range, the effectiveness of the waypoint-optimized ZEM/ZEV algorithm is apparent. For these cases, the scheme with adjusted t_f uses significantly more fuel than the optimal solution. ZEM/ZEV with optimal t_f performs nearly as well as the optimal solution, but for these cases the no-subsurface flight condition is violated, resulting in mission failure. The waypoint-optimized ZEM/ZEV algorithm achieves near-optimal performance while satisfying the constraint. In other words, for a variety of initial conditions, the proposed waypoint scheme achieves almost identical performance with the open-loop optimal solution.



(a) Performance index J (m^2/s^3).

(b) Total flight time t_f (s).

Figure 4.4 Performance comparison of ZEM/ZEV and open-loop methods (power-limited engine).

4.7.2 Mars Landing with Thrust-Limited Engine

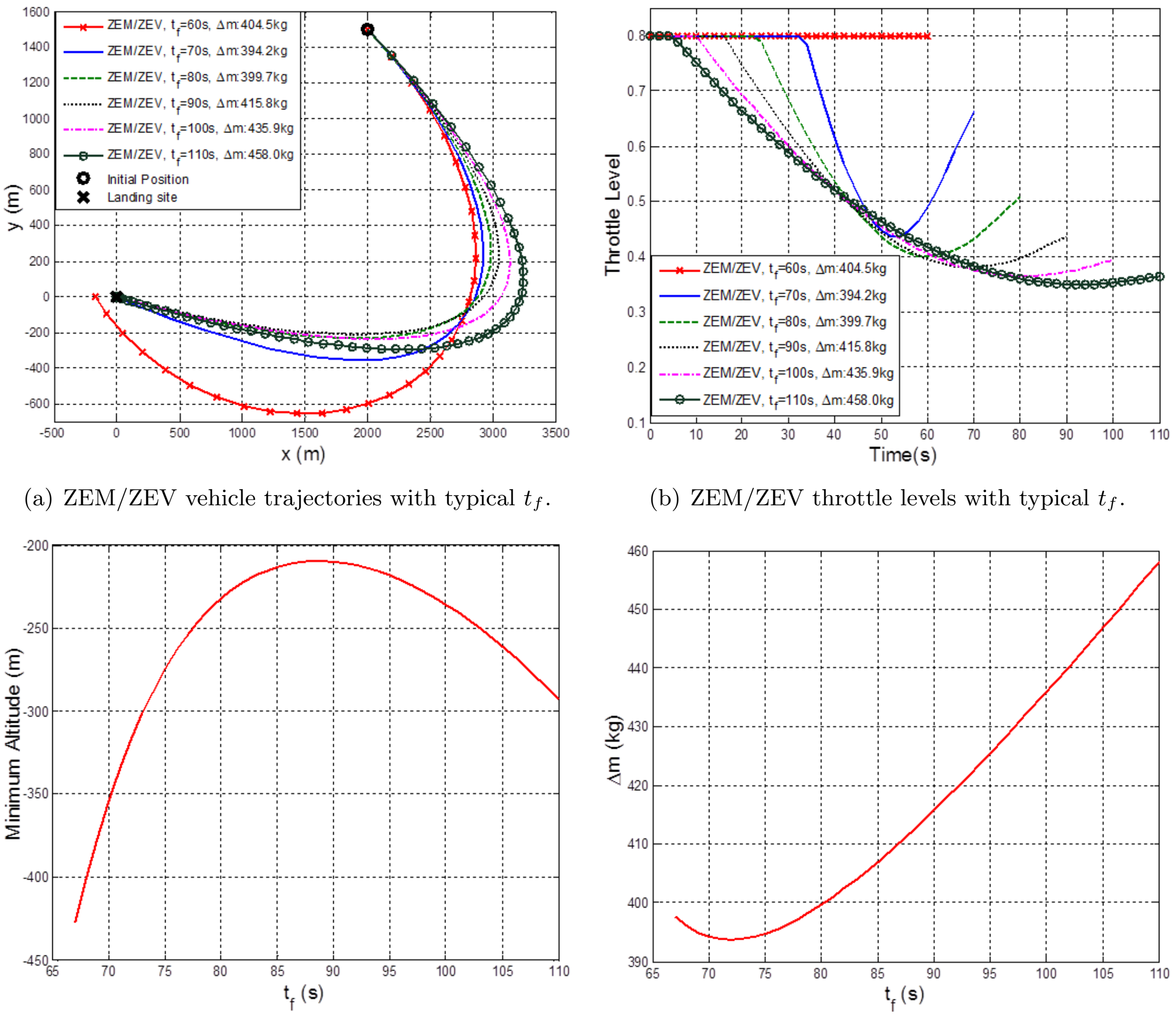
Unlike for a power-limited engine, for a thrust-limited engine, minimizing J is not equivalent to minimizing Δv. Furthermore, due to the nonlinear constraint on control acceleration, the optimal flight time found from step 2 does not guarantee an optimal J. Although the proposed ZEM/ZEV approach is not intended to be fuel-optimal, we will consider fuel usage to be the performance measure of interest, because it is of interest in any practical mission.

For the simulations shown, the spacecraft model is similar to that discussed in [26]. The maximum control force from the thrusters is assumed to be 16.753 kN, of which only 80% is available for control (leaving a conservative margin). The engine exhaust velocity, c, is fixed at 1.964 km/s ($1/c = 5.09\text{x}10^{-4} s/m$).

Equation 4.19, used to determine the upper bound of t_f, does not take into account saturation constraints. The method of adjusting t_f does not work well in general. Typically, t_f is decreased to force a "straight shot" and avoid violating the no-subsurface flight constraint. However, fulfilling the same mission in less time usually requires higher acceleration levels during the mission. Adjusting the flight time downward to avoid collision thus results in control saturation.

Figure 4.5 shows vehicle trajectories and throttle level histories for various mission times for the ZEM/ZEV algorithm, and corresponding minimum altitudes and fuel usage. When the flight time is less than 67 s, the controls are saturated the entire time, and the terminal requirements are not met. By increasing the flight time above 67 s, the vehicle can successfully meet the terminal requirements, but the no-subsurface constraint is violated. For all times tested, the minimum altitude is below the surface. Fuel usage increases with increasing flight time, but even setting aside concerns on fuel usage, the landing mission cannot be accomplished solely by adjusting t_f.

Figure 4.6 shows a family of fuel-optimal solutions for a vehicle with a thrust-limited engine. For these initial conditions, the initial velocity and altitude are held constant, and the initial horizontal range is varied. These results can be obtained either by directly using TOMLAB optimization software, or by solving a second-order cone problem using the SeDuMi software



(a) ZEM/ZEV vehicle trajectories with typical t_f.

(b) ZEM/ZEV throttle levels with typical t_f.

(c) Minimum altitude for various t_f.

(d) Fuel usage Δm for various t_f.

Figure 4.5 Performance of ZEM/ZEV algorithm with various flights times (thrust-limited engine).

package. [26] GPOPS can also be used to find these results, but for this particular problem GPOPS requires more computational time. From the family of trajectories shown, it is seen that when the landing site is ahead of the vehicle, the fuel-optimal trajectories converge inside an envelope before landing. When the landing site is behind the vehicle's initial position, the optimal trajectory turns around, then touches the surface, before turning upwards and finally landing. These trajectories also converge on an envelope on the final approach.

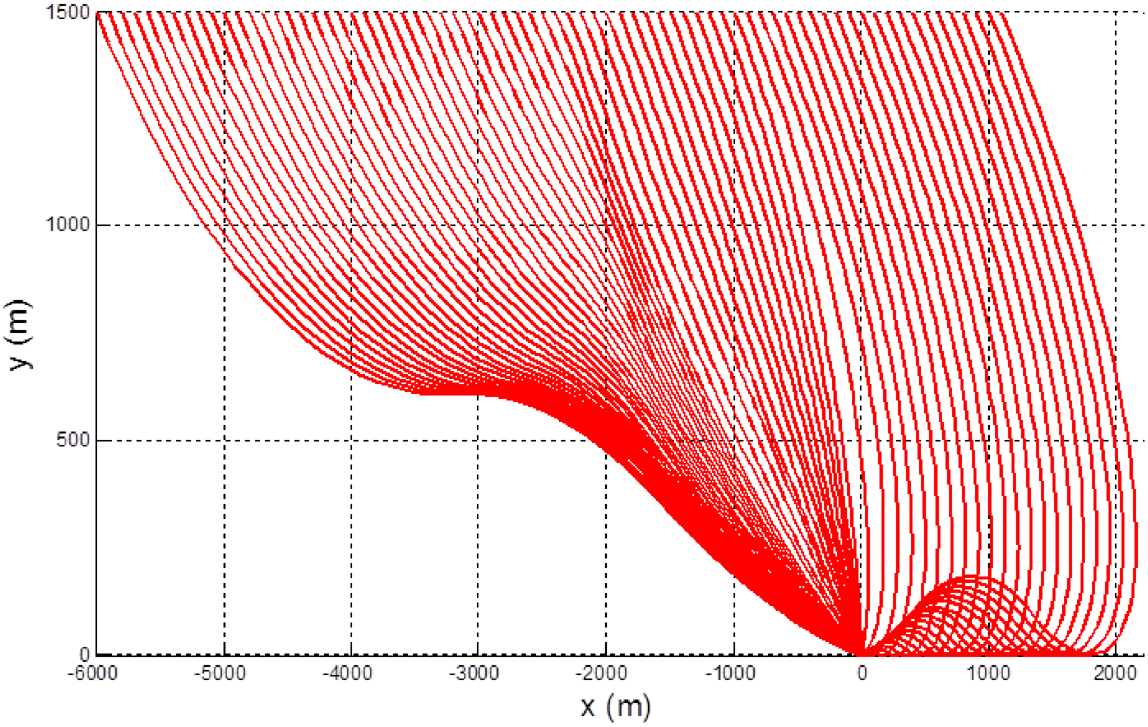
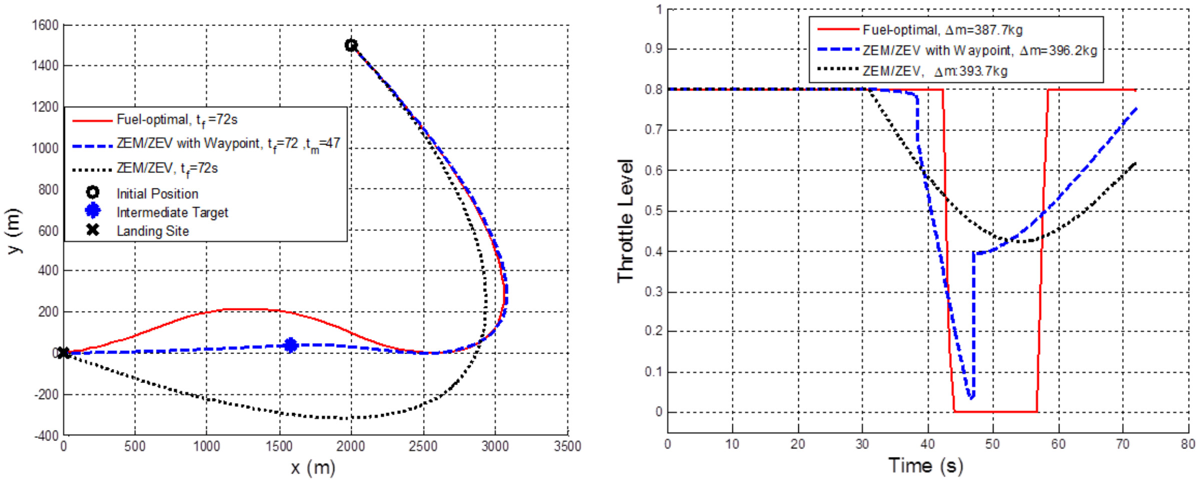


Figure 4.6 Fuel-optimal vehicle trajectories (thrust-limited engine).

The waypoint optimization scheme was examined for the case where simply adjusting t_f fails. From Fig. 5, it can be seen that the optimal t_f is 72 s. The corresponding time of minimum altitude is 47 s, chosen for t_m. Figure 4.7 shows the vehicle trajectories and throttle histories for the standard ZEM/ZEV algorithm, the ZEM/ZEV waypoint method, and the open-loop optimal solution. By using GPOPS to find an optimized waypoint, the ZEM/ZEV waypoint method successfully completes the landing mission in the presence of control saturation and a no-subsurface flight constraint. The fuel-optimal throttle shows the classical bang-off-bang form, resulting in fuel usage of 387.7 kg. The ZEM/ZEV waypoint algorithm uses 396.2 kg, only

2.2% more. Figure 4.2, previously shown, also provides detailed engine performance histories for the case with one waypoint, shown as a dotted line in Fig. 4.7. The piecewise nature of the control, with a jump at the waypoint, is easily seen.



(a) Vehicle trajectory.

(b) Throttle level.

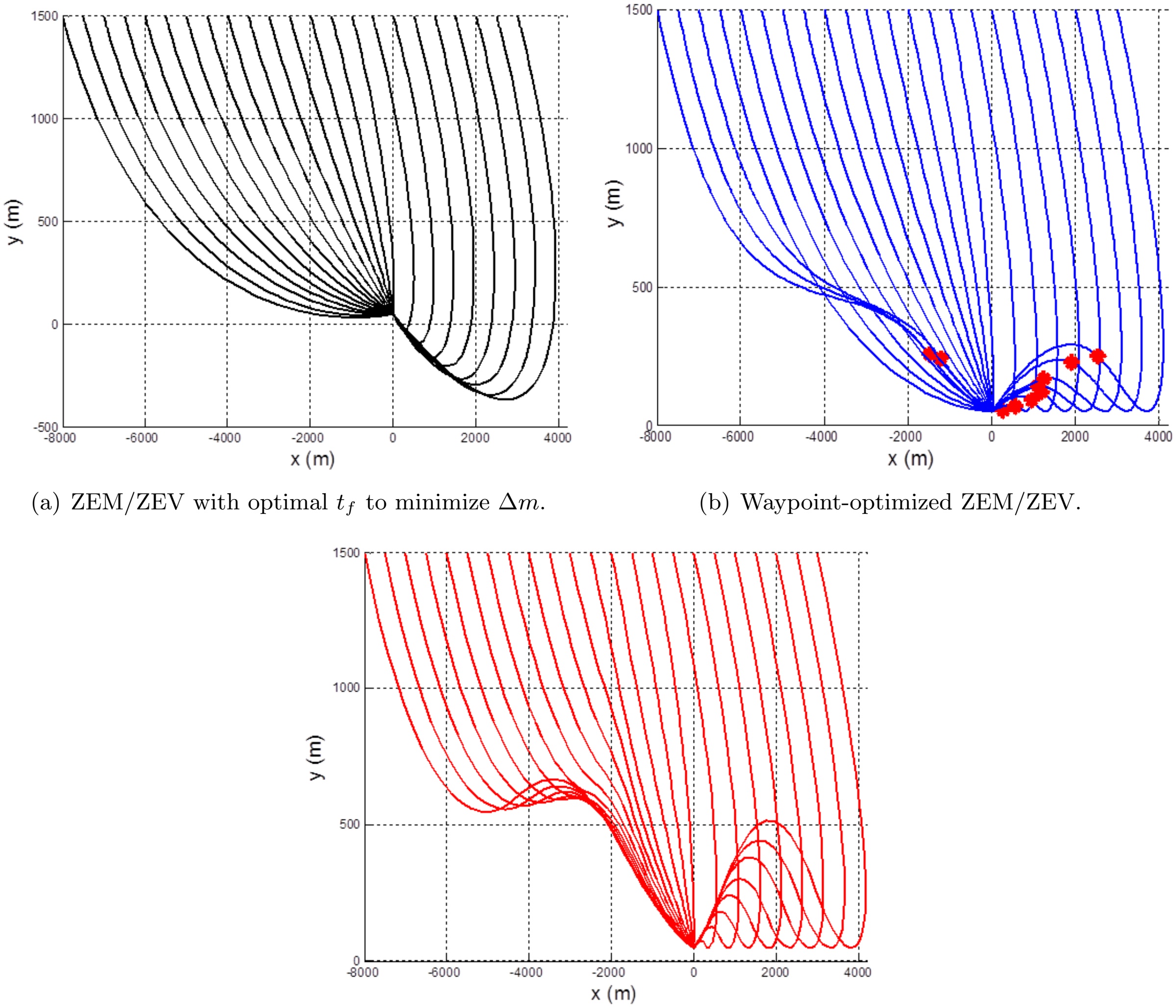
Figure 4.7 Comparison of ZEM/ZEV algorithm and fuel-optimal solution (thrust-limited engine).

In order to demonstrate the effectiveness of the proposed ZEM/ZEV waypoint strategy for a range of initial conditions, the initial position was varied, as in the last section. The initial velocity and altitude are held constant, while the initial horizontal position ranges from -8 km to 3 km. A 50-m minimum altitude is imposed to avoid collision with surface hazards, making the final position $\mathbf{r}_f = (0, 50, 0)\, m$. From here the lander can descend straight down onto the landing site. Figure 8 shows the optimal trajectories for a variety of initial conditions. Two ZEM/ZEV methods are shown, as well as the open-loop optimal solution. The three control schemes are:

- *ZEM/ZEV with optimal t_f to minimize Δm* (Fig. 4.7.2): The dynamic system is numerically propagated using the ZEM/ZEV algorithm. A Newton iteration method is used to find the t_f which minimizes Δm. For initial horizontal ranges from -8 km to -1 km successfully complete the mission, while initial ranges larger than -1 km violate the no-subsurface flight constraint.

- *Waypoint-optimized ZEM/ZEV* (Fig. 4.7.2): Fuel-optimal waypoints are found using GPOPS. Analytical derivatives are provided to GPOPS, utilizing the differentiable saturation function in Eq. 4.37. The waypoints are shown as asterisks. The mission can be successfully carried out for all of the initial conditions shown.

- *Open-loop fuel optimal* (Fig. 4.7.2): The fuel-optimal results, similar to those shown in Fig. 4.7.1, are generated by using either TOMLAB or SeDuMi.



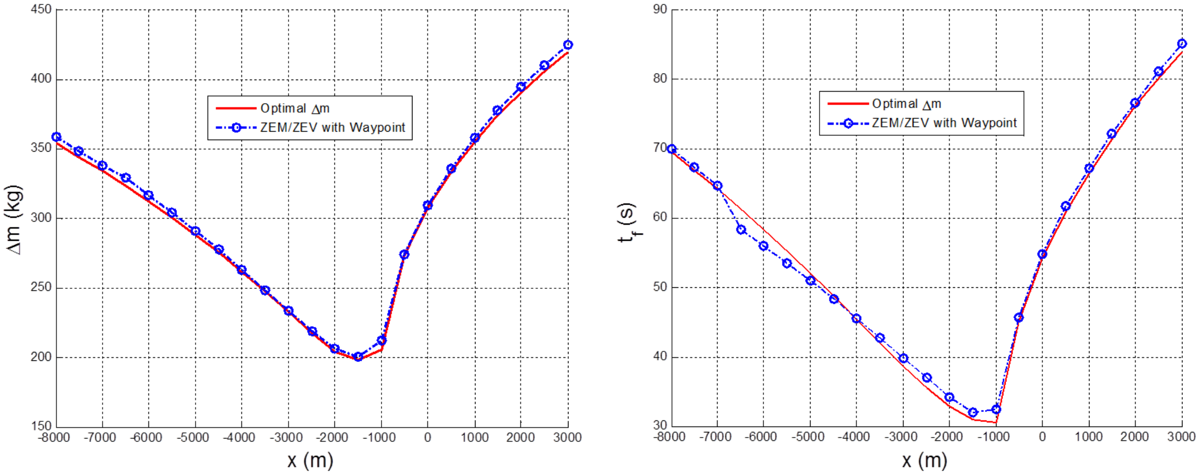
(a) ZEM/ZEV with optimal t_f to minimize Δm.

(b) Waypoint-optimized ZEM/ZEV.

(c) Open-loop fuel-optimal.

Figure 4.8 ZEM/ZEV and open-loop trajectories for minimizing Δm (thrust-limited engine).

The waypoint-optimized scheme results in trajectories that are qualitatively similar overall to the open-loop optimal trajectories, showing that adding one waypoint can change an infeasible ZEM/ZEV mission to a nearly optimal one. Figure 4.9 shows the fuel used and the total flight time for the various initial positions. The fuel usage from the waypoint method tracks very closely to the optimal, with slight variations at either extreme. The total mission times are also very close, showing that the waypoint method does approximate the open-loop optimal solution.



(a) Performance index Δm.

(b) Total flight time.

Figure 4.9 Fuel-optimal solution and waypoint-optimized ZEM/ZEV (thrust-limited engine)..

Based on the numerical tests shown here, the optimal waypoint can usually be found within 5-15 s when t_f and t_m are fixed. When those times are free, the optimization takes longer, and depends on the initial guess and range for each. Finding the optimal solution for a broad range of times can take up to a couple of minutes. For this reason, it is recommended that a fixed t_f and t_m are used.

4.8 Concluding Remarks

This chapter investigated the implementation of ZEM/ZEV feedback guidance scheme for the powered descent phase of a Mars pinpoint landing mission. Two types of engine, power-limited and thrust-limited, were considered.

When a collision with the surface of Mars was predicted for the standard ZEM/ZEV algorithm, an intermediate waypoint is chosen using a numerical optimization program. The lander uses ZEM/ZEV feedback guidance to reach the waypoint, then uses ZEM/ZEV guidance to reach the landing site. For the power-limited engine, the control acceleration is not directly constrained, and so only a minimum altitude constraint is considered. The waypoint optimization problem can be formulated as a standard quadratic programming problem. Several numerical optimization software packages can be used to solve this problem.

For the thrust-limited engine, a non-linear saturation constraint is considered, in addition to the no-subsurface flight constraint. The GPOPS package was used to find the optimal waypoint. A continuously differentiable approximation to the standard saturation function is introduced to provide analytical partial derivatives of various constraints with respect to states and controls, these derivatives are passed to GPOPS to speed up optimization process.

Numerical simulations for a few illustrative examples were performed. The ZEM/ZEV algorithms described in this chapter were used, and the corresponding open-loop optimal solutions were found for comparison. Results show that constraints cannot be consistently satisfied solely by changing flight time. Instead, an optimized waypoint is used as an intermediate target for the ZEM/ZEV scheme. Comparison with the open-loop optimal solution shows that the waypoint method achieves near-optimal performance, while maintaining the desirable robustness properties of a feedback controller.

The waypoint-optimized ZEM/ZEV feedback guidance algorithm is more suitable for autonomous implementation than any purely open-loop method. The scheme requires only one waypoint to be stored, and the feedback nature is robust to disturbances. A real autonomous mission requires real-time computation of the optimal waypoint, starting from a range of possible initial conditions. A specialized optimizer for this problem is needed to ensure that the waypoint can be found quickly. Developing such an optimizer is a future research task.

Finally, the waypoint-optimized ZEM/ZEV algorithm is not restricted to Mars landing. In its most general form, it can be potentially apply to many other orbital maneuvering problems. A further detailed study of the application of the generalized ZEM/ZEV algorithm, both with and without a waypoint, can be found in Ref. [25].

CHAPTER 5. Conclusion

5.1 General Summary

This dissertation has discussed a variety of guidance laws, as applied to asteroid missions. The guidance laws discussed range from classical to modern. The main applications considered are asteroid intercept and rendezvous. The guidance laws are also shown to be well-suited to other applications, such as planetary landing and orbital transfer.

The guidance laws discussed lead to the zero-effort-miss/zero-effort-velocity (ZEM/ZEV) guidance law, which is seen to be a more general form of the other laws. The ZEM/ZEV guidance law is discussed, and several extensions and modifications are shown to increase the number of applications it can be used for. These modifications include adding one or more waypoints to satisfy constraints, and including information about the dynamics of the system to overcome nonlinearity effects.

The ZEM/ZEV guidance law is shown through examples to be applicable to a wide variety of applications. In many cases, some form of ZEM/ZEV guidance can be used to achieve near-optimal performance with a closed-loop feedback guidance law.

5.2 Additional topics and future work

In addition to the research described in this dissertation, several other topics of interest have been studied at the Asteroid Deflection Research Center. These topics will be briefly discussed here, as well as descriptions of future work to be undertaken at the ADRC.

5.2.1 High-fidelity simulation with CLEON software

The effectiveness of guidance laws even in the presence of realistic measurement noise and uncertainties must be verified to assure that they can be applied in practical situations. One such simulator, GMV's CLEON, has been used at the ADRC to test guidance laws with realistic disturbances. CLEON is a Matlab-based simulator that models the full spacecraft, including sensors such as accelerometers and navigation cameras, as well as thruster dynamics. Motion of the center of brightness, described in Section 5.2.3, is also included.

CLEON can perform Monte Carlo analysis of different guidance laws, with options for different thrusters, sensors, and filters. Results of Monte Carlo analysis of the PN-based guidance laws and the predictive guidance laws can be found in Ref. [9] A full description of the CLEON package is also included.

As the ADRC continues to build a knowledge base, an in-house high-fidelity simulator should be developed, including sensors, attitude dynamics, and thrusters. Monte Carlo analysis of guidance law performance with uncertainties will continue to verify the performance of the guidance laws used for asteroid missions.

5.2.2 Missions in the irregular gravity field near an asteroid

Most asteroids of particular interest to the ADRC are irregular, and thus have a complex gravitational field. Proximity operations, including orbit and soft landing, are affected by the gravitational field. The gravitational fields of asteroids need to be studied for both modeling and guidance law design. Accurate models of gravitational fields are required for simulations to verify the performance of spacecraft systems near asteroids. Additionally, many guidance laws benefit from inclusion of a model of the asteroid's gravity field.

Fuel-efficient feedback control of spacecraft near irregular-shaped asteroids was investigated in Ref. [44]. Other aspects of proximity operations, including transfer between orbits and soft landing, was investigated in Ref. [45]. More sophisticated research into gravitational fields, including guidance law design for missions where the gravitational field is not well-known a priori, is a future research area.

5.2.3 Feedback guidance with a realistic optical navigation model

Hypervelocity impact, as demonstrated by missions such as NASA's Deep Impact mission, require autonomous GNC systems for terminal guidance. The combination of lighting conditions and the irregular shape of the asteroid presents a challenge for optical navigation systems. The ADRC is currently developing an optical navigation simulator capable of simulating camera images based on a polyhedron model of an asteroid. From these images the center of brightness is computed. The center of brightness moves as the asteroid rotates, and as the spacecraft approaches. Continued development of this optical navigation simulator will allow the ADRC to simulate missions to realistic target asteroids.

BIBLIOGRAPHY

[1] Zarchan, P., *Tactical and Strategic Missile Guidance*, 5th Ed., Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 2007.

[2] Yanushevsky, R., *Modern Missile Guidance*, CRC Press, Boca Raton, FL, 2007

[3] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, Reston, VA, 1987.

[4] Ebrahimi, B., Bahrami, M., and Roshanian, J., "Optimal Sliding-mode Guidance with Terminal Velocity Constraint for Fixed-interval Propulsive Maneuvers", *Acta Astronautica*, Vol. 62, No. 10-11, 2008, pp. 556-562. doi: 10.1016/j.actaastro.2008.02.002

[5] Wie, B., *Space Vehicle Dynamics and Control*, 2nd Ed., AIAA, Reston, VA, 2008.

[6] Vallado, D., *Fundamentals of Astrodynamics and Applications*, 3rd Ed., Microcosm Press, Hawthorne, CA, 2007.

[7] Gil-Fernández, J., Cadenas-Gorgojo, R., Prieto-Llanos, T., and Graziano, M., "Autonomous GNC Algorithms for Rendezvous Missions to Near-Earth-Objects", AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, HI, 2008.

[8] Gil-Fernández, J., Panzeca, R., and Corral, C., "Impacting Small Near Earth Objects", *Advances in Space Research*, Vol. 42, No. 8, 2008, pp. 1352-1363. doi: 10.1016/j.asr.2008.02.023

[9] M. Hawkins, A. Pitz, B. Wie, and J. Gil-Fernández, "Terminal-Phase Guidance and Control Analysis of Asteroid Interceptors", AIAA Guidance, Control, and Navigation Conference, Toronto, Canada, August 2-5, 2010.

[10] Kim, M. and Grider, K. V., “Terminal Guidance for Impact Attitude Angle Constraint Flight Trajectories”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-9, No. 6, 1973, pp. 269-278. doi: 10.1109/TAES.1973.309659

[11] Kim, B. S., Lee, J. G., and Han, H. S., “Biased PNG Law for Impact with Angular Constraint”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, 1998, pp. 277-288.

[12] Ryoo, C. K., Cho, H. J., and Tahk, M. J., “Optimal Guidance Laws with Terminal Impact Angle Constraint”, *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 724-732. doi: 10.2514/1.8392

[13] Lu, P., Doman, D. B., and Schierman, J. D., “Adaptive Terminal Guidance for Hypervelocity Impact in Specified Direction”, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 724-732. doi: 10.2514/1.14367

[14] Shaferman, V., and Shima, T., “Linear Quadratic Guidance Laws for Imposing a Terminal Intercept Angle”, *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1400-1412. doi: 10.2514/1.32836

[15] Ratnoo, A., and Ghose, D., “Impact Angle Constrained Guidance Against Nonstationary Nonmaneuvering Targets”, *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 269-275. doi: 10.2514/1.45026

[16] Yoon, M. G., “Relative Circular Navigation Guidance for Three-Dimensional Impact Angle Control Problem”, *Journal of Aerospace Engineering*, Vol. 33, No. 4, 2010, pp. 300-308. doi: 10.1061/(ASCE)AS.1943-5525.0000043

[17] Shima, T., “Intercept-Angle Guidance”, *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 484-492. doi: 10.2514/1.51026

[18] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, and Control*, Wiley, New York, 1975.

[19] Hawkins, M. and Wie, B., “Impact-Angle Control of Asteroid Interceptors/Penetrators”, AAS/AIAA Spaceflight Mechanics Meeting, New Orleans, LA, 2011.

[20] D’Souza, C. N., “An Optimal Guidance Law for Planetary Landing,” AIAA Paper 1997-3709, 1997.

[21] Guo, Y., Hawkins, M., and Wie, B., “Optimal Feedback Guidance Algorithms for Planetary Landing and Asteroid Intercept,” American Astronomical Society Paper 2011-588, 2011. AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, July 31 - August 4, 2011.

[22] Hawkins, M., Guo, Y., and Wie, B., “Guidance Algorithms for Asteroid Intercept Missions with Precision Targeting Requirements,” American Astronomical Society Paper 2011-531, 2011. AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, July 31 - August 4, 2011.

[23] Furfaro, R., Selnick, S., Cupples, M. L., and Cribb, M. W., “Nonlinear Sliding Guidance Algorithms for Precision Lunar Landing,” American Astronomical Society Paper 2011-167, 2011. 21st AAS/AIAA Space Flight Mechanics Meeting, New Orleans, LA, February 13-17, 2011.

[24] Guo, Y., Hawkins, M., and Wie, B., “Waypoint-Optimized Zero-Effort-Miss / Zero-Effort-Velocity Feedback Guidance For Mars Landing”, AAS/AIAA Space Flight Mechanics Meeting, Charleston, SC, 2012. To appear in Journal of Guidance, Control and Dynamics.

[25] Guo, Y., Hawkins, M., and Wie, B., “Applications of Generalized Zero-Effort-Miss/Zero-Effort-Velocity Feedback Guidance Algorithm”, AAS/AIAA Space Flight Mechanics Meeting, Charleston, SC, 2012. To appear in Journal of Guidance, Control and Dynamics.

[26] Açıkmeşe, B., and Ploen, S. R., “Convex Programming Approach to Powered Descent Guidance for Mars Landing”, *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353-1366. doi: 10.2514/1.27553

[27] Joseph, Z. B.-A., *Optimal Control Theory with Aerospace Applications*, AIAA Education Series, AIAA, Reston, VA, 2010. pp. 234-238.

[28] Sharma, R., Vadali, S. R., and Hurtado, J. E., "Optimal Nonlinear Feedback Control Design Using a Waypoint Method," *Journal of Guidance, Control, and Dynamics* , Vol. 34, No. 3, 2011, pp. 698-705. doi:10.2514/1.52470

[29] Elnagar, G., Kazemi, M., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal-Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793-1796. doi:10.1109/9.467672

[30] Huntington, G. T., and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2011, pp. 689-698. doi:10.2514/1.31083

[31] Rao, A. V., Benson, D. A., Darby, C., Patterson, M.A., Francolin, C., Sanders, I., and Huntington, G.T., "Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Transactions on Mathematical Software*, Vol. 37, No. 2, 2010, Article 22. pp. 29-36. doi:10.1145/1731022.1731032

[32] Karpenko, M., Bedrossian, N., Bhatt, S., Fleming, A., and Ross, I. M., "First Flight Results on Time-Optimal Spacecraft Slews," American Astronomical Society Paper 2011-110, 2011. 21st AAS/AIAA Space Flight Mechanics Meeting, New Orleans, LA, February 13-17, 2011

[33] Holmstrom, K., Goran, A. O., and Edvall, M. M., "Users Guide for TOMLAB 7," http://tomopt.com/docs/TOMLAB.pdf [retrieved May 19, 2011].

[34] Rao, A. V., Benson, D. A., Darby, C., Mahon, B., Francolin, C., and Patterson, M. A., "Users Manual for GPOPS Version 4.x: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Pseudospectral Methods," http://www.gpops.org/gpopsManual.pdf [retrieved May 19, 2011].

[35] National Research Council, *Defending Planet Earth: Near-Earth Object Surveys and Hazard Mitigation Strategies, Final Report, Committee to Review Near-Earth Object Surveys and Hazard Mitigation Strategies*, The National Academies Press, Washington, D.C., Jan. 2010. pp. 97-100.

[36] Braun, R. D., and Manning, R. M., "Mars Exploration Entry, Descent and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 310323. doi:10.2514/1.25116

[37] Wolf, A. A., Graves, C., Powell, R. W., and Johnson, W., "Systems for Pinpoint Landing at Mars," *14th AIAA/AAS Space Flight Mechanics Meeting*, AAS Paper 04-272, 2004.

[38] Desai, P. N., Prince, J. L., Queen, E. M., Cruz, J. R., and Grover, M. R., "Entry, Descent, and Landing Performance of the Mars Phoenix Lander," AIAA Paper 2008-7346, 2008.

[39] Blackmore, L., Açıkmeşe, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161-1171. doi:10.2514/1.47202

[40] Steinfeldt, B. A., Grant, M. J., Matz, D. A., Braun, R. D., and Barton, G. H., "Guidance, Navigation, and Control System Performance Trades for Mars Pinpoint Landing," *Journal of Spacecraft and Rockets*, Vol. 47, No. 1, 2010, pp. 188-198. doi:10.2514/1.45779

[41] Seifert, H. ed., *Space Technology*, Wiley, New York, 1959, Chaps. 910.

[42] Prussing, J. E., "Equation for Optimal Power-Limited Spacecraft Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 391-393. doi:10.2514/3.21017

[43] Conway, B. ed., *Spacecraft Trajectory Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2010, Chaps. 23.

[44] Winkler, T., Hawkins, M., Lyzhoft, J., and Wie, B., "Fuel-Efficient Feedback Control of Orbital Motion Around an Irregular-Shaped Asteroid", AIAA Guidance, Control, and Navigation Conference, Minneapolis, Minnesota, August, 2012.

[45] Hawkins, M., Guo, Y., and Wie, B., “ZEM/ZEV Feedback Guidance Application to Fuel-Efficient Orbital Maneuvers Around an Irregular-Shaped Asteroid”, AIAA Guidance, Control, and Navigation Conference, Minneapolis, Minnesota, August, 2012.