UNIVERSITY OF CALIFORNIA, IRVINE

Entry Guidance and Navigation for High Elevation Mars Landing

DISSERTATION

submitted in partial satisfaction of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Guangfei Duan

Dissertation Committee:

Professor Kenneth D. Mease, Chair

Professor Athanasios Sideris

Professor Haithem Taha

June 2019

Table of Contents

1 Introduction **14**

2 Entry Vehicle Dynamics Modeling **19**

2.1 Three Degree-of-freedom Modeling 20

2.1.1 Equations of Motion 20

2.1.2 Equations of Motion with Attitude Initialization Errors 22

2.2 Six Degree-of-freedom Modeling 23

2.3 System Models and Auxiliary Variables 24

2.3.1 Aerodynamic Models 24

2.3.2 Trajectory Length and Specific Energy 25

2.3.3 Downrange and Crossrange 26

3 Entry Guidance Problem **28**

4 Guidance Strategy **30**

4.1 Reference Trajectory Generation 30

4.1.1 Pre-Mission Designed Reference Trajectory 30

4.1.2 Predictor-corrector (PC-mode) 32

4.2 Reference Trajectory Tracking (RT-mode) 39

4.3 Deployment Position Alignment (DPA-mode) 42

4.4 Parachute Deployment Trigger 44

4.5 Multi-mode Guidance Logic 46

5 State Estimation **47**

5.1 Controllability and Observability of the Linearized Entry Dynamics . . 47
5.2 Nonlinear Controllability and Observability 49
5.3 Unscented Kalman Filter Enhanced Navigation Scheme 52
5.4 Analysis of the Unscented Kalman Filter 55
5.4.1 Bias Analysis 55
5.4.2 Cramer-Rao Bound 56
5.4.3 Comparison with Conventional Inertial Navigation (CIN) 58
5.4.4 Comparison with the Particle Filter Estimator 61

6 Guidance and Navigation Testing Conditions 63
6.1 Entry Conditions and Targets 63
6.2 Simulation Flow 67
6.3 Simulation Environment Setup 68
6.4 Simulation of Six Degree of Freedom Entry Dynamics 71

7 Guidance and Navigation Performance 73
7.1 Guidance Performance with Perfect Onboard Knowledge 73
7.1.1 Reference Tracking Guidance (RT) Performance 73
7.1.2 Reference Tracking with Deployment Position Alignment Guidance (RTA) 74
7.1.3 Reference Tracking, Predictor-Corrector and Deployment Position Alignment Guidance (RUTA) 75
7.1.4 Comparison of the Different Guidance Strategies 76
7.2 Delivery Performance with Navigation Errors 78
7.2.1 Guidance with Conventional Inertial Navigation (RUTA-CIN) . 80
7.2.2 Guidance with UKF-enhanced Navigation (RUTA-UKF) 83
7.2.3 Comparison of the CIN and UKF-enhanced Navigation 83

8 Conclusions 88

List of Figures

Figure 2.1 Nominal density profile. 25

Figure 2.2 Downrange (DR) and crossrange (CR). 26

Figure 3.1 Parachute deployment box. 29

Figure 4.1 Typical optimal bank angle profile generated by GPOPS− II . . 32

Figure 4.2 Parameterized bank angle profile. 33

Figure 4.3 Optimization processes of golden section and Nelder-Mead methods. 35

Figure 4.4 Boundaries of one thousand profiles of density disturbances. . . 36

Figure 4.5 Comparison of the PC-mode executed at different times. 37

Figure 4.6 Difference between planned/replanned drag profiles and the actual drag profile. 38

Figure 4.7 Parachute deployment trigger. 45

Figure 5.1 Histories of eigenvalues of the transformed controllability Grammians. 49

Figure 5.2 Time histories of the biases of state variables. 55

Figure 5.3 Time history of the bias of trajectory length. 56

Figure 5.4 History of the minimum eigenvalues of $(P - \bar{P})$ 58

Figure 5.5 Comparison of mean squared errors of CIN and UKF-enhanced navigation. 59

Figure 5.6 Histories of mean correlation coefficients between state variables. 60

Figure 6.1 g-loads performance for different entry conditions 64

Figure 6.2 Reachable sets for four entry flight path angles. 65

Figure 6.3 Trajectories with $\gamma_0 = -12^o$ and $\gamma_0 = -14.15^o$ 66
Figure 6.4 Aerodynamic coefficients profiles. 69

Figure 7.1 Bank angle profile of the RT. 74
Figure 7.2 RT performance under disturbances. 74
Figure 7.3 RTA performance under disturbances. 75
Figure 7.4 RUTA performance with optimizing three parameters under disturbances. 77
Figure 7.5 RUTA performance with optimizing one parameter under disturbances. 78
Figure 7.6 Comparison of horizontal errors of RT, RTA and RUTA under disturbances. 79
Figure 7.7 Comparison of final altitudes distribution of RT, RTA and RUTA under disturbances. 79
Figure 7.8 RUTA-CIN performance for $\gamma_0 = -15^o$ 80
Figure 7.9 RUTA-CIN performance for $\gamma_0 = -14^o$ 81
Figure 7.10 RUTA-CIN performance for $\gamma_0 = -13^o$ 82
Figure 7.11 RUTA-CIN performance for $\gamma_0 = -12^o$ 82
Figure 7.12 RUTA-UKF performance for $\gamma_0 = -13^o$ 83
Figure 7.13 RUTA-UKF performance for $\gamma_0 = -12^o$ 84
Figure 7.14 Performance comparison of RUTA-CIN and RUTA-UKF for $\gamma_0 = -12^o$ and $\gamma_0 = -13^o$ 85

List of Tables

Table 4.1 Multi-mode guidance 31

Table 4.2 Comparison between Nelder-Mead and golden section methods. . 35

Table 5.1 Particle filter algorithm 61

Table 5.2 Sampling importance resampling (SIR) algorithm 62

Table 6.1 Targets for different entry conditions 66

Table 6.2 Entry state delivery errors. 70

Table 6.3 Honeywell MIMU parameters (σ) 72

Table 7.1 Statistical results of RUTA from the simulation testing 76

Table 7.2 Performance comparison of RUTA-CIN and RUTA-UKF 86

ACKNOWLEDGEMENTS

I would like to first express my deepest gratitude to Prof. Kenneth D. Mease, whose expertise was invaluable in the formulating of the research topic and methodology in particular. His valuable guidance has helped me to proceed through each stage of the research. Throughout the writing of this dissertation, I have received a great deal of support and assistance from him. His academic rigor and clear writing have been of great help for me to become a better researcher in the future.

I want to thank the rest of the committee members, Prof. Athanasios Sideris and Prof. Haithem E. Taha for their help to my research and their feedback to my dissertation. I would also like to thank Prof. Tryphon Georgiou for his mentorship, who has pointed out a new direction of my research for further explorations.

I would like to thank my labmates, who have been providing valuable suggestions to me. I am very honored to have worked with so many intelligent labmates, including Dr. Eric Trumbauer, Dr. Alessandro Bombelli, Connor Noyes and Dr. Zixuan Liang. They have built an enthusiastic environment when I first got into this research group. Our discussions of research topics have motivated me with new approaches and our daily conversations have brought so much joy to my lab life.

I would like to acknowledge the financial support from the Homes Endowed Fellowship and the Amelia Earhart Fellowship, which have helped me to finish my Ph.D program.

Finally, I want to express my warmest gratitude to my parents, who have been of great support for me pursuing a doctoral degree in another country. Even though we are thousands of miles apart, your love is with me all the time. I would also like to

thank my friends Tina Chen, Limiao Wang and Jack Jie. When we flew together to come to the America, I never thought one day we could become best friends. All of you have added so much happiness and richness to my life.

CURRICULUM VITAE

Guangfei Duan

2013	B.S. Aerospace Engineering, Nanjing University of Aeronautics and Astronautics
2015	M.S. Mechanical and Aerospace Engineering, University of California Irvine
2015–2018	Teaching Assistant, University of California Irvine
2018	Lecturer, California State Polytechnic University Pomona
2019	Ph.D. Mechanical and Aerospace Engineering, University of California Irvine

FIELD OF STUDY

Mars Entry Guidance, Navigation system design

PUBLICATIONS

G. Duan and K. D. Mease, Mars Entry Guidance and Navigation for High Elevation Landing, Journal of Guidance, Control, and Dynamics. (In preparation)

G. Duan and K. D. Mease, Mars Entry Guidance for High Elevation Landing with Uncertainty Quantification, AIAA/AAS Astrodynamics Specialist Conference, Orlando, Sept 13-16, 2018.

J. Benito, A. Bombelli, G. Duan and K. D. Mease, Entry Trajectory Planner for Higher Elevation Mars Landing, Journal of Guidance, Control, and Dynamics. (In preparation)

G. Duan, M. Rovira Navarro and K. D. Mease, Trajectory Tracking and Online Replanning for Mars Entry, AIAA/AAS Astrodynamics Specialist Conference, Long Beach, Sept 13-16, 2016.

Z. Liang, G. Duan and Z. Ren, Mars Entry Guidance Based on An Adaptive Reference Drag Profile, Advances in Space Research, Vol. 60, No. 3, 2017, pp. 692–701.

S. F. Rafano, G. Duan, et al., Advanced Concepts for Moon Exploration - A Preliminary Study on Lunar Massive In-situ Resource Utilization to Future Space Mission Costs Reduction, 67th International Astronautical Congress, 2016.

AWARDS

Zonta International Amelia Earhart Fellowship	2017 - 2018
Holmes Endowed Fellowship	2014 - 2015

ABSTRACT OF THE DISSERTATION

Entry Guidance and Navigation for High Elevation Mars Landing

By

Guangfei Duan

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine. 2019

Professor Kenneth D. Mease, Chair

One challenge driving Mars entry, descent and landing technology development is to land at higher elevation sites without decreasing the horizontal landing accuracy relative to that achieved in the 2012 Mars Science Laboratory (MSL) Mission. Higher elevation landing capability would allow missions to the southern highlands. A combined multi-mode guidance and navigation strategy for high elevation landing is developed and tested, building on previous work on reference trajectory planning and tracking and on final position alignment. A new enhanced inertial navigation strategy is investigated to reduce the state knowledge error especially for shallower, longer duration entries.

The new contributions to the high elevation entry guidance scheme are computationally efficient reference trajectory updating requiring the optimization of a single parameter at a time and improved downrange performance in the final position guidance. To reduce the error in the estimated vehicle state on which the guidance law operates, unscented Kalman filter (UKF)-enhanced inertial navigation is proposed and

tested. The UKF-enhanced inertial navigation scheme accounts for the nonlinearity of entry dynamics and factors in the modeling errors.

The performance of the multi-mode guidance algorithm, combined with the inertial navigation system, is assessed for an MSL-type entry vehicle. Using the simulation testing, the performance with realistic levels of vehicle modeling, atmospheric density, inertial sensor, and entry delivery errors is characterized. A range of nominal entry flight path angles is considered, encompassing the MSL value as well as shallower entries that might be considered for future missions to reduce the peak acceleration and heat rate to which the vehicle is subjected. The results show that, using the multi-mode guidance and conventional inertial navigation, landing elevations as high as 1.5 km relative to the Mars Orbiter Laser Altimeter (MOLA) reference are achievable for the steep entry with -14^o initial flight path angle. For shallow entry with -12^o initial flight path angle, the multi-mode guidance with UKF-enhanced navigation can achieve the landing elevations above 0 km MOLA for 99% of the cases. For all the tested entry flight path angles, the requirement of the horizontal accuracy within 10 km has been achieved by the guidance and navigation system. For 1-2 km elevation landing, as required to access more of the southern highlands, the IMU-based navigation error is too large and modifications, possibly further improvement in the guidance and navigation algorithms or the addition of navigation sensors, would be required.

Chapter 1

Introduction

Introduction In the 2012 Mars Science Laboratory (MSL) Mission, the Curiosity rover landed with unprecedented accuracy for Mars landings by using entry guidance. For perspective, the 99.7% confidence landing ellipses for previous unguided landings were 299 km x 45 km for the 1997 Mars Pathfinder [1] and 70 km x 5 km for the 2004 Mars Exploration Rovers [1], where the major axis corresponds to downrange error and the minor axis to crossrange error. The improvement in 2004 was due to greater accuracy in the approach navigation and in the atmospheric entry state control. The 99.7% confidence landing ellipse for MSL was 19.1 km x 6.9 km [2], primarily due to the active control of the vehicle bank angle as commanded by the onboard entry guidance algorithm operating on inertial sensor data. The Curosity rover landed at an elevation of -4.4 km with a target miss distance of 2.4 km [3].

For future Mars missions, a goal is to land entry vehicles at higher elevations in order to reach scientifically interesting sites in the southern highlands [4, 5]. For higher elevation landing, the parachute must be deployed at higher altitude to provide sufficient timeline for the subsequent descent and landing phases. Previously developed guidance approaches have not addressed the objective of high elevation landing, with the exception of the work [6, 7] on which the present work builds. MSL used a neighbor-

ing guidance law which was initially developed for Apollo reentry at Earth. Based on a pre-designed reference trajectory and a prescribed heading angle corridor, the entry guidance consists of a feedback control law that commands the bank angle to compensate for final range, drag acceleration, and altitude rate errors relative to the reference values and bank reversal logic to stay within the heading corridor [9, 10, 11]. The MSL parametrized bank angle profile was not designed to achieve high elevation landings, and it has been shown that the MSL bank profile does not yield the maximum altitude achievable, even when the parameters are optimized [6, 7, 8]. Predictor-corrector entry guidance has been investigated during recent years [11, 12, 13, 14]. Instead of tracking a pre-designed reference trajectory, a predictor-corrector generates bank angle commands by recomputing the entire trajectory and bank angle profile repeatedly during entry, based on the onboard measurements and the nominal entry dynamics, using a parametrization of the bank angle profile, typically with only one parameter to adjust [11]. The guidance accuracy can be improved relative to following a fixed reference trajectory at the expense of increased onboard computation. However, the objective of high elevation landing has not been addressed in predictor-corrector guidance research.

In order to achieve high elevation landing, a multi-mode guidance strategy has been pursued. Given a pre-designed near-optimal reference trajectory, the guidance operates in the modes of reference trajectory tracking, bank reversal execution, and final position alignment. The reference trajectory pre-design [7] generates an entry trajectory and bank angle profile that achieves nearly the highest parachute deployment altitude, for which the vehicle is capable, above the specified target landing site. The optimal bank angle profile is bang-bang with three switches in the bank angle sign. Using a 3-parameter profile allows the optimal control problem to be recast, approximately, as a parameter optimization problem, making it more feasible for onboard usage. A model predictive feedback law, developed in [15, 16], is used to track the near-optimal reference trajectory. At a certain Mach number, there is a switch to final position alignment [17], an enhanced version of the MSL heading alignment algorithm that takes the

high-altitude objective into consideration as well as the heading error.

The goal of the research documented in this dissertation is to develop a combined guidance and navigation strategy that enables high elevation landing, building on previous work described in the previous paragraph. The guidance and navigation strategy is designed for an MSL-type entry vehicle with bank angle control and inertial sensor based navigation for the more specific goal of determining the highest elevation landing for which such a vehicle is capable. The guidance and navigation algorithms are implemented in software and assessed via computer simulation against the requirement to deliver the entry vehicle to within 10 km of the specified high elevation landing site with at least the minimum surface-relative altitude of 6 km, and within the other constraints for successful parachute deployment, descent and landing. The delivery error includes both the guidance error and the navigation error. The delivery requirements, with the exception of site elevation, are approximately the MSL entry guidance requirements. Preliminary results [19] indicate that further development of the previously developed multi-mode guidance is required to meet the delivery requirements. With the expectation that future entry missions will require, relative to the MSL entry, shallower entry, in order to reduce the maximum g-load and peak heat rate [2], an enhanced inertial navigation strategy is investigated to reduce the vehicle state knowledge error especially for shallower, longer duration entries.

The specific objectives of the research are (i) to improve the guidance performance without increasing significantly the onboard computation by showing that a few 1-parameter updates to the reference trajectory are sufficient, when combined with tracking and final position alignment, to achieve the required guidance accuracy, (ii) to improve the final position alignment by modifying it to reduce downrange errors, (iii) to reduce navigation errors, especially for shallower, longer duration entries, through the use of an unscented Kalman filter (UKF) to process the inertial sensor data, and (iv) to determine via simulation testing whether or not the performance of the combined

multi-mode guidance and enhanced inertial navigation allows higher elevation landing while meeting all the other delivery requirements.

Navigation schemes for Mars entry vehicles have been investigated since the first two Viking missions [21] . Up to now, all the Mars landing missions, including MSL, have used the IMU acceleration measurements during entry without other sensors providing information on the estimates of the trajectory state [2]. Most of the recent research has focused on adding additional potential navigation information to improve estimation accuracy [22, 23, 24]. With the objective of more accurate navigation with only IMU sensors, the feasibility processing IMU data with an extended Kalman filter was investigated in [21]; it was found that the navigation errors were not reduced sufficiently, because of the density modeling errors and the linearization of the dynamics [21]. Heyne proposed the use of an unscented Kalman filter [25] . A UKF was used in the reconstruction of MSL trajectories [26, 27]. In [22, 28], UKF-based navigation was studied for Mars entry, but extra measurements such as flight path angle rate or range measurement were added to improve the performance. In that study, the UKF propagated the state mean and covariance through the unscented transformation; the uncertain parameters, which significantly affect the delivery error are not considered. In this dissertation, we propose and evaluate a UKF-enhanced inertial navigation strategy. The filter state vector is augmented to include uncertain model and sensor error parameters, which has been shown beneficial for a system with nonlinear process and measurement models, as compared with the standard Kalman filter [22].

This dissertation is organized as follows. The entry dynamics and modeling are given in Chapter 2. In Chapter 3, the entry guidance problem is stated. The multi-mode entry guidance algorithm is presented in Chapter 4, and UKF-enhanced inertial navigation is covered in Chapter 5. In Chapter 6, the simulation testing conditions for the guidance and navigation performance assessment are provided. The results of the performance assessment are presented in Chapter 7, and the conclusions are given in

Chapter 8.

Chapter 2

Entry Vehicle Dynamics Modeling

Mars entry problem can be modeled using four coordinate frames. The translational dynamics for a Mars entry vehicle are defined relative to a Mars-fixed coordinate frame $O - X_m Y_m Z_m$. This frame is rotating along the Z_m axis with angular velocity ω_p, the Martian self-rotation angular velocity. The navigation frame (local horizontal frame) $O - X_n Y_n Z_n$ is obtained by rotating around Z_m axis with angle θ, vehicle longitude and then rotating around Y_n with angle $-\phi$, vehicle latitude. It is centered at the center of Mars, with X_n axis along the radial position vector, Y_n pointing to the East, and Z_n pointing to the North, i.e., a UEN frame. Y_n and Z_n span the local horizontal plane. The transformation from $O - X_m Y_m Z_m$ to $O - X_n Y_n Z_n$ is

$$R_{m2n} = \begin{bmatrix} cos(-\phi) & 0 & -sin(-\phi) \\ 0 & 1 & 0 \\ sin(-\phi) & 0 & cos(-\phi) \end{bmatrix} \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Velocity frame $M - X_v Y_v Z_v$ is centered at the center of the vehicle. Y_v is along with the planet relative velocity vector, and X_v is aligned with the lift vector component in the vertical plane. Z_v completes the coordinate system. The transform from $O - X_n Y_n Z_n$

to $O - X_v Y_v Z_v$ is

$$R_{n2v} = \begin{bmatrix} cos(-\gamma) & sin(-\gamma) & 0 \\ -sin(-\gamma) & cos(-\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\psi) & sin(\psi) \\ 0 & -sin(\psi) & cos(\psi) \end{bmatrix}$$

where γ and ψ define the vehicle path angle and heading angle. The body frame $M - X_b Y_b Z_b$ is defined along the axes of the onboard inertial measurement unit (IMU). The directional cosine matrix, which transforms the body frame to the navigation frame is

$$C_{b2n} = \begin{bmatrix} q_a^2 + q_b^2 - q_c^2 - q_d^2 & 2(q_b q_c - q_a q_d) & 2(q_b q_d + q_a q_c) \\ 2(q_b q_c + q_a q_d) & q_a^2 - q_b^2 + q_c^2 - q_d^2 & 2(q_d q_c - q_a q_b) \\ 2(q_b q_d - q_a q_c) & 2(q_d q_c + q_a q_b) & q_a^2 - q_b^2 - q_c^2 + q_d^2 \end{bmatrix}$$

where $Q = [q_a,\ \mathbf{q}]^T = [q_a,\ q_b,\ q_c,\ q_d]^T$ is the quarternion vector which defines the attitude of the vehicle.

2.1 Three Degree-of-freedom Modeling

The three degree-of-freedom translational dynamics for a Mars entry vehicle are defined relative to a Mars-fixed coordinate frame $O - X_m Y_m Z_m$. In this three degree-of-freedom modeling, we have assumed that the velocity frame and the body frame are aligned and do not consider the vehicle attitude.

2.1.1 Equations of Motion

The entry vehicle radial position projected on the navigation frame is $r_m^n = [r,\ 0,\ 0]^T$, where r is the distance between the center of the vehicle and the Mars center. Thus the relative velocity to the Mars-fixed frame projected on the navigation frame would be

$$V_m^n = \frac{dr_m^n}{dt} = \dot{r} \cdot i_n + r \cdot \dot{i}_n = V \cdot i_n + r \cdot (\Omega_{mn} \times i_n) \tag{2.1}$$

where $\Omega_{mn} = \dot{\theta} \cdot (R_{m2n} \cdot k_m) - \dot{\phi} \cdot j_n$ is the angular velocity of the navigation frame. The relative velocity is defined in the velocity frame $M - X_v Y_v Z_v$ as $V_m^v = [0,\ V,\ 0]^T$. It is transformed into the navigation frame as

$$V_m^n = R_{n2v}^T \cdot V_m^v = \begin{bmatrix} V_U \\ V_E \\ V_N \end{bmatrix} = \begin{bmatrix} V sin\gamma \\ V cos\gamma cos\psi \\ V cos\gamma sin\psi \end{bmatrix} \tag{2.2}$$

Combining Eqs.(2.1) and (2.2), we can get

$$\begin{aligned} \dot{r} &= V \sin\gamma \\ \dot{\theta} &= \frac{V}{r} \frac{\cos\gamma \cos\psi}{\cos\phi} \\ \dot{\phi} &= \frac{V}{r} \cos\gamma \sin\psi \end{aligned} \tag{2.3}$$

Drag and lift expressed in the velocity-fixed frame are $D^v = [0,\ -D,\ 0]^T$, $L^v = [Lcos\sigma,\ 0,\ -Lsin\sigma]^T$. Gravitational acceleration in the navigation frame is defined as $g_m^n = [-\mu/r^2,\ 0,\ 0]^T$, where $\mu = 42,409\ km^3/s^2$ is the gravitational constant. Thus the relative acceleration of the entry vehicle with respect to the Mars-fixed frame is

$$a_m^n = R_{n2v}^T \cdot (D^v + L^v) + g_m^n \tag{2.4}$$

The inertial acceleration of the vehicle is

$$a = \frac{dV_m^n}{dt} = \frac{d^2 r_m^n}{dt^2} \tag{2.5}$$

And

$$a = a_m^n - 2\Omega_{im} \times V_m^n - \Omega_{im} \times (\Omega_{im} \times r_m^n) \tag{2.6}$$

where $\Omega_{im} = [\omega_p sin\phi,\ 0,\ \omega_p cos\phi]^T$ is the Mars angular velocity ω_p projected on the navigation frame. The centripetal acceleration, which is proportional to ω_p^2, is negligible and therefore is usually not included in the entry dynamics. Substituting Eq.(2.4) and

Eq.(2.5) into Eq.(2.6) yields

$$\begin{aligned}
\dot{V} &= -D - g\sin\gamma \\
\dot{\gamma} &= \frac{1}{V}\left[L\cos\sigma - \left(g - \frac{V^2}{r}\right)\cos\gamma\right] + 2\omega_p\cos\psi\cos\phi \\
\dot{\psi} &= -\frac{1}{V\cos\gamma}\left(L\sin\sigma + \frac{V^2}{r}\cos^2\gamma\cos\psi\tan\phi\right) + 2\omega_p\left(\tan\gamma\sin\psi\cos\phi - \sin\phi\right)
\end{aligned} \tag{2.7}$$

Eq.(2.3) and Eq.(2.7) define the dynamics for a Mars entry vehicle, where $x = [r,\ \theta,\ \phi,\ V,\ \gamma,\ \psi]^T$ is the state vector, $(r,\ \theta,\ \phi)$ are spherical coordinates for the vehicle center of mass. (V, γ, ψ) define the relative velocity vector by flight path angle and heading angle. Heading angle is defined as a clockwise rotation angle with $\psi = 0$ for East. Bank angle σ, the clockwise rotation angle between the lift vector and the vertical plane containing relative velocity vector is considered as the control variable, with $\sigma = 0$ correspondings to the full lift-up condition. The lift and drag accelerations are defined by

$$\begin{aligned}
L &= \frac{1}{2}\rho V^2\frac{A}{m}C_L \\
D &= \frac{1}{2}\rho V^2\frac{A}{m}C_D
\end{aligned} \tag{2.8}$$

where ρ represents the Martian atmosphere density. A and m are the surface area and the mass of the entry vehicle. C_L and C_D are the vehicle lift and drag coefficients. Though the drag and lift accelerations are defined in the velocity-fixed frame, where IMU measurements are defined in the body-fixed frame of the spacecraft, with the simplified three degree-of-freedom modeling, we can assume these two frames are aligned and directly use the outputs of IMU as the measurements of drag and lift accelerations.

2.1.2 Equations of Motion with Attitude Initialization Errors

The entry dynamics are propagated using measurements from the onboard inertial measurement units (IMU) for lift and drag accelerations. When the IMU is initialized prior to entry, some attitude initialization error may cause IMU misalignment. Assume the attitude initialization error is ϵ^o around any random axis $\vec{u} = [u_x\ u_y\ u_z]^T$. Thus

the relative acceleration vector measured by IMU is rotated from the navigation frame with a rotation matrix

$$\Delta R = \begin{bmatrix} cos(\epsilon) + u_x^2(1-cos(\epsilon)) & u_x u_y(1-cos(\epsilon)) - u_z sin(\epsilon) & u_x u_z(1-cos(\epsilon)) + u_y sin(\epsilon) \\ u_x u_y(1-cos(\epsilon)) + u_z sin(\epsilon) & cos(\epsilon) + u_y^2(1-cos(\epsilon)) & u_y u_z(1-cos(\epsilon)) + u_x sin(\epsilon) \\ u_x u_z(1-cos(\epsilon)) - u_y sin(\epsilon) & u_y u_z(1-cos(\epsilon)) + u_x sin(\epsilon) & cos(\epsilon) + u_z^2(1-cos(\epsilon)) \end{bmatrix}$$

Then the relative acceleration projected on the navigation frame becomes

$$a_m^n = \Delta R \cdot R_{n2v}^T \cdot (D^v + L^v) + g_m^n \tag{2.9}$$

Substituting Eq.(2.9) into Eq.(2.6) yields the equations of motion with attitude initialization errors.

2.2 Six Degree-of-freedom Modeling

The state vector, used in the modeling of the 6-DOF entry dynamics is usually expressed in the navigation frame as $X_{nav} = [\mathbf{p}, \mathbf{V}, \mathbf{q}]$. $\mathbf{p} = [h,\ \theta,\ \phi]$ and $\mathbf{V} = [V_U,\ V_E,\ V_N]$ define the translational dynamics as in the 3-DOF modeling. $\mathbf{q} = [q_a,\ q_b,\ q_c,\ q_d]^T$ is the quarternion vector which defines the vehicle attitude. The propagation of the quarternion vector is given by

$$\begin{aligned} \dot{q_a} &= -\frac{1}{2}(q_b w_x + q_c w_y + q_d w_z) \\ \dot{q_b} &= \frac{1}{2}(q_a w_x - q_d w_y + q_c w_z) \\ \dot{q_c} &= \frac{1}{2}(q_d w_x + q_a w_y - q_b w_z) \\ \dot{q_d} &= -\frac{1}{2}(q_c w_x - q_b w_y - q_a w_z) \end{aligned} \tag{2.10}$$

where $\omega_b = [w_x,\ w_y,\ w_z]^T$ is the body rate vector, which can be obtained from the IMU gyro readout. The propagation of the relative velocity is defined in the navigation frame as

$$\dot{V}_m^n = C_{b2n} f_b - (2\omega_{im} + \omega_{mn}) \times V_m^n - \omega_{im} \times (\omega_{im} \times r_m^n) + g_m^n \tag{2.11}$$

where f_b is the readout from the IMU accelerometers. During the simulations, readouts from IMU gyro and accelerometer are generated as

$$\begin{aligned} a_m &= R_{v2n}^T(D^v + L^v) \\ f_b &= (C_{n2b}a_m) \\ \omega_{ib} &= (C_{n2b}(\omega_{im} + \omega_{mn}) + \omega_{nb}) \end{aligned} \tag{2.12}$$

where $R_{v2n} = R_{n2v}^T$ and $C_{n2b} = C_{b2n}^T$, and ω_{nb} is the angular velocity of the entry vehicle expressed in the body frame

$$\omega_{nb} = \begin{bmatrix} \dot{\gamma}sin(\sigma) + \dot{\psi}cos(\gamma)cos(\sigma) \\ \dot{\sigma} + \dot{\psi}sin(\gamma) \\ -\dot{\gamma}cos(\sigma) + \dot{\psi}cos(\gamma)sin(\sigma) \end{bmatrix} \tag{2.13}$$

Finally, the propagation of entry vehicle position is defined as

$$\begin{aligned} \dot{h} &= V_U \\ \dot{\theta} &= \frac{V_E}{r\cos\phi} \\ \dot{\phi} &= \frac{V_N}{r} \end{aligned} \tag{2.14}$$

Thus, Eqs. (2.10)-(2.14) define the 6 degree-of-freedom entry dynamics for the entry vehicle.

2.3 System Models and Auxiliary Variables

2.3.1 Aerodynamic Models

Mars is assumed as a spherical planet with radius $r_p = 3397\ km$ and angular velocity $\omega_p = 10^{-5}\ rad/s$. The altitude of the center of the vehicle is computed as $h = r - r_p$. The nominal atmospheric density model is in the form of an exponential function

$$\rho(h) = \rho_0 \cdot exp\left(\beta_1 \cdot h + \beta_2 \cdot h^2 + \beta_3 \cdot h^3 + \beta_4 \cdot h^4\right) \tag{2.15}$$

where ρ_0 and β_i are constant coefficients that are curve-fitted from 500 MSL data profiles [30]. The nominal density profile is shown in Fig. 2.1. The vehicle model of the

MSL type is assumed. The aerodynamic coefficients C_L and C_D are functions of Mach number which are also curve-fitted from 500 MSL data profiles [30]. The MSL-type of entry vehicle has a low lift-to-drag ratio around 0.24 - 0.31.

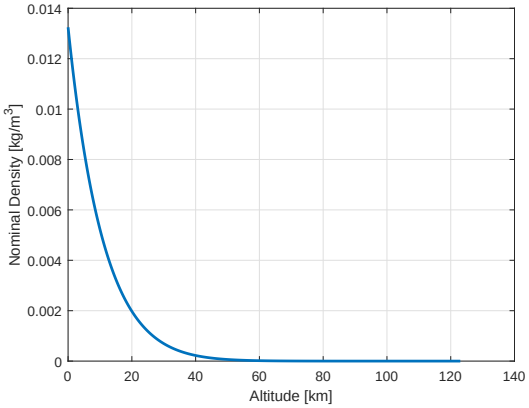


Figure 2.1: Nominal density profile.

2.3.2 Trajectory Length and Specific Energy

Trajectory length is computed as the integration of the velocity projected on the local horizontal plane

$$S = \int_{t_0}^{t_f} V \cdot cos\gamma \; dt \tag{2.16}$$

During the entry phase, the energy, defined as $E = \frac{1}{2}V^2 - \frac{\mu}{r}$ is usually used as the independent variable instead of the time. Because

$$\begin{aligned} \dot{S} &= V \cdot cos(\gamma) \\ \dot{E} &= V\dot{V} + \frac{\mu}{r^2}\dot{\gamma} \end{aligned} \tag{2.17}$$

we can write

$$\frac{dS}{dE} = \frac{\dot{S}}{\dot{E}} \tag{2.18}$$

Substituting Eq.(2.17) into Eq.(2.18), we can get that $\frac{dS}{dE} = -cos(\gamma)/D$. Therefore, the relationship between trajectory length and the specific energy can be written

$$S = -\int_{E_0}^{E_f} \frac{cos(\gamma)}{D} dE \tag{2.19}$$

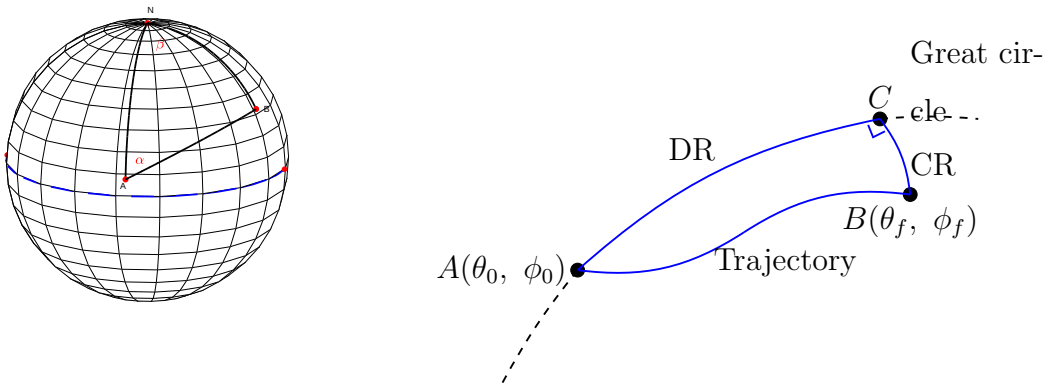
If the vehicle is flying with a constant altitude, which means $cos(\gamma) \approx 1$, we can further simplify Eq.(2.19) to

$$S = -\int_{E_0}^{E_f} \frac{1}{D} dE \tag{2.20}$$

This equation shows a direct relationship between the trajectory length and the drag acceleration.

2.3.3 Downrange and Crossrange

Vehicle ground track points are expressed in downrange (DR) and crossrange (CR). Downrange is defined as the distance along the great circle segment from the entry state to the target and crossrange is defined as the distance normal to the great circle, as shown in Fig. 2.2 [30].



(a) (b)

Figure 2.2: Downrange (DR) and crossrange (CR).

In Fig. 2.2, point $A(\theta_0, \phi_0)$ denotes the vehicle position in the entry interface, and point $B(\theta_f, \phi_f)$ is the vehicle current position or the target. Downrange and crossrange segments intersect at point C. N is the north pole. According to the spherical

trigonometry

$$\begin{aligned} cos\widehat{NB} &= cos\widehat{NA} \cdot cos\widehat{AB} + sin\widehat{NA} \cdot sin\widehat{AB} \cdot cos\alpha \\ sin\phi_f &= sin\phi_0 \cdot cos\widehat{AB} + cos\phi_0 \cdot sin\widehat{AB} \cdot cos\alpha \end{aligned} \tag{2.21}$$

According to Fig. 2.2(b), $\widehat{AC}$ is the downrange, $\widehat{BC}$ is the crossrange, and

$$\begin{aligned} cos\widehat{AB} &= cos\widehat{AC} \cdot cos\widehat{BC} \\ sin\frac{\widehat{CR}}{r_p} &= sin\zeta \cdot sin\widehat{AB} \end{aligned} \tag{2.22}$$

where ξ is the angle between the great circle crossing A and B and the great circle crossing A and C, and $\widehat{AB}$ denotes the angular distance between the two points defined by longitude and latitude, which can be computed by spherical trigonometry relations, as [30]

$$\widehat{AB} = 2r_p \cdot arcsin\sqrt{sin^2(\frac{\phi_f - \phi_0}{2}) + cos(\phi_f)cos(\phi_0)sin^2(\frac{\theta_f - \theta_0}{2})} \tag{2.23}$$

Solving Eq.(2.21) and Eq.(2.22), we can get

$$\begin{aligned} \widehat{AB} &= arccos(cos\frac{\widehat{DR}}{r_p} \cdot cos\frac{\widehat{CR}}{r_p}) \\ \zeta &= arcsin(sin(\frac{\widehat{CR}}{r_p})/sin\widehat{AB}) \\ \alpha &= \zeta - \psi_0 + \pi/2 \end{aligned} \tag{2.24}$$

Using Eq.(2.24), given the initial point and required downrange and crossrange, we can get the final position of the entry vehicle, and vice versa, i.e., given the initial point and the target point, we can calculate the downrange and crossrange.

Chapter 3

Entry Guidance Problem

We consider the entry guidance objective to be delivering the vehicle to a parachute deployment state that satisfies the deployment position and parachute requirements. The entry guidance must accommodate dispersions (off-nominal values) in the initial entry state, the atmospheric density, and the vehicle aerodynamics. We assume the flight path can be modified by adjusting the direction of the lift force via changes in the bank angle σ. The guidance logic issues bank angle commands to the reaction control system (RCS). The bank angle commands should respect magnitude rate and acceleration constraints $|\sigma| \leq \sigma_{max}$, $|\dot{\sigma}| \leq \dot{\sigma}_{max}$ and $|\ddot{\sigma}| \leq \ddot{\sigma}_{max}$.

The final state $x(t_f) = x_f$ is constrained by the parachute deployment box and the surface location of the targeted landing site. MSL used a single supersonic parachute with 16 m diameter. It can be deployed at dynamic pressures between 300 and 800 Pa and Mach numbers between 1.4 and 2.2. We assume the same parachute and converted these constraints into the form of velocity and altitude constraints in order to efficiently implement them in the parachute deployment trigger. In addition to these parachute deploy constraints, we impose a minimum deployment altitude in order to avoid the low-elevation deployments which might fail the mission. This is a timeline margin requirement, allowing sufficient time to be spent on the powered descent and landing.

A minimum altitude of 6.0 km above the ground has been found necessary and 6.0 km or more is preferred [2]. For the proposed guidance with the objective of high elevation landing, we select the 6.0 km as the minimum deployment altitude. Fig. 3.1 shows the parachute deployment box in the altitude-velocity plane.

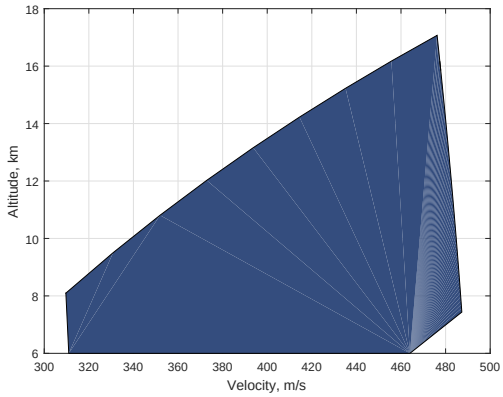


Figure 3.1: Parachute deployment box.

Future Mars landing missions expect that EDL sequence will lead the vehicle to the target site elevations above 0 km relative to the Mars MOLA [18]. Deploying the parachute at an altitude above 6.0 km, which leaves enough time and altitude for the subsequent descent and landing phases, would enable a positive landing elevation [2]. Furthermore, the MSL mission required a touchdown ellipse of 19.1 $\times$ 6.9 km for the site of scientific interest, which led to the requirement for parachute deployed within 10 km to the target. Therefore, the primary goal of the proposed guidance is to achieve a parachute deployment elevation above 6.0 km and a total delivery error less than 10 km to the target for 99% cases, by tracking the reference trajectory under actual entry conditions with all the above guidance constraints satisfied.

Chapter 4

Guidance Strategy

The proposed entry guidance for high elevation landing is multi-mode with switching logic. The modes are as shown in Table 4.1. The guidance algorithm is initialized with a pre-designed reference trajectory. It is tracked using the RT-mode until the switching logic calls for a reference trajectory update using the PC-mode. During the updating, the RT-mode tracks the current reference, and upon completion of the update, it starts tracking the updated reference. The current reference trajectory dictates bank reversal initiation times. At such a time, there is a switch to BR-mode until completion of the reversal, then there is a switch back to RT-mode. There is a switch to DPA-mode in the latter part of the entry. Each mode is introduced in one the following sections and the multi-mode guidance logic is summarized in Section 4.5.

4.1 Reference Trajectory Generation

4.1.1 Pre-Mission Designed Reference Trajectory

The optimal reference trajectory is first generated pre-mission using numerical optimization algorithms. A general-purpose MATLAB software program called General Pseudospectral Optimal Control Software (GPOPS− II) is used as the programming solver to solve the multiple-phase optimal control problems using variable-order Gaus-

Table 4.1: Multi-mode guidance

Guidance Mode	Description
Predictor-Corrector (PC-mode)	most computationally intensive, used sparingly to update reference for consistency with current state and data-based correction of density model
Reference Trajectory Tracking (RT-mode)	MPC feedback tracking of reference, follows drag profile, rather than altitude profile, for robustness to density model errors
Bank Reversal (BR-mode)	minimum-time open-loop rotation, through zero bank angle, to opposite sign of current bank angle command, respecting constraints on rate and acceleration
Deployment Position Alignment (DPA-mode)	nonlinear inversion feedback regulation of heading with modifications for parachute deployment position control

sian quadrature collocation methods. The software employs a Legendre-Gauss-Radau quadrature orthogonal collocation method where the continuous-time optimal control problem is transcribed to a large sparse nonlinear programming problem (NLP) [31].

During the optimization process, bank angle, bank angle rate, and bank angle acceleration are constrained as $|\sigma| \leq 90^o,\ |\dot{\sigma}| \leq 20^o/s,\ |\ddot{\sigma}| \leq 5^o/s^2$. We have set the final horizontal accuracy as a hard constraint, i.e., $d = 0$, while maximizing the final altitude with a cost function

$$J = -h_f \tag{4.1}$$

The optimal bank angle profile, which is in the format of bang-bang control, typically has three bank reversals, as shown in Fig. 4.1. Using GPOPS− II [31] as the solver generates the reference trajectory with good performance, but yet is computationally

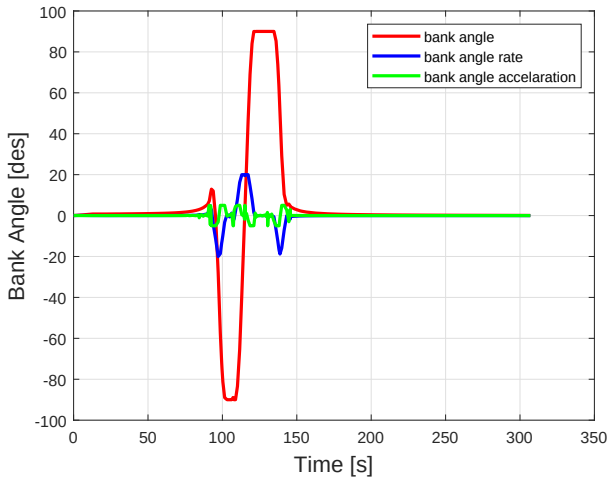


Figure 4.1: Typical optimal bank angle profile generated by GPOPS− II

expensive. It cannot be implemented efficiently for onboard trajectory updating. Therefore, a 3-parameter bank profile that captures the structure of the optimal bang-bang profile allows the reference planning to be cast as a parameter optimization problem [8].

4.1.2 Predictor-corrector (PC-mode)

The reference trajectory in the guidance algorithm is generated by a high elevation trajectory planner [8] under nominal conditions, where a low-order parameterized bank angle profile is used. The time t is the independent variable with $t \in [0, t_f]$ and t_f is a free variable. The bank angle profile is parameterized with three bank reversal times $t_1,\ t_2,\ t_3$, as shown in Fig. 4.2. The boundary of the bank angle is set as 85^o in order to reserve some margin for lateral control during the RT-mode. For the same reason, the first and last segments of the bank angle profiles are set as $\pm 20^o$, instead of the 0^o as in the optimal trajectory. The same constraints as mentioned in Section 4.1.1 are applied to bank angle rate and bank angle acceleration. During the bank reversal, bank acceleration is set as its maximum value in order to finish the reversal process within the shortest time.

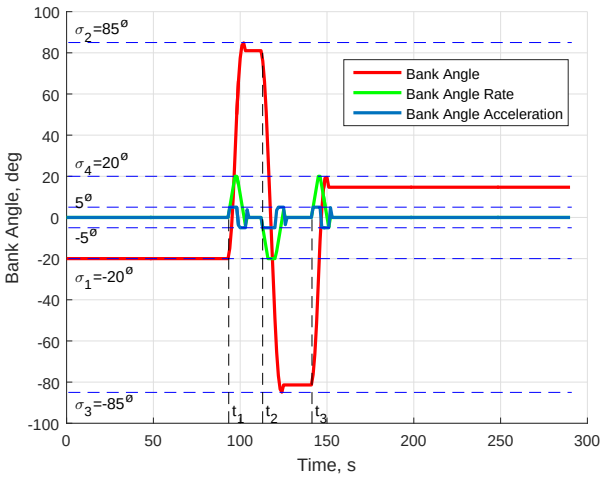


Figure 4.2: Parameterized bank angle profile.

In order to realize the objective of landing a vehicle with a low lift-to-drag ratio on high elevation sites while keeping the horizontal accuracy required by MSL and maintaining the control authority to correct for dispersion, compared with Eq.(4.1), we add two terms representing the horizontal error and altitude control authority to the cost function. Therefore, the performance index for this optimization problem is a weighted combination of the vehicle horizontal error, final altitude and final control authority at the parachute deployment level, as follow

$$J = -k_h h_f + k_d d + k_\gamma \gamma_f^2 \tag{4.2}$$

where d is the horizontal distance between the vehicle final location and the target, which can be computed from Eq. (2.23). The magnitude of final flight path angle, i.e., $|\gamma_f|$ presents the altitude control authority. In the altitude acceleration

$$\ddot{h} = -g - D\sin\gamma + \frac{V^2}{r}\cos^2\gamma + 2\omega_p V\cos\gamma cos\psi cos\phi + L\cos\gamma\cos\sigma \tag{4.3}$$

the term $Lcos\gamma$ determines the effect of the control input $\cos\sigma$. As the magnitude of lift vector decreases during the latter phase of entry, the primary mean of maintaining the control authority would be keeping the magnitude of the flight path angle as small

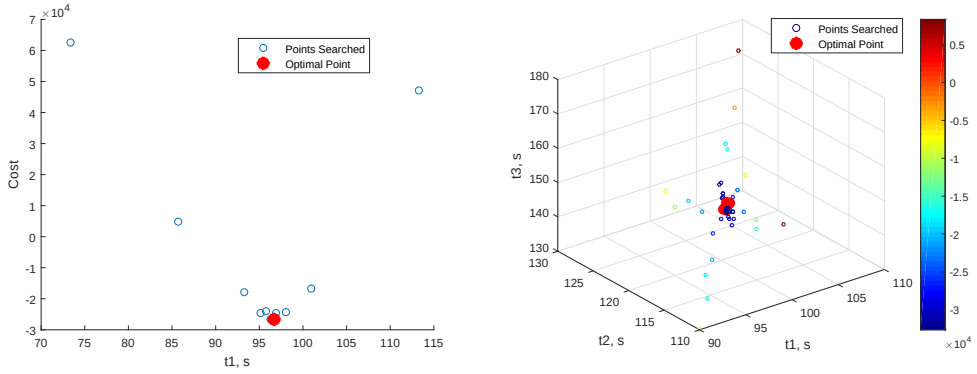
as possible. Choosing the target with maximum landing elevation within the reachable set for a given entry condition, this reference trajectory generated by this planner has a final horizontal error around 200 m and a final altitude above 11 km. Both of the horizontal and altitude performances of this reference trajectory are sufficiently good, compared with the MSL requirements. The reference trajectory is stored prior to entry as drag-vs-energy and trajectory length-vs-energy profiles that are to be tracked by entry guidance.

The PC-mode generates new reference trajectory accounting for the current state and a sensor-based adjustment of the density model. As mentioned before, an MSL-like vehicle with low lift-to-drag ratio might not have enough capacity to adjust the trajectory for a fixed reference trajectory.

However, the implementation of optimization during entry is time-consuming. If more than two parameters are replanned, the optimization method for the PC-mode is the same as the method used in planning, which is the Nelder-Mead simplex method [8]. This method takes more computational effort since it optimizes two or three parameters in a two or three dimensional space. In addition, this method requires well-chosen initial values. A poorly chosen set of initial values might not converge to a local minimum. However, if only one parameter needs to be optimized during the PC-mode, a simple line search such as the golden section method would search for the local minimum much more efficiently. Take one trajectory as an example. Fig. 4.3 and Table 4.2 show the optimization processes of the golden section and the Nelder-Mead methods. We can see the number of the function evaluations of line search is much less than the that of the Nelder-Mead method and the one dimensional search process greatly reduces the optimization computational time .

Table 4.2: Comparison between Nelder-Mead and golden section methods.

Optimization Method	Nelder-Mead	Nelder-Mead	Golden section
Parameters Optimized	$t_1,\ t_2,\ t_3$	t_2, t_3	t_3
Number of Function Evaluations	82	61	13
Time cost(s)	4.56	2.68	0.78



(a) Golden section

(b) Nelder-Mead

Figure 4.3: Optimization processes of golden section and Nelder-Mead methods.

Furthermore, in order to make adjustments to all the reversal times in each PC-mode, even with only one parameter optimized, the intervals between any two bank reversal times are kept constant. So if one reversal time is replanned, the other two reversal times are translated correspondingly in order to keep the intervals unchanged. This technique have all the three parameters adjusted in each PC-mode with only one optimized, which reduces the computational time. The simulation testing in Chapter 7 shows the comparison of the PC-mode using Nelder-Mead with updating three parameters and the PC-mode using golden section with updating one parameter. And the results show that one dimensional optimization is sufficient to achieve good performance in reducing final delivery errors.

In order to reduce the impact of atmosphere modeling errors on the future trajectory, the onboard planner takes into account perturbations when updating the drag profile. This is done by using the coefficient k_D

$$k_D = \frac{1}{t - t_a} \int_{t_a}^{t} \frac{D}{D_{\text{expected}}} dt \tag{4.4}$$

where D_{expected} is the drag under nominal conditions for the atmosphere profile. This coefficient reflects the accumulated difference between the actual atmosphere and the nominal condition from the entry state to the current moment. The drag and lift profiles used in the PC-mode are multiplied by this coefficient to be corrected. This assumes, in the PC-mode, the vehicle is flying through a thinner or thicker atmosphere than expected based on the trajectory that the vehicle has flown so far. However, it is not necessary to include all the perturbations from the entry interface. Fig. 4.4 shows the boundary of 1000 density disturbances profiles, which are generated by the Mars Climate Data (MCD). From this figure, we can see the atmospheric disturbances above 40 km can be as large as 24% of the nominal values, which are much larger than those of lower altitudes. Using k_D from high elevations would result to an inaccurate prediction of the future atmosphere conditions. Thus t_a in Eq. (4.4) is chosen as the time when the vehicle altitude is around 40 km.

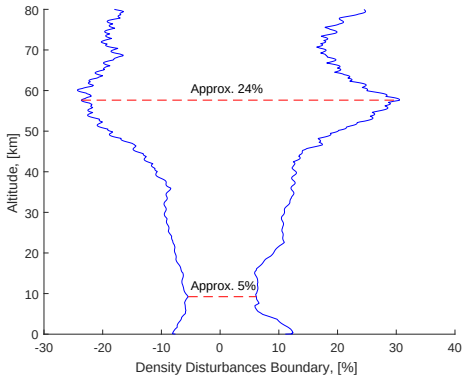
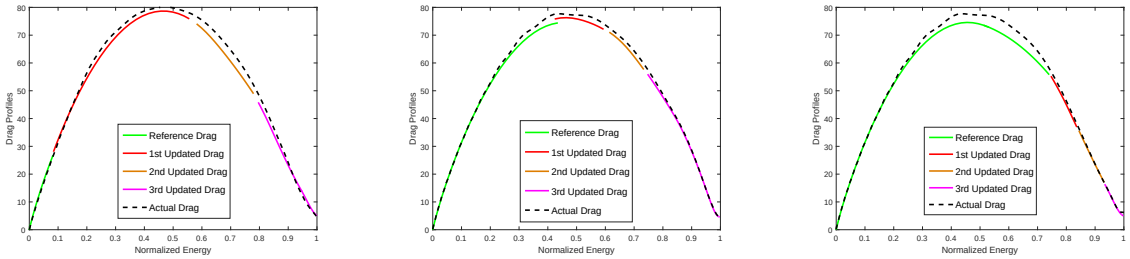


Figure 4.4: Boundaries of one thousand profiles of density disturbances.

The timing and frequency to execute the PC-mode are important to guidance performance. Possible initiation times of the PC-mode during the entry are: 1) when the

vehicle altitude is under 40 km; 2) when the vehicle just started the first bank reversal; 3) when the vehicle just started the second bank reversal; 4) when the vehicle just finished the second bank reversal; 5) when the vehicle just started the third bank reversal. The PC-mode cannot be executed too frequently in order to save the onboard computational time. After the preliminary assessment, it is determined that the PC-mode are initiated three times during the entry. Fig. 4.5 shows a comparison between scenarios where the PC-mode are executed at different times.



(a) Update at 1), 2), 3). (b) Update at 2), 3), 4). (c) Update at 3), 4), 5).

Figure 4.5: Comparison of the PC-mode executed at different times.

Fig. 4.5 shows the profiles of reference drag, updated drag in the PC-mode and actual drag that the vehicle experienced for the same trajectory. In Fig. 4.5(a), updating at 1), 2) and 3) are executed at an earlier time during entry, from which we can see a significant difference exists between updated drag and actual drag profile. This is due to the inaccuracy of the updating parameter in Eq. (4.4). Since multiple PC-modes are executed during an earlier phase, the updating parameter only reflects the atmosphere conditions in high elevations, which might be quite different from those in a lower elevation. The updated drag profile can be an inaccurate reference to follow. Figs. 4.5(b) and 4.5(c) show a consistency between updated drag and the actual drag. In Fig. 4.5(c), the PC-modes are executed at a later time, and due to this fact, the vehicle might not have enough control authority to adjust the trajectory. Even the optimal replanned trajectory might not be able to achieve a good performance. Thus the PC-modes cannot be carried out too early or too late so that either the prediction for

future atmosphere is not accurate or the vehicle has not enough control authority left. Finally, three times of updating, 2), 3) and 4) are chosen to be executed onboard during entry. The first execution of the PC-mode can diminish the initial state errors and large atmospheric disturbances in the beginning stage of entry. After the vehicle has flown a certain range, the second and the third executions of the PC-mode can further adjust the trajectory. Fig. 4.6 shows tracking errors between the actual drag that the vehicle experienced and the profiles of reference drag and updated drag in the PC-mode for the same trajectory. From Fig. 4.6 we can see three executions of the PC-mode gradually reduce the tracking errors. Details of the PC-mode techniques are presented in [19]. The PC-mode minimizes the same cost function Eq. (4.2) of the final state based on the current state. Therefore the early phase of tracking a newly replanned trajectory might show significant tracking errors, as shown in the first replanned trajectory in Fig. 4.6, but the tracking errors are reduced gradually as the vehicle is approaching the target.

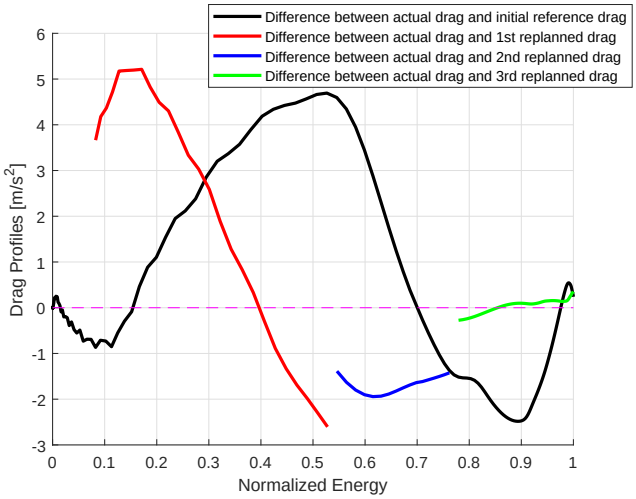


Figure 4.6: Difference between planned/replanned drag profiles and the actual drag profile.

Entry vehicles might reach the entry interface with different flight path angles because of different mission requirements. For shallow entries, the entry will be of longer duration, and guidance performance is degraded due to longer exposure to disturbances.

(Refer to Chapter 6.1 for more details.) Replannings the three bank reversal times is not sufficient to reduce the final delivery errors under this circumstance. To adapt to the actual atmosphere, we optimize the upper boundary for the bank angle. As shown in Fig. 4.2, $|\sigma_2|$ and $|\sigma_3|$ are set as 85^0 to leave some margin for the RT-mode. In the PC-mode, a finite difference method is used to update this upper boundary based on its effects on the predicted downrange errors

$$\begin{aligned} A &= \frac{\Delta DR(\sigma_{max} + \delta\sigma) - \Delta DR(\sigma_{max})}{\delta\sigma} \\ \sigma_{max} &= \sigma_{max} + k\frac{\Delta DR}{A}, \quad \text{if } A \geq \epsilon \\ \sigma_{max} &= \text{sat}[\sigma_{max},\ 75^o,\ 90^o] \end{aligned} \tag{4.5}$$

where σ_{max} is the upper boundary of bank angle and initialed set as $\sigma_{max}(0) = \sigma_2$. $\Delta DR(\sigma_{max})$ is the predicted downrange error based on current bank angle boundary σ_{max}. $\delta\sigma$ is a small perturbation to the bank angle boundary. During the simulation, we have set $\delta\sigma = 0.5^o$. The updating is initiated if the change in downrange error is greater than a certain threshold, which is represented by a small positive number ϵ in Eq.(4.5). $k = 0.9$ is a scale factor to control the updating speed. The updated bank angle boundary is saturated within a certain range to guarantee a positive bank angle and sufficient control capability. This iterative updating scheme is terminated when the predicted downrange error is less than a certain threshold. This updating scheme has significant effects on the guidance performance, since it not only updates the reference trajectory, but also changes the constraints on the tracking effort. Because the iterative updating algorithm takes onboard computational time, it is only used for shallow entry with entry duration longer than 300 seconds when the updating of bank reversal times is not sufficient to improve the guidance performance.

4.2 Reference Trajectory Tracking (RT-mode)

The RT-mode starts when the drag first reaches 0.2 Earth g's [2]. Initially the pre-designed reference trajectory is tracked. When the reference trajectory is updated, the

RT-mode switches to track the updated reference. Due to the fact that drag can be measured accurately onboard, and it is independent of the system model, drag is taken as the primary variable for the RT-mode. The RT-mode is in the form of the model predictive control (MPC) [15, 30]. Given an nonlinear system of the form

$$\begin{aligned} \dot{x} &= F(x) + G(x,u) \\ y &= C(x) \end{aligned} \tag{4.6}$$

where $x \in \mathbb{R}^6$ is the state vector, u is the control vector and y is the output vector. The outputs used in the simulation are drag, drag rate and trajectory length, hence $y \in \mathbb{R}^3$. The objective is to track a reference output $q(t)$ by minimizing the performance index

$$J = \frac{1}{2}e(t+\Delta t)^T Q e(t+\Delta t) + \frac{1}{2}u(t+\Delta t)^T R u(t+\Delta t) \tag{4.7}$$

where

$$e(t+\Delta t) = y(t+\Delta t) - q(t+\Delta t) = \begin{bmatrix} D - D_r \\ \dot{D} - \dot{D}_r \\ S - S_r \end{bmatrix}_{t+\Delta t}$$

and $Q = \text{diag}\{\alpha_1,\ \alpha_2,\ \alpha_3\} \in \mathbb{R}^{3\times 3}$ is a diagonal positive definite matrix, $R \in \mathbb{R}$ is a positive number and Δt is the prediction step size. $D_r,\ \dot{D}_r,\ S_r$ are the reference drag profile and trajectory length profile that are stored during the trajectory planning. Both $y(t+\Delta t)$ and $q(t+\Delta t)$ can be calculated by Taylor series expansion as functions of $y(t),\ q(t),\ \Delta t$ and $u(t)$, to the order determined by the relative degrees of the tracked variables. For drag,

$$\ddot{D} = a + b \cdot cos\gamma \tag{4.8}$$

where

$$\begin{aligned} a =& \dot{D}Vsin\gamma\frac{d\beta}{dh} - D\frac{d\beta}{dh}(D + gsin\gamma)sin\gamma - Dcos^2\gamma\frac{d\beta}{dh}(g - \frac{V^2}{r}) + DV^2sin^2\gamma\frac{d^2\beta}{dh^2} - \frac{2\dot{D}}{V}(D + gsin\gamma) \\ & - \frac{2D}{V^2}(D + gsin\gamma)^2 - \frac{2D\dot{D}}{V} + 4\frac{Dg}{r}sin^2\gamma + \frac{2D}{V^2}gcos^2\gamma(g - \frac{V^2}{r}) - Dcos\gamma(\frac{2g}{V} - V\frac{d\beta}{dh})C_\gamma \\ b =& - DLcos\gamma(\frac{2g}{V^2} - \frac{d\beta}{dh}) \end{aligned} \tag{4.9}$$

For drag and drag rate, we have their relative degrees as $\eta_1 = 2$ and $\eta_2 = 1$, correspondingly. For the trajectory length, we have $e_3 = \Delta S = -\int_0^t \frac{\Delta D}{D_r} V d\tau$, which gives us $\eta_3 = 3$, so we can write the tracking errors as

$$
\begin{aligned}
e_1(t+h) \approx & \Delta D + h\Delta\dot{D} + \frac{h^2}{2}\Delta\ddot{D} \\
e_2(t+h) \approx & \Delta\dot{D} + h\Delta\ddot{D} \\
e_3(t+h) \approx & \Delta S - h\frac{\Delta D}{D_r}V - \frac{h^2}{2}(\frac{\Delta\dot{D}}{Dr}V + \frac{\Delta D}{D_r}\dot{V} - \frac{\Delta D}{D_r^2}V\dot{D}_r) - ... \\
& ... - \frac{h^3}{3!}(\frac{\Delta\ddot{D}}{D_r}V + 2\frac{\Delta\dot{D}}{D_r}\dot{V} - 2\frac{\Delta\dot{D}}{D_r^2} + \frac{\Delta D}{D_r}\ddot{V} - 2\frac{\Delta D}{D_r^2}\dot{V}\dot{D}_r - \frac{\Delta D}{D_r^2}V\ddot{D}_r + 2\frac{\Delta D}{D_r^3}V\dot{D}_r^2)
\end{aligned}
\tag{4.10}
$$

The control input is not optimized, instead, the upper boundaries of the bank angle and its rate are set as hard constraints. Therefore, we can set $R = 0$ and the cost function becomes

$$J = \frac{1}{2}\sum_{i=1}^{3}\alpha_i \cdot e_i^2(t + \Delta t) \tag{4.11}$$

The value of u that minimizes the performance index J is obtained in absence of control saturation by setting $\partial J/\partial u = 0$[16]. This optimal solution, generated by the first order necessary condition produces the analytical expression of the input variable. This closed-form solution saves onboard computational time.

During the RT-mode, three bank reversals are executed based on the reversal times from the reference trajectory. During each bank reversal, the minimum time reversal, subject to the bank angle constraints mentioned above, is executed in open-loop fashion. During the RT-mode, specific energy is the independent variable, which means within each guidance cycle, the reference value and the actual value of the tracked variables are compared at a specific energy value.

4.3 Deployment Position Alignment (DPA-mode)

The range-to-go is decreasing as the vehicle approaches the target during the entry. At a certain Mach number, there is a switch from the RT-mode to the DPA-mode [17], a generalization of the heading alignment logic used for MSL [2], in order to directly focus on reducing the target position delivery error. The DPA-mode forces an exponentially decaying heading error to achieve a better crossrange accuracy, which is similar to the algorithm implemented in MSL. We define the heading angle error as the difference between the current heading angle $\psi(t)$ and the desired heading angle $\psi_d(t)$ that leads to the target following a great circle arc [17]

$$e(t) = \psi(t) - \psi_d(t) \tag{4.12}$$

The desired heading angle $\psi_d(t)$ is obtained using

$$sin\psi_d = \frac{sin\phi_T - cos(S_{\text{togo}}) \cdot sin\phi}{sin(S_{\text{togo}}) \cdot cos\phi} \tag{4.13}$$

where ϕ_T is the target latitude and the vehicle range-to-go, S_{togo} can be computed by Eq. (2.23). The equation for the heading dynamics can be approximated by

$$\dot{\psi}(t) \approx \frac{-L}{Vcos\gamma} sin\sigma \tag{4.14}$$

Applying the feedback linearization and the exponentially decaying heading errors, we can get the heading error dynamics

$$e(t) = e(0) \cdot e^{-k_1 t} \tag{4.15}$$

Thus the heading error dynamics become

$$\dot{e}(t) = -k_1 e(t) \tag{4.16}$$

By substituting Eq. (4.12) and Eq. (4.14) into Eq. (4.16), we can get the guidance law becomes

$$sin\sigma = -\frac{Vcos\gamma}{L}[-k_1(\psi(t) - \psi_d(t)) + \dot{\psi}_d(t)] \tag{4.17}$$

Guidance algorithm switched to this heading alignment phase at a later time during entry, where the atmosphere is significantly denser than the beginning phase. Thus a focus on lateral control while leaving longitudinal dynamics and altitude uncontrolled would potentially result in large delivery errors and low deploy altitude. In [30], the final position alignment takes the altitude control into considerations by analyzing the sensitivity of the final altitudes to the heading error dynamics and updating the dynamics to achieve higher final altitudes. In the DPA-mode presented in this dissertation, a different technique is implemented to account for downrange accuracy and the altitude control is also taken into consideration.

First, the bank angle is constrained within a certain range based on the predicted downrange errors. At each guidance cycle, a prediction of the final downrange is calculated using the current state. If the vehicle is in an undershoot position based on the predicted downrange error, the magnitude of the bank angle will be constrained within 30^o. If the vehicle is in an overshoot position, the bank angle would be constrained within $[30^o,\ 90^o]$. Although the strategy for the overshoot case would potentially decrease the final altitude, it effectively avoids large positive downrange errors, and the final altitude can be guaranteed to be above 6 km by the parachute deployment trigger logic, which will be introduced in the following section. The second strategy is to dynamically choose the switch Mach number based on the downrange error of replanned trajectory. Since the DPA-mode is implemented after the RT-mode, one important criterion is the switch Mach number at which the DPA-mode is initiated. Due to the fact that the DPA-mode focuses on lateral control, switching to the DPA-mode too early might lead to large downrange errors that constraining the bank angle is not enough to compensate, while switching too late will leave not enough control force to reduce the crossrange error. Thus an appropriate switching time needs to be chosen depending on different situations. The strategy used in the simulation is

$$M_{\text{switch}} = 4 + \Delta DR_r[km]/4[km] \tag{4.18}$$

where M_{switch} is the Mach number at which the guidance switches to the DPA-mode and ΔDR_r is the downrange error of the last replanned reference trajectory. If the replanned trajectory has a large downrange error, it means there is not enough control force left to lead the vehicle in the current state to the target, even under the nominal condition. Tracking this inaccurate replanned trajectory would lead to worse performance. Instead, the enhanced DPA-mode uses control laws that reduce heading errors and downrange errors to the actual target in a real-time fashion. An earlier switching to the DPA-mode would be necessary under this circumstance. On the other hand, when ΔDR_r is small, tracking the replanned trajectory would be sufficient, and it is not necessary to switch to the DPA-mode too early. In cases like this, a switching Mach number to the DPA-mode between 4 and 5 could improve the performance at the end of the entry.

4.4 Parachute Deployment Trigger

The entry phase is terminated when the parachute is deployed. The parachute is allowed to deploy safely when the vehicle state is within the parachute deployment box, as shown in Fig. 3.1. The guidance is designed to achieve a high elevation landing with small horizontal errors. Thus those factors need to be taken into account at the parachute deployment level. The trigger used in the guidance is similar to the velocity-constrained range triggering strategy from [32], where the deployment is triggered on the range when the vehicle velocity is below an upper limit V_{high} and above a lower limit V_{low}. In this trigger, instead of using constant upper limit and lower limit as velocity constraints, we use the right and left boundary of the parachute deployment box as shown in Fig. 3.1 as two constraints which are transferred from deploy Mach number and pressure constraints. We also add a hard constraint of minimum altitude to avoid low deploy altitude. This strategy takes all the concerns for safe deployment into consideration, and the deploy box can be adjusted once the safety constraints are changed. The details of the trigger logic are shown in Fig. 4.7.

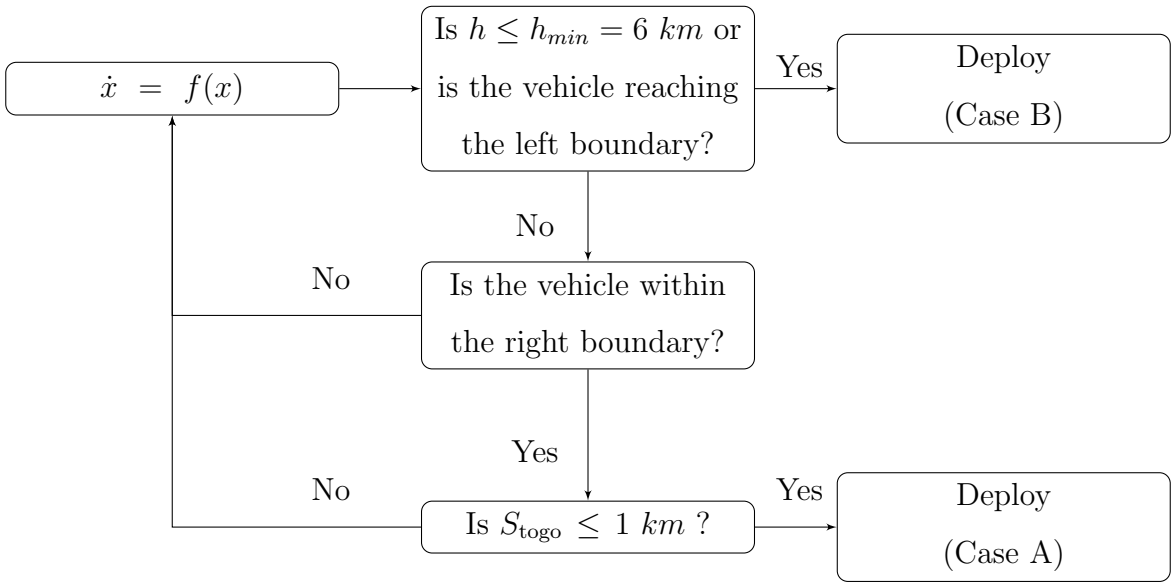


Figure 4.7: Parachute deployment trigger.

In Fig. 4.7, S_{togo} is the vehicle range-to-go to the target. It shows that once the vehicle state is within the right boundary of the parachute deployment box, the trigger begins to examine its altitude and range-to-go to the target. If the vehicle has a positive range-to-go within 1 km or a negative range-to-go, the parachute will be deployed, and the entry phase is terminated. Otherwise, the vehicle keeps flying until the conditions mentioned above are met. The parachute is forced to deploy when the vehicle altitude is lower than 6 km even the deploy constraints are not satisfied or when the vehicle state reaches the left boundary or the deployment box. One kilometer range-to-go to the target is chosen as the deploy threshold in consideration of achieving high elevation landings. This value could be adjusted if more accurate horizontal errors are required. Thus, with this trigger shown in Fig. 4.7, Case A will result in a higher altitude deployment and horizontal error either within 1 km or more overshooting. Case B will result in undershooting with lower altitude, which is the case that guidance should try to avoid.

4.5 Multi-mode Guidance Logic

Multi-mode guidance starts with the RT-mode with the pre-designed reference and drag tracking with a focus on the longitudinal control. During the entry, the PC-mode is executed three times when the first bank reversal starts, when the second bank reversal starts and when the second bank reversal is accomplished. In the PC-mode, only one parameter is updated while the other parameters are adjusted correspondingly. Once the first two PC-modes are accomplished, the guidance switches to the BR-mode, while after the completion of the last PC-mode, the guidance switches back to the RT-mode. The BR-mode is initiated three times, based on the updated bank reversal times. After each BR-mode, the guidance switches back to the RT-mode. The DPA-mode is triggered once the Mach number drops below a certain threshold, which is dynamically determined by the updated reference trajectory. The DPA-mode is terminated by the parachute deployment. The trigger to deploy the parachute considers the constraints on the velocity, altitude and horizontal range-to-go. Once the constraints are met, the parachute is deployed at the earliest time to achieve the high elevation deployment. The multi-mode guidance operates in a single mode at a time, and each mode is designed to achieve the two objectives of high elevation landing and small horizontal error using different strategies.

Chapter 5

State Estimation

In this chapter, we investigate the use of a state estimator to reduce the navigation error, without requiring additional navigation sensors beyond the IMU. In particular, the unscented Kalman filter is considered [35]. The controllability and observability is analyzed to understand the system composed of the dynamics, the control input, and the measurements. Next the UKF is formulated. The performance of the UKF is then assessed and compared with that of conventional inertial navigation and a particle filter.

5.1 Controllability and Observability of the Linearized Entry Dynamics

The entry dynamics, shown in Eq.(2.3) and Eq.(2.7), can be summarized as a nonlinear and non-affine system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, u) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{5.1}$$

By linearizing the model around the reference trajectory, we can get

$$\begin{aligned} \delta\dot{\mathbf{x}} &= \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta u \\ \delta\mathbf{y} &= \mathbf{C}\delta\mathbf{x} \end{aligned} \tag{5.2}$$

where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_r$, $\delta u = u - u_r$ and $\delta\mathbf{y} = \mathbf{y} - \mathbf{y}_r$ are the linearized errors. $A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_r}$, $B = \frac{\partial \mathbf{f}}{\partial u}|_{\mathbf{x}=\mathbf{x}_r}$ and $C = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_r}$ are Jacobian matrices of the dynamic model and measurement model to the state vector and the input variable, evaluated at the reference values. The linearized system is controllable and observable at time t_0 if and only if there exist $t > t_0$ and the controllability and observability Grammians, as defined as

$$\begin{aligned} P &= \int_{t_0}^{t} e^{A\tau} B B^T (e^{A\tau})^T d\tau \\ Q &= \int_{t_0}^{t} (e^{A\tau})^T C^T C e^{A\tau} d\tau \end{aligned} \tag{5.3}$$

are nonsingular. Based on the controllability Grammian and observability Grammian, we would like to consider a basis where the system is balanced, such that we are able to detect the state variable which is difficult to control and observe. We transformed the two Grammians through balanced realization.

$$\begin{aligned} \hat{P} &= T_{\text{bal}} P T_{\text{bal}}^T \\ \hat{Q} &= T_{\text{bal}}^T Q T_{\text{bal}} \end{aligned} \tag{5.4}$$

where the balancing transformation T_{bal} can be computed as

$$\begin{aligned} P &= UU^T \\ U^T Q U &= K \Sigma^2 K^T \\ T_{\text{bal}} &= \Sigma^{\frac{1}{2}} K^T U^{-1} \end{aligned} \tag{5.5}$$

Applying the balancing transformation, we have obtained $\hat{P} = \hat{Q} = \mathbf{diag}(\sigma_1,\ \sigma_2,\ ...,\ \sigma_n)$. For entry dynamics with $\mathbf{x} \in \mathbb{R}^6$, we would expect all the six eigenvalues of $\hat{P}$ are positive to show full controllability and observability. Simulation testings are conducted to generate the balanced controllability Grammian ($\hat{P}$) for 200 trajectories, and the histories of the eigenvalues of $\hat{P}$ are shown in Fig 5.1.

All the results of the simulations shown in this section are run on a small sample space of 200 trajectories. Preliminary results have shown that a sampling of 200 trajectories is sufficient to generate similar results with the same conclusion as a large scale

sampling of 1000 trajectories that is used in the assessment of the guidance approach, as shown in Chapter 7.

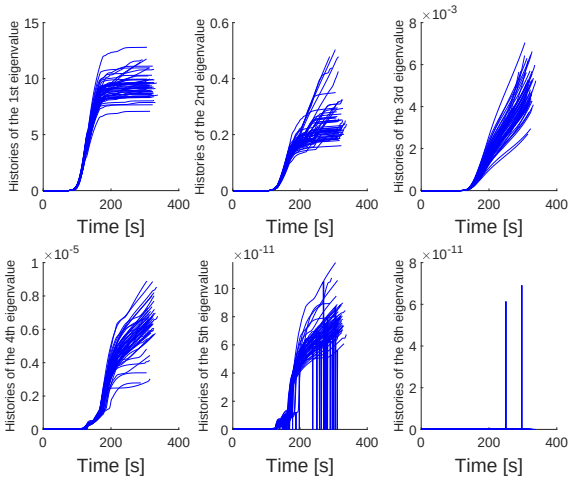


Figure 5.1: Histories of eigenvalues of the transformed controllability Grammians.

Fig. 5.1 shows that the third to the sixth eigenvalues of the Grammians are close to zeros. Therefore, we can conclude that with the linearized entry dynamics, the four states $\phi,\ V,\ \gamma,\ \psi$ are, at the same time difficult to reach and observe. Based on the numerical results, we can conclude that a linear or linearized estimator might not be capable of inferring the values of the state vector and generating accurate estimates by using only the measurements of drag and lift accelerations.

5.2 Nonlinear Controllability and Observability

In this section, the nonlinear controllability and observability of the entry dynamics have been analyzed using the Lie algebra. For the nonlinear and non-affine entry dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, u) \tag{5.6}$$

the system is locally controllable if the controllability matrix

$$\mathcal{C} = [G,\ \mathbf{ad}_F G,\ \mathbf{ad}_{F^2} G,\ ...,\ \mathbf{ad}_{F^{(n-1)}} G] \tag{5.7}$$

where $F(x) = \mathbf{f}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_r} + \frac{\partial \mathbf{g}(\mathbf{x},u)}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_r}$ and $G(x) = \frac{\partial \mathbf{g}(\mathbf{x},u)}{\partial u}|_{\mathbf{x}=\mathbf{x}_r}$ with $x \in \mathbb{R}^n$ has n linearly independent columns [33], . The notation $\mathbf{ad}_F G$ stands for the Lie bracket of $\mathbf{f}$ and $\mathbf{g}$, which is another vector field defined by

$$\mathbf{ad}_F G = [F,\ G] = \nabla G \cdot F - \nabla F \cdot G \tag{5.8}$$

and

$$\mathbf{ad}_{F^{n+1}} G = [F,\ \mathbf{ad}_{F^n} G] \tag{5.9}$$

The nonlinear system is locally observable if the observability matrix

$$\mathcal{O} = [\mathbf{h},\ \mathbf{ad}_{\mathbf{f}}\mathbf{h},\ \mathbf{ad}_{\mathbf{f}^2}\mathbf{h},\ ...,\ \mathbf{ad}_{\mathbf{f}^{(n-1)}}\mathbf{h}] \tag{5.10}$$

has the rank of n [34]. For Mars entry dynamics with a six-dimensional system, the Lie brackets and Lie derivatives up until the 5th order have to be computed. Calculating high order Lie brackets is computationally expensive, especially for Mars entry dynamics which is a highly nonlinear model. Therefore, the quadratic approximations of the Lie derivatives are employed to reduce the complexity of the computations of controllability and observability matrices [34]. For the observability matrix, the $(n+1)$th order Lie derivative can be approximated around the current state $\mathbf{x}_r$ as

$$\mathbf{ad}_{\mathbf{f}^{(n+1)}}\mathbf{h} = \mathbf{ad}_{\mathbf{f}^n}\mathbf{h} + J_L^n(\mathbf{x}-\mathbf{x}_r) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_r)^T H_L^n(\mathbf{x}-\mathbf{x}_r) \tag{5.11}$$

where J_L^n and H_L^n refer to the Jacobian and Hessian matrix of $\mathbf{ad}_{\mathbf{f}^n}\mathbf{h}$ at the current state $\mathbf{x}_r$, respectively. Moreover, the term $\mathbf{f}(\mathbf{x})$ is linearized as

$$\mathbf{f} = \mathbf{f}_r + J_{\mathbf{f}}(\mathbf{x}-\mathbf{x}_r) \tag{5.12}$$

in which $\mathbf{f}_r$ is the evaluation of $\mathbf{f}$ at $\mathbf{x}_r$, and $J_{\mathbf{f}}$ is the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}_r$. Two consecutive orders of the Lie derivative have the relationship as

$$\mathbf{ad}_{\mathbf{f}^{(n+1)}}\mathbf{h} = \nabla \mathbf{ad}_{\mathbf{f}^n}\mathbf{h} \cdot \mathbf{f} \tag{5.13}$$

Substituting Eq.(5.11) and Eq.(5.12) into Eq.(5.13), we can get the following equations

$$\begin{aligned} J_L^{n+1} &= J_L^n J_{\mathbf{f}} + \frac{1}{2}([H_L^n + (H_L^n)^T]\mathbf{f}_r)^T \\ H_L^{n+1} &= \frac{1}{2}[H_L^n + (H_L^n)^T]J_{\mathbf{f}} \end{aligned} \tag{5.14}$$

and the observability matrix can be computed as

$$\mathcal{O} = [(J_L^0)^T,\ (J_L^1)^T,\ (J_L^2)^T,\ ...,\ (J_L^{(n-1)})^T]^T \tag{5.15}$$

We can use Eq.(5.14) to iteratively compute the Jacobian matrix and Hessian matrix of the nth order Lie derivative, and use the Jacobian matrix as an approximation of the Lie derivative. The same procedure is applied to the computations of the controllability matrix. This 2nd order approximation has greatly reduced the computational time.

For entry dynamics with the state vector $\mathbf{x} \in \mathbb{R}^6$, $rank(\mathcal{C}) = 6$ and $rank(\mathcal{O}) = 6$ is required for full controllability and observability. Two hundred trajectories are generated to compute the controllability and observability matrices using the 2nd order approximations. The results show that the controllability matrix has $rank(\mathcal{C}) = 6$, where the observability matrix has $rank(\mathcal{O}) = 5$. This concludes that the entry dynamics are locally controllable, but not completely observable. Given that the observability matrix has the rank of 5, we can find the non-observable variable by removing a specific column in this matrix and calculating the rank of the reduced matrix [34]. By eliminating the ith column in matrix $\mathcal{O}$ and observing the change of the rank condition, we have determined that the non-observable variable is θ, the vehicle longitude.

A system is said to be observable if the current state can be determined in finite time from the measured quantities. The IMU measurements of drag and lift accelerations, as shown in Eq.(2.8) are functions of the vehicle altitude and velocity. Both of these two state variables appear in the dynamics of the entire six state variables. Moreover, the only non-observable state variable, the vehicle longitude does not influence the dynamics of any of the state variables. This observability condition indicates that the

other state variables can be inferred from knowledge of its external outputs. Based on the controllability and observability analysis of the entry dynamics, we conclude that a state estimator such as the UKF which accounts for the nonlinearity of the dynamics should be investigated and implemented.

5.3 Unscented Kalman Filter Enhanced Navigation Scheme

The general unscented Kalman filter propagates the predicted mean and covariance through an unscented transformation [35] and may produce a more accurate state estimate for a system with the nonlinear process and measurement models than the extended Kalman filter. During entry, the disturbances the vehicle experiences such as density modeling errors and aerodynamic coefficients uncertainties introduce errors in the nonlinear entry dynamics, as well as in the nonlinear measurement models of drag and lift. Thus the entry state is augmented with uncertain parameters and measurement noise as $\mathbf{x^a} = [\tilde{\mathbf{x}},\ \mathbf{w},\ \mathbf{v}]$, where $\tilde{\mathbf{x}} = \mathbf{x_{nav}}$ represents the estimated state vector of the entry vehicle, and $\mathbf{w} = [\Delta\rho,\ \Delta C_D,\ \Delta C_L]$ represents the uncertain parameters in the dynamics as process noise. Each of these parameters is modeled as a constant percentage offset of the nominal value

$$\begin{aligned} \tilde{\rho} &= \rho(1 + \Delta\rho) \\ \tilde{C}_D &= C_D(1 + \Delta C_D) \\ \tilde{C}_L &= C_L(1 + \Delta C_L) \end{aligned} \tag{5.16}$$

$\mathbf{v} = [\delta D,\ \delta L]$ represents the measurement noise of drag and lift taken from the IMU. With the three degree-of-freedom modeling, we have assumed that the IMU drift and bias are negligible. (They are modeled in the six degree-of-freedom entry dynamics.) And only the additive white noise contributes to the measurement noise. $\mathbf{w}$ and $\mathbf{v}$ are assumed to be independent of each other and normally distributed [22]

$$E[\mathbf{w}_k\mathbf{w}_j^T] = \mathbf{P_w}\delta_{kj}, \quad E[\mathbf{v}_k\mathbf{v}_j^T] = \mathbf{P_v}\delta_{kj}, \quad E[\mathbf{w}_k\mathbf{v}_j^T] = 0 \tag{5.17}$$

where $\mathbf{P_w}$ and $\mathbf{P_v}$ are covariance matrices for the uncertain parameters vector and measurement noise vector. During each navigation cycle, the augmented state vector and augmented covariance matrix are constructed as

$$\begin{aligned} \tilde{\mathbf{x}}_k^{\mathbf{a}} &= E[\mathbf{x}_k^{\mathbf{a}}] = [\tilde{\mathbf{x}}_k^T \quad 0 \quad 0]^T \\ \mathbf{P}_k^{\mathbf{a}} &= E[(\mathbf{x}_k^{\mathbf{a}} - \tilde{\mathbf{x}}_k^{\mathbf{a}})(\mathbf{x}_k^{\mathbf{a}} - \tilde{\mathbf{x}}_k^{\mathbf{a}})^T] = \begin{bmatrix} \mathbf{P}_k & & \\ & \mathbf{P_w} & \\ & & \mathbf{P_v} \end{bmatrix} \end{aligned} \tag{5.18}$$

Then the sigma points are calculated by the Cholesky decomposition of the covariance matrix and combination with the mean

$$\chi_k^{\mathbf{a}} = [\tilde{\mathbf{x}}_k^{\mathbf{a}} \quad \tilde{\mathbf{x}}_k^{\mathbf{a}} + \sqrt{(n+\lambda)\mathbf{P}_k^{\mathbf{a}}} \quad \tilde{\mathbf{x}}_k^{\mathbf{a}} - \sqrt{(n+\lambda)\mathbf{P}_k^{\mathbf{a}}}] \tag{5.19}$$

where n is the number of state variables, and λ is the scaling factor for the sigma points, which is calculated as

$$\lambda = \alpha^2(n+\kappa) - n \tag{5.20}$$

where α and κ are scaling parameters. α determines the spread of the sigma points and is usually set to a small positive value. It can be tuned based on the propagation of the state vector. κ is a secondary scaling parameter which is usually set to 0 [35].

The UKF can be separated into two steps as the prediction step and measurement update. The prediction step is the propagation of the sigma points through the entry dynamic model

$$\dot{\chi}_{k+1/k}^{\mathbf{x}} = \mathbf{f}(\chi_k^{\mathbf{x}}, \chi_k^{\mathbf{w}}) \tag{5.21}$$

The uncertain parameters are modeled as white noise with zero means, therefore they are assumed constant during the propagation of the sigma points and their covariance matrix $P_{\mathbf{w}}$ is fixed and will not be updated during each navigation cycle. The mean state and covariance during the prediction step can be calculated from the propagated

sigma points as

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1/k} &= \sum_{i=0}^{2n} W_i^{(m)} \chi_{i,k+1/k}^{\mathbf{x}} \\ \mathbf{P}_{k+1/k} &= \sum_{i=0}^{2n} W_i^{(c)} [\chi_{i,k+1/k}^{\mathbf{x}} - \tilde{\mathbf{x}}_{k+1/k}][\chi_{i,k+1/k}^{\mathbf{x}} - \tilde{\mathbf{x}}_{k+1/k}]^T\end{aligned} \tag{5.22}$$

where $W_i^{(m)},\ W_i^{(c)}$ are weighting coefficients defined as

$$\begin{aligned}W_0^{(m)} &= \frac{\lambda}{n+\lambda} \\ W_0^{(c)} &= \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) \\ W_i^{(m)} &= W_i^{(c)} = \frac{\lambda}{2(n+\lambda)}, \quad i = 1, 2, ..., 2n\end{aligned} \tag{5.23}$$

where β is the scaling parameter used to incorporate prior knowledge of the distribution of $\mathbf{x}$. Then the estimated measurements are calculated by applying the predicted mean state to the nonlinear measurement model

$$\begin{aligned}\upsilon_{k+1/k} &= \mathbf{h}(\chi_k^{\mathbf{x}}, \chi_k^{\mathbf{w}}, \chi_k^{\mathbf{v}}) \\ \tilde{\mathbf{y}}_{k+1/k} &= \sum_{i=0}^{2n} W_i^{(m)} \upsilon_{i,k+1/k}\end{aligned} \tag{5.24}$$

Since the uncertain parameters are shown in the nonlinear measurement models, $\chi_k^{\mathbf{v}}$ is considered in the calculation of the estimated measurements. For the measurement update, the estimated state vector and covariance matrix are calculated as

$$\begin{aligned}\mathbf{P}_{yy} &= \sum_{i=0}^{2n} W_i^{(c)} [\upsilon_{i,k+1/k}^x - \tilde{\mathbf{y}}_{k+1/k}][\upsilon_{i,k+1/k}^x - \tilde{\mathbf{y}}_{k+1/k}]^T \\ \mathbf{P}_{xy} &= \sum_{i=0}^{2n} W_i^{(c)} [\chi_{i,k+1/k}^x - \tilde{\mathbf{x}}_{k+1/k}][\upsilon_{i,k+1/k}^x - \tilde{\mathbf{y}}_{k+1/k}]^T \\ \mathbf{K}_{k+1} &= \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} \\ \tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{x}}_{k+1/k} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \tilde{\mathbf{y}}_{k+1/k}) \\ \mathbf{P}_{k+1} &= \mathbf{P}_{k+1/k} - \mathbf{K}_{k+1}\mathbf{P}_{yy}\mathbf{K}_{k+1}^T\end{aligned} \tag{5.25}$$

Within each navigation cycle, the estimated state vector is propagated and updated through these two steps.

Using the UKF, the estimation accuracy is expected to be improved in several aspects. First, other than purely propagating the estimated state variables, an estimation of the non-addictive uncertainties represented by $\chi^{\mathbf{w}}$ in Eq.(5.21), which is generated by the decomposition of $P_{\mathbf{w}}$, has been integrated into the nonlinear entry dynamics and measurement models. Second, the sigma points have been propagated with the actual dynamics without linearizing the dynamics, which improves the filter accuracy. Finally, the state is updated by the measurements of drag and lift from onboard IMU system.

5.4 Analysis of the Unscented Kalman Filter

5.4.1 Bias Analysis

A bias analysis has been conducted for this UKF-based estimator. An estimator is said to be unbiased if the expected value of an estimator is equal to the actual value, i.e., $E(\tilde{\mathbf{x}}) = \mathbf{x}$. In order to compute the expectation of different realizations, the simulation testing of 200 trajectories is implemented. The expectations of each estimated state variable are calculated and compared with the actual values. The time histories of the biases, $E(\tilde{\mathbf{x}}_i) - \mathbf{x}_i,\ i = 1, 2, ...7$ are shown in Figs. 5.2 and 5.3.

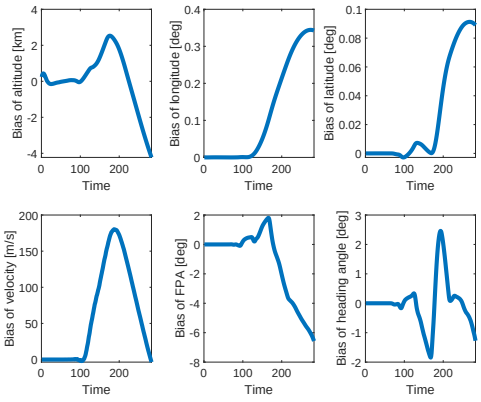


Figure 5.2: Time histories of the biases of state variables.

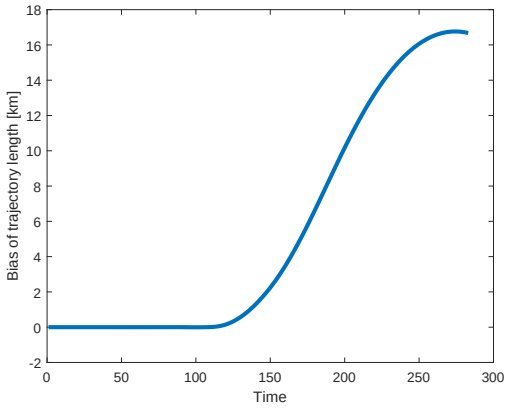


Figure 5.3: Time history of the bias of trajectory length.

From Figs. 5.2 and 5.3 we can see there are non-zero biases during much of the trajectory. Except for velocity, the biases of all the state variables exist until the end of the trajectory, and some of them are increasing along the trajectory. In [36], the author wrote "From a Bayesian perspective, the principle of unbiasedness is reasonable in the limit of large samples, but otherwise it is potentially misleading." Actually, most Bayesians are rather unconcerned about the unbiasedness of their estimates. One consequence of the biased estimate is that the Cramer-Rao bound (CRB) may not accurately reflect the estimation accuracy. Therefore, the CRB of the UKF-enhanced navigation is computed to verify this conclusion.

5.4.2 Cramer-Rao Bound

The Cramer-Rao lower bound states that for an unbiased estimator, under general conditions, there is a matrix P_{cr} such that

$$P \geq P_{cr} \tag{5.26}$$

which means $(P - P_{cr})$ is a positive semidefinite matrix [40]. The CRB matrix P_{cr} is defined as

$$P_{cr} = -\{E(\frac{\partial^2 \ln p(\mathbf{y}, \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T})\}^{-1} \tag{5.27}$$

where the $\ln p(\mathbf{y}, \mathbf{x})$ is the measurement log-likelihood function. However, for a biased estimator, there might be a lower mean squared error than the CRB [40]. We can compute the CRB using the Fisher information matrix (FIM). The FIM, which is defined as [40]

$$J = -P_{cr}^{-1} \tag{5.28}$$

can be obtained numerically by the simulation testing using the simultaneous perturbation method [39]. In this section, we only considered the perturbations in vehicle altitude and velocity. Thus we can define $\delta\mathbf{x} = [\delta h,\ \delta V]^T$ as the perturbation vector, which is sampled in the same distribution as the initial knowledge error. The FIM is estimated as the negative average of the numerically computed Hessian matrix

$$\begin{aligned} \bar{H}_n &= \frac{1}{2}\{\frac{\delta G}{2}[\delta h^{-1},\ \delta V^{-1}] + (\frac{\delta G}{2}[\delta h^{-1},\ \delta V^{-1}])^T\} \\ J &= -\frac{1}{N}\sum_{n=1}^{N} \bar{H}_n \end{aligned} \tag{5.29}$$

where

$$\begin{aligned} \delta G &= G(\mathbf{x} + \delta\mathbf{x}) - G(\mathbf{x} - \delta\mathbf{x}) \\ G(\mathbf{x} \pm \delta\mathbf{x}) &= \frac{L(\mathbf{x} \pm \delta\mathbf{x} + \delta\tilde{\mathbf{x}}) - L(\mathbf{x} \pm \delta\mathbf{x} - \delta\tilde{\mathbf{x}})}{2} \cdot [\delta h^{-1},\ \delta V^{-1}] \end{aligned} \tag{5.30}$$

and $L(\mathbf{x} + \delta\mathbf{x}) = \ln p(y_k|\mathbf{x}_k + \delta\mathbf{x}) = -\frac{1}{2}(y_k - \bar{y}_k)^T P_{\mathbf{v}}^{-1}(y_k - \bar{y}_k)$ is the measurement log-likelihood function where y_k is the measurements of drag and lift accelerations given the state vector $\mathbf{x} + \delta\mathbf{x}$, and $\bar{y}_k$ represents the mean of the measurement vector [40]. $\delta\mathbf{x}$ and $\delta\tilde{\mathbf{x}}$ are identically independently distributed. The Cramer-Rao bound P_{cr} can be then computed using Eq.(5.28). Then the performance of the estimator can be assessed by comparing the error covariance P of UKF-enhanced navigation with the Cramer-Rao bound of the state vector $\bar{P}$, which can be calculated using the P_{cr} as

$$\bar{P} = (\frac{\partial \mathbf{f}}{\partial \mathbf{x}})P_{cr}(\frac{\partial \mathbf{f}}{\partial \mathbf{x}})^T \tag{5.31}$$

where $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ is the Jacobian matrix of the entry dynamics with respect to the perturbation vector $\delta\mathbf{x}$. We have again implemented the simulation of 200 trajectories to compute the numerical approximation of $\bar{P}$, and the comparison of $\bar{P}$ and P is shown in Fig. 5.4.

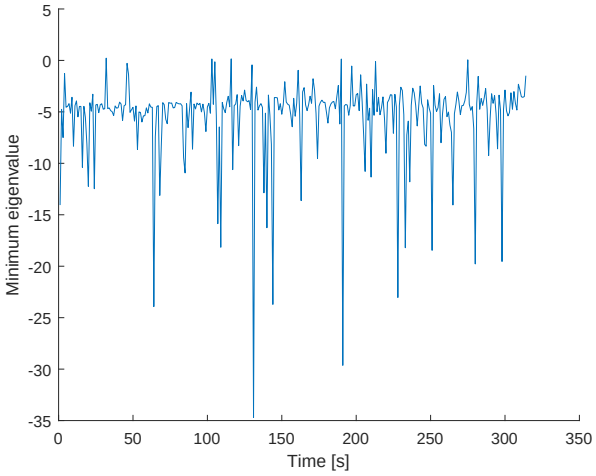


Figure 5.4: History of the minimum eigenvalues of $(P - \bar{P})$

Fig. 5.4 shows the history of the minimum eigenvalues of $(P-\bar{P})$, from which we can see the this UKF-based estimator has shown a smaller variance than the CRB, which verifies the conclusions in Section 5.4.1 that this estimator for Mars entry dynamics is biased and not consistent.

This biased UKF estimator is selected as the state estimator for the proposed Mars entry guidance. As mentioned in Subsection 5.4.1, the estimator bias would not be considered as a potential factor to degrade the estimation performance. Instead, the mean squared error (MSE) is usually used to assess the quality of an estimator.

5.4.3 Comparison with Conventional Inertial Navigation (CIN)

The general Kalman filter is constructed by minimizing the mean squared error

$$\text{MSE} = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{e}_k^j)^2 \tag{5.32}$$

where $\mathbf{e}_i^j = \tilde{\mathbf{x}}_i^j - \mathbf{x}_i^j$ is the estimation error of the ith state variable in the jth simulation run and N is the number of the samples. A comparison of the mean square errors of the UKF-enhanced navigation and the conventional inertial navigation is shown in

this subsection. The time histories of the MSE of each state variable, using the same simulation setup in the previous subsection are shown in Fig. 5.5, where the blue curves represent the MSE histories of the CIN and the red ones represent the MSE histories of the UKF-enhanced navigation.

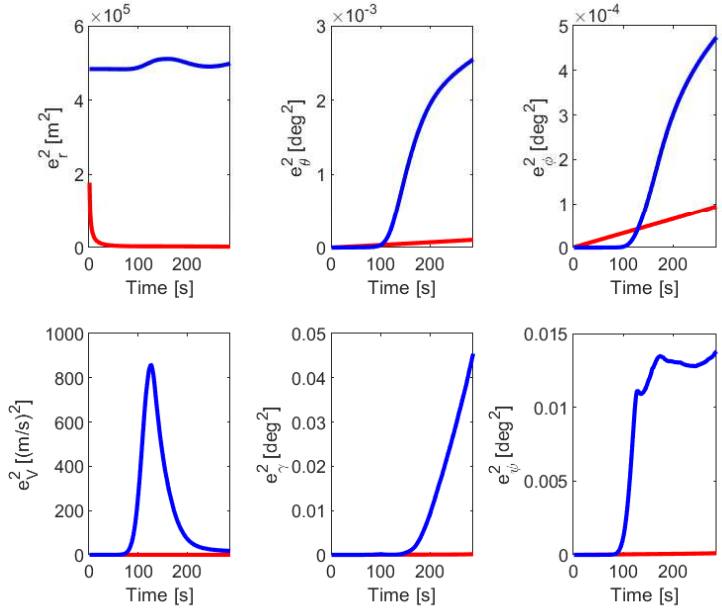


Figure 5.5: Comparison of mean squared errors of CIN and UKF-enhanced navigation.

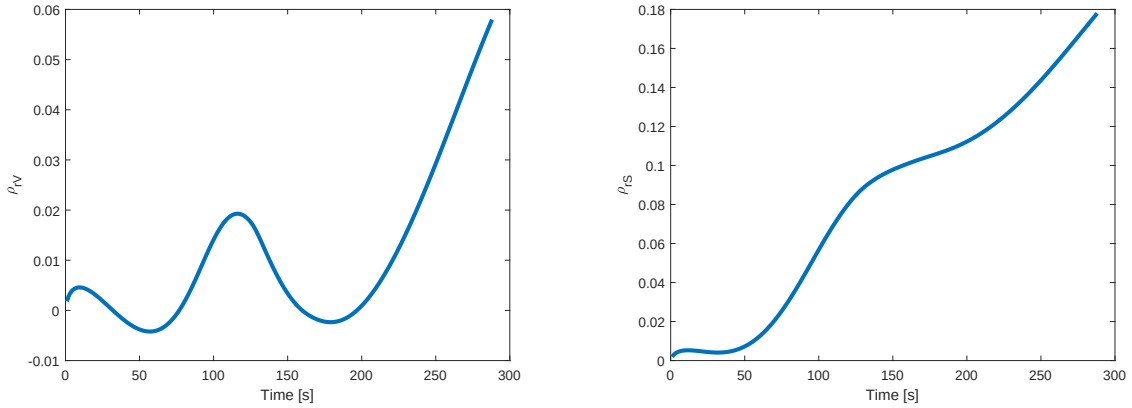
From Fig. 5.5 we can see that with the CIN, the MSE of the longitude, latitude, flight path angle and heading angle are increasing along the trajectory. The MSE of the altitude stays rather constant. The MSE of the velocity is reduced to a small value at the end of the trajectory, but the errors in the middle of the trajectory impact the estimation of other state variables, for that the velocity contributes to their dynamics. Furthermore, we can see the UKF-enhanced navigation reduces the mean squared errors of all the state variables. The improvement is particularly significant in the vertical plane.

In order to determine the effects of the estimation error in one state variable on

other state variables, the correlations between the estimation of several state variables with UKF-enhanced navigation are computed. The correlation coefficient between two estimated state variables $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ are defined as

$$\rho_{\tilde{\mathbf{x}}_i,\tilde{\mathbf{x}}_j} = \frac{\text{cov}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)}{\sigma_{\tilde{\mathbf{x}}_i}\sigma_{\tilde{\mathbf{x}}_j}} \tag{5.33}$$

The correlation coefficients of the altitude and velocity and the altitude and trajectory length are computed in particular to assess the correlations between the state variables that define the vehicle vertical and horizontal performance. The average correlations of the 200 trajectories are shown in Fig. 5.6, from which we can see that both mean correlation coefficients stay within the range of 0 to 0.2. This indicates that there are positive correlations between $(r,\ V)$ and $(r,\ S)$. And as the entry vehicle approaches the target, the correlations are increasing. This correlation between the horizontal and vertical state variables indicates that the estimation error in one state variable can affect the error of the other state variable. On the other hand, these state variables have weak correlations, which implies that the estimation error in one state variable would not have significant effects on the other one.



(a) Mean correlation between altitude and velocity

(b) Mean correlation between altitude and trajectory length

Figure 5.6: Histories of mean correlation coefficients between state variables.

5.4.4 Comparison with the Particle Filter Estimator

Particle filtering is a general Monte Carlo method based on particle representations of the probability densities. Similar to the UKF, these points are all propagated through the nonlinear dynamics. Therefore, this filter has been shown to give better performance than a standard EKF for some problems [41]. The general particle filter algorithm is shown in Table 5.1, where N_s is the number of particles, and each particle $\mathbf{x}_i$ is associated with a weight w_i. $\mathbf{q}(\mathbf{x}_k^i|\mathbf{x}_{k-}^i, \mathbf{y}_k)$ is called the importance density. And usually we choose the importance density to be the prior, $\mathrm{w}_k^i = \mathrm{w}_{k-1}^i \cdot \mathbf{p}(\mathbf{y}_k|\mathbf{x}_k^i)$ [41]. This is the most common choice of importance density since it is intuitive and simple to implement. There are many other densities that can be used, and choosing the correct one is a crucial design step in the design of a particle filter.

Table 5.1: Particle filter algorithm

- For i=1:N_s
  - -Draw $\mathbf{x}_k^i = \mathbf{p}(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$
  - -Assignment weight $\mathrm{w}_k^i = \mathrm{w}_{k-1}^i \cdot \frac{\mathbf{p}(\mathbf{y}_k|\mathbf{x}_k^i)\mathbf{p}(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{\mathbf{q}(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\mathbf{y}_k)}$
- End for
- Normalize the weight $w_i = w_i / \sum_{i=1}^{N_s} w_k^i$
- Calculate the effective sample size $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2}$
- If $\hat{N}_{eff} < N_t$
  - -Resampling using SIR
- End if

In the algorithm, $\hat{N}_{eff}$, the effective sampling size is approximated as the measurement of the filter degeneracy. Whenever a significant degeneracy is observed, i.e., when $\hat{N}_{eff}$ is smaller than some threshold (N_t), a resampling is initiated to reduce the effects of degeneracy.

The basic idea of resampling is to eliminate particles that have small weights and to

concentrate on particles with large weights. A set of new particles is selected based on the sampling importance resampling algorithm, as shown in Table 5.2. This algorithm is simple to implement since it only requires information from $\mathbf{p}(\mathbf{y}_k|\mathbf{x}_k^i)$. This also implies the particles are updated without directly taking into account the information from the measurements $\mathbf{y}_k$. A small scale simulation testing is implemented to assess the

Table 5.2: Sampling importance resampling (SIR) algorithm

•Construct the cumulated density function (CDF) of the weights
•For i=1:N_s
-Generate a uniformly distributed random number $u \sim U[0,1]$
-Find the first index j in the CDF of the weights that has a larger probability than u
-Select the jth particle as one of the new samplings
•End if
•$w_i = 1/N_s$

performance of this particle filter estimator mentioned above in Mars entry guidance. The preliminary results have shown some improvement in reducing navigation errors, compared with the conventional inertial navigation. However, several techniques in the filter design need to be explored before it can generate better performance. As mentioned above, the selections of the importance density function and resampling algorithm need to be tuned to match the special structure of the entry dynamics. A more detailed assessment of the particle filter will be conducted in the future work.

Chapter 6

Guidance and Navigation Testing Conditions

In the next chapter, the performance of the proposed entry guidance and navigation strategy is assessed. In the present chapter, we describe the sensor, modeling, and initial entry state errors that are considered in the simulation. Also the test cases are given and the entry simulation procedure is indicated.

6.1 Entry Conditions and Targets

The Viking mission, launched in 1975, developed the 70-degree sphere-cone aeroshell [42]. A scaled variant of the aeroshell has been employed on every subsequent Mars landing mission due to its relatively high hypersonic drag coefficient. The MSL-type of entry vehicle structure is designed for 15-g peak loads during entry [2]. Acceleration loads are primarily determined by the initial entry flight-path angle. The vehicle will experience higher acceleration loads for steeper entry. A comparison of g-loads for entry flight path angles $\gamma_0 = -12^0$ and $\gamma_0 = -14.15^0$ are shown in Fig. 6.1. Except for the entry flight path angle, both trajectories are generated under the same nominal conditions using the high elevation trajectory planner. It shows that the steep entry, employed in MSL, results in a higher peak g-load. Furthermore, the entry flight-path

angle has a similar impact on the heat rate. For future crewed entry vehicles, a smaller g-load limit of 5-g is expected. Therefore, a shallow entry is one of the possible strategies for future Mars missions to avoid exceeding the constraints on acceleration load and heat rate. On the other hand, steeper entries result in shorter trajectories with better landing accuracy. Until today, the U.S. robotic Mars missions have used different entry flight path angles from -12^o to -17^o, with ballistic coefficients varying from 64 to 142 kg/m^2 [43]. In this dissertation, the guidance approach is assessed for an MSL-type of entry vehicle with the ballistic coefficient varying around 135 kg/m^2, for a range of entry flight path angles $\{-12^o,\ -13^o,\ -14^o,\ -15^o\}$.

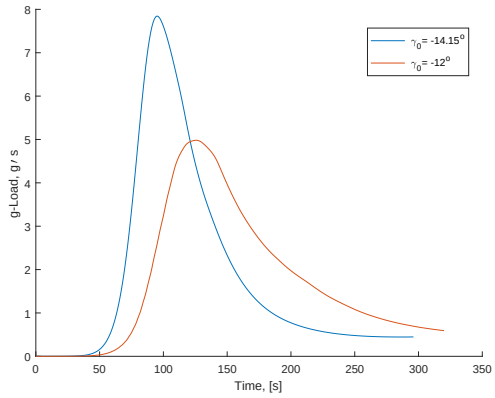
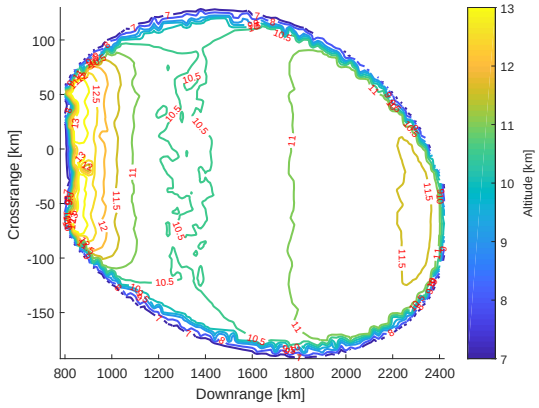
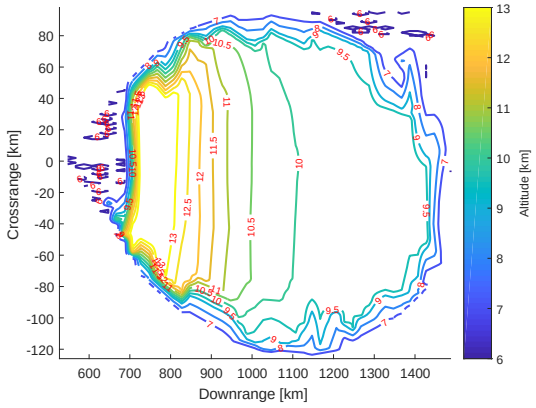


Figure 6.1: g-loads performance for different entry conditions

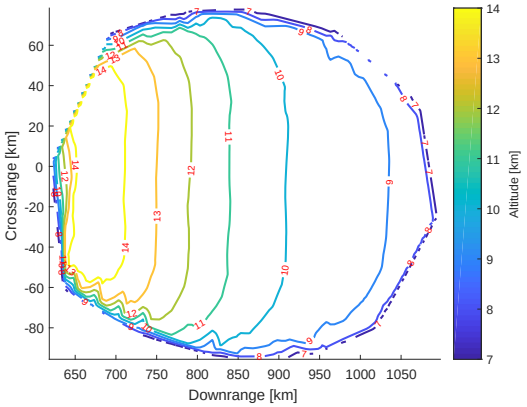
For each initial flight path angle, with the other initial conditions fixed and the assumed vehicle model, there is a reachable set of potential parachute deployment states [29]. The longitude-latitude boundary points for the surface projection of this reachable set are generated by maximizing or minimizing the final longitude and latitude. For a given $[\theta,\ \phi]$ pair, there is more than one feasible trajectory within the reachable set. We determine the maximum achievable altitude using GPOPS− II [44]. For each entry condition mentioned above, the corresponding reachable set is generated. The results are shown in Fig. 6.2.



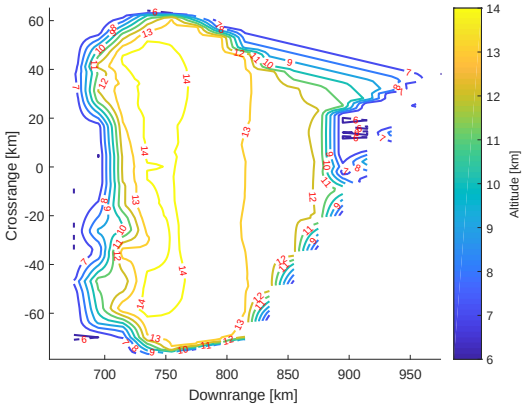
(a) Reachable set for $\gamma_0 = -12^o$



(b) Reachable set for $\gamma_0 = -13^o$



(c) Reachable set for $\gamma_0 = -14^o$



(d) Reachable set for $\gamma_0 = -15^o$

Figure 6.2: Reachable sets for four entry flight path angles.

Fig. 6.2 shows the boundary points in downrange and crossrange pairs $(DR,\ CR)$, as defined in Section 2, for four entry flight path angles, and the maximum achievable altitude contours are indicated. The corresponding velocities are compatible with the parachute box constraint. By comparing these four reachable sets, we can see that most feasible trajectories are much longer for shallow entries than for steep entries. And from the shallow entry to steep entry, the size of the reachable set decreases. For the entry condition with $\gamma_0 = -15^o$, the feasible downrange only varies within 200 km.

The nominal MSL trajectory has a crossrange of -5 km. The targets for the different flight path angles are chosen with the same crossrange. Furthermore, the target location should not be too close to the edge of the reachable set. Otherwise, there would be the possibility of the target being unreachable due to modeling errors. Therefore from each reachable set, the crossrange is fixed at -5 km, and a downrange with a relatively high elevation is chosen as the target point. The target downrange and crossrange for the four entry angles are shown in Table 6.1. Fig. 6.3 shows a comparison between two trajectories in the longitude and latitude plane with $\gamma_0 = -12^o$ and $\gamma_0 = -14.15^o$. The lengths of these two trajectories are 914 km and 755 km, and the durations of the trajectories are 321 s and 285 s, correspondingly. Therefore we can see the shallow entry results in a much longer trajectory.

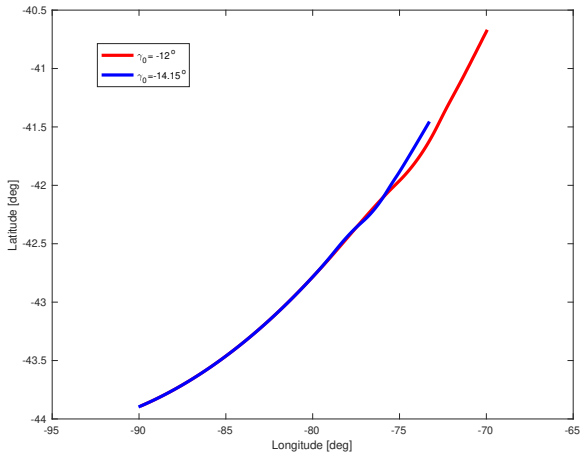


Figure 6.3: Trajectories with $\gamma_0 = -12^o$ and $\gamma_0 = -14.15^o$

Table 6.1: Targets for different entry conditions

Entry flight path angle	$\gamma_0 = -12^o$	$\gamma_0 = -13^o$	$\gamma_0 = -14^o$	$\gamma_0 = -15^o$
Target downrange[km]	900	750	680	740
Target crossrange[km]	-5	-5	-5	-5
Maximum final altitude[km]	13.8	14.2	14.9	14.5

6.2 Simulation Flow

Combining the actual state, estimated state, uncertain parameters and measurement noise vector, the final state variable is defined as $\mathbf{X}_{\text{aug}} = [\mathbf{x},\ \ \tilde{\mathbf{x}},\ \mathbf{w},\ \mathbf{v}]_{1\times 17}$, and $\tilde{\mathbf{x}} = [\tilde{r},\ \tilde{\theta},\ \tilde{\phi},\ \tilde{V},\ \tilde{\gamma},\ \tilde{\psi},\ \tilde{S}]$ represents the estimated state vector. The nominal initial state is disturbed by control errors and knowledge errors

$$\mathbf{x}_0 = \mathbf{x}_{0,\text{nominal}} + \Delta\mathbf{x}_{0,\text{control}}$$

$$\tilde{\mathbf{x}}_0 = \mathbf{x}_0 + \Delta\mathbf{x}_{0,\text{knowledge}}$$

$$X_{0,\text{aug}} = [\mathbf{x}_0,\ \ \tilde{\mathbf{x}}_0,\ \ \mathbf{0}_{1\times 3},\ \ \mathbf{0}_{1\times 2}]$$

During entry, the IMU measures the drag and lift accelerations with white noise

$$\tilde{D} = D + \delta D$$

$$\tilde{L} = L + \delta L$$

where D and L are the actual drag and lift accelerations. The guidance algorithm described in Chapter (3) operates on the measurements and the current state estimation to generate the commanded bank angle σ_c

$$\underset{cos\sigma}{\arg\min}\ \ J = f(\tilde{\mathbf{x}},\ \tilde{D},\ \tilde{L})$$

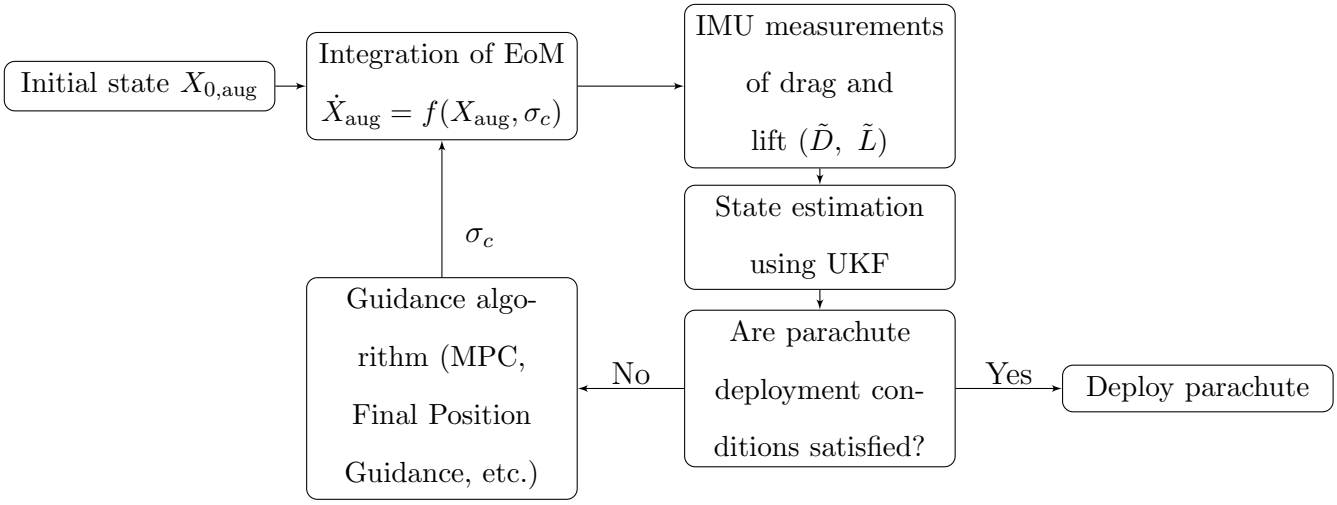
Both of the actual state and the estimated state are integrated simultaneously

$$\dot{\mathbf{x}} = f_1(\mathbf{x}, D, L, \sigma_c(\tilde{\mathbf{x}}))$$

$$\dot{\tilde{\mathbf{x}}} = f_2(\tilde{\mathbf{x}}, \tilde{D}, \tilde{L}, \sigma_c(\tilde{\mathbf{x}}), \epsilon^o)$$

where f_1 represents the entry dynamics with perfect onboard knowledge, as shown in Eq.(2.3) and Eq.(2.7), f_2 represents the entry dynamics with attitude initialization errors ϵ^o, as shown in Eq.(2.3) and Eq.(2.9). The navigation frequency, which is dictated by the IMU measurement frequency, is usually higher than guidance frequency. Within each guidance cycle, there are multiple estimation cycles to generate estimates with more accuracy.

At parachute deployment, which is considered the end of the entry phase, the vehicle will have the actual final state $\mathbf{x}_f$ and the estimated final state $\tilde{\mathbf{x}}_f$. Substituting them into Eq.(2.24), we obtain the actual horizontal error (DR, CR) and the estimated horizontal error $(\tilde{DR}, \tilde{CR})$, and the estimated horizontal error is the one of interest regarding the guidance performance. The simulation flow is shown in the following chart



6.3 Simulation Environment Setup

Guidance performance is assessed on an MSL-type vehicle, with reference area $s = 15.9\ m^2$ and mass $m = 2804\ kg$. The nominal aerodynamic coefficients profiles are shown in Fig. 6.4. It shows the lift-to-drag ratio of a typical MSL-type vehicle is in the range from 0.24 to 0.31.

During the RT-mode, bank angle and its derivatives are constrained by $\sigma_{max} = 90\ deg$, $\dot{\sigma}_{max} = 20\ deg/s$ and $\ddot{\sigma}_{max} = 5\ deg/s^2$. The initial entry state, $\mathbf{x}_{0,nominal}$, used for nominal conditions of the numerical results is

$$\begin{bmatrix} r_0 & \theta_0 & \phi_0 & V_0 & \gamma_0 & \psi_0 \end{bmatrix}^T = \begin{bmatrix} 3520km & -90.07^o & -43.90^o & 5.505km/s & \gamma_0 & 4.99^o \end{bmatrix}^T \tag{6.1}$$

Three main contributors of modeling errors implemented in the simulation are at-

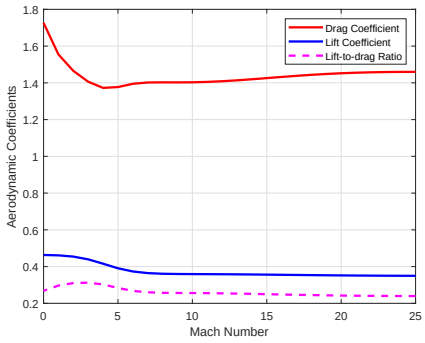


Figure 6.4: Aerodynamic coefficients profiles.

mospheric density modeling errors, aerodynamics coefficients uncertainties and initial delivery errors. A high-fidelity atmosphere model, the Mars Climate Database (MCD) is implemented in the simulation. The MCD generates disturbance profiles in a combination of large-scale and small-scale variability models. Large-scale variability model uses randomly chosen sets of Empirical Orthogonal Functions, which represent environmental quantities with larger variances that are extracted from the datasets over several Martian years. The small-scale variability model is based on the propagation of vertical gravity waves. The gravity wavelength is set as a fixed number for each simulation and in total follows a uniform distribution within the range from 2 km to 30 km to include all the possibilities. One thousand profiles are collected at the target location at a fixed local time of a specific season when $Ls = 0$, where Ls is the Martian solar longitude. Boundaries of one thousand profiles of the density disturbances are shown in Fig. 4.4. One may use the MCD model by setting a different time or season for a specific mission once the mission date is chosen, and the resulting disturbances might be smaller or larger than the disturbances shown in Fig. 4.4. In this model, we have generated the density disturbances under the average Martian atmospheric conditions, which show similar distributions as the ones generated using the Mars-GRAM 2010 [45].

The aerodynamic coefficient uncertainties used in the simulation are constant during each entry simulation. Since varying errors are considered in the density disturbances,

the total knowledge errors for each trajectory are the combination of constant errors and a varying error. Drag and lift coefficients for one thousand trajectories are generated with Gaussian distributions [45]

$$\begin{aligned} \Delta C_D &\sim N(0,\sigma),\ 3\sigma = 10\% \\ \Delta C_L &\sim N(0,\sigma),\ 3\sigma = 10\% \end{aligned} \tag{6.2}$$

The initial control errors($\mathbf{x}_{0,\text{control}}$) are shown in Table 6.2 with their 3σ values. They are also generated with Gaussian distributions. Since the entry point is defined by the vehicle reaching a certain altitude, the initial control error for altitude is set as zero.

Table 6.2: Entry state delivery errors.

$\Delta\mathbf{x}_0$	Δr_0	$\Delta\theta_0$	$\Delta\phi_0$	ΔV_0	$\Delta\gamma_0$	$\Delta\psi_0$
3σ	0	0.3(deg)	0.03(deg)	3(m/s)	0.15(deg)	0.2(deg)

When navigation errors are considered, three more possible errors are included in the simulations. For initial knowledge errors $\mathbf{x}_{0,\text{knowledge}}$, only initial position and velocity errors are considered [46]

$$\begin{aligned} \Delta r_{0,\text{knowledge}} &\sim N(0,\sigma_r),\ 3\sigma_r = 2\ km \\ \Delta V_{0,\text{knowledge}} &\sim N(0,\sigma_V),\ 3\sigma_V = 1.5\ m/s \end{aligned}$$

Attitude initialization error is propagated through the entire trajectory and would result in inaccurate measurements within each navigation cycle, and thus has a significant effect on the final horizontal errors. The error is set as $\epsilon = 0.25^o(3\sigma)$, the maximum error to meet the MSL requirement [32]. IMU measurement noises for drag and lift are generated by Gaussian distribution and smoothed by a low-pass filter with a time constant of one second

$$\begin{aligned} \delta D &\sim N(0,\sigma_D),\ 3\sigma_D = 3\% \\ \delta L &\sim N(0,\sigma_L),\ 3\sigma_L = 3\% \end{aligned}$$

Statistical Analysis Approach

At the entry interface, all the errors including the initial control error, initial knowledge error and IMU attitude initialization error are generated following the multivariate normal distribution, where Latin hypercube sampling is implemented to generate the overall initial position distribution. During the entry, modeling errors and the IMU measurement errors are integrated into the entry dynamics. At the parachute deploy point, the delivery requirement is considered satisfied if 99% of the cases satisfy it.

Simulation testing of 1000 trajectories is run for each initial flight path angle. During the simulation, guidance frequency is set at 1 Hz [22]. The nominal sampling rate of the measurement data is 200 Hz [27]. Several navigation frequencies of 200 Hz, 100 HZ and 10 Hz are tested in the simulation testing.

6.4 Simulation of Six Degree of Freedom Entry Dynamics

The simulation setup of the 6-DOF entry dynamics is shown in this section. The preliminary simulation results are not shown in Chapter 7. Further assessment is required to generate reasonable results.

When considering the 6-DOF entry dynamics, where the vehicle attitude is included, simulations using Eqs. (2.10)-(2.14) are implemented. In this case, several errors sources from by IMU measurements are considered. The measurements from IMU gyros and accelerometers are generated by adding the sensor errors to the nominal values as

$$\begin{aligned} \tilde{\omega}_{ib} &= (I + S_g + M_g)\omega_{ib} + B_g + W_g \\ \tilde{f}_b &= (I + S_a + M_a) f_b + B_a + W_a \end{aligned} \tag{6.3}$$

$S_g,\ M_g,\ B_g,\ W_g$ are the errors due to the scale factor, misalignment, bias stability and white noise of the gyros. And $S_a,\ M_a,\ B_a,\ W_a$ are the errors due to the scale factor,

misalignment, bias stability and white noise of the accelerometers.

The IMU configuration is modeled as the Honeywell Miniature Inertial Measurement Unit (MIMU) used on the Mars 2005 entry vehicle. The sensor parameters are listed in Table 6.3 [47]. The scale factor errors, misalignment errors, and bias stability errors

Table 6.3: Honeywell MIMU parameters (σ)

	Scale factor	Misalignment	Bias stability	White noise
Accelerometer	$175ppm$	$5ppm$	$0.1mg$	$10\mu g$
Gyro	$5ppm$	5arcsec	$0.01^o/hr$	$0.001^o/\sqrt{hr}$

are modeled as zero-mean, Gaussian-distributed random constants with standard deviations shown in Table 6.3. White noise is modeled as a zero-mean, Gaussian-distributed random walk

$$W_a(t + \delta T) = W_a(t) + \epsilon \tag{6.4}$$

where $\epsilon \sim N(\mathbf{0}, \Sigma)$, and $\Sigma = \mathbf{diag}(10\mu g)$ for the accelerometers and $\Sigma = \mathbf{diag}(0.001^o/\sqrt{hr})$ for the gyros. δT is the navigation cycle period.

Chapter 7

Guidance and Navigation Performance

7.1 Guidance Performance with Perfect Onboard Knowledge

7.1.1 Reference Tracking Guidance (RT) Performance

The RT-mode is first tested under disturbed conditions with MPC as the only controller. One thousand cases are tested using the control laws introduced in Chapter 4.1-4.2. The RT performance is shown in Figs. 7.1 and 7.2. Fig. 7.1 shows the profiles for bank angle, bank angle rate and bank angle acceleration, from which we can see all the profiles satisfy their constraints. Fig. 7.2(a) shows the deployment footprint distribution in the longitude-latitude plane. With only the RT-mode, the longitude and latitude errors can be quite large under disturbed conditions. Fig. 7.2(b) shows the ground tracks of the trajectories. It shows that a number of the trajectories do not converge to the target due to large crossrange errors. This is due to the fact that the RT-mode mainly focuses on longitude. Although bank reversals are executed during the RT-mode to reduce the crossrange error, the reversal times from the pre-designed reference

trajectory may not be appropriate for disturbed conditions. Fig. 7.2(c) presents the final altitude distribution with the parachute deployment box. Although all the constraints are satisfied, some trajectories end with low altitude around 6 km.

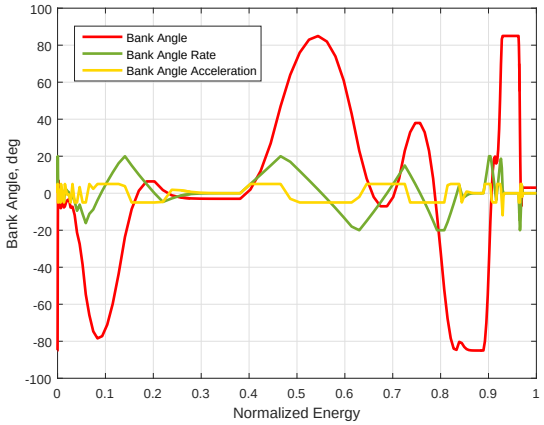
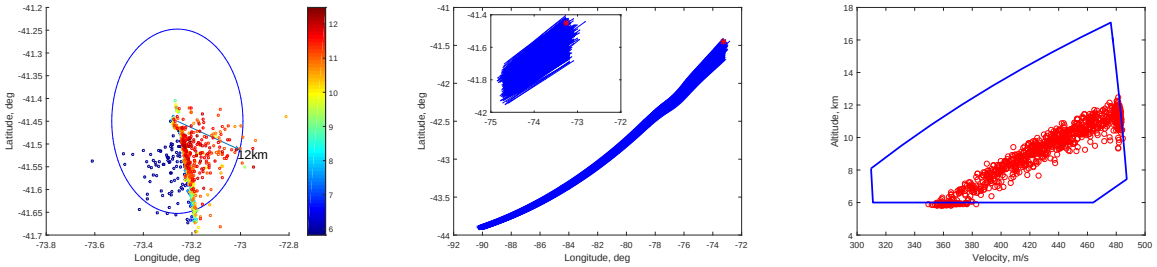


Figure 7.1: Bank angle profile of the RT.



(a) Footprint in longitude-latitude plane.

(b) Ground tracks of trajectories.

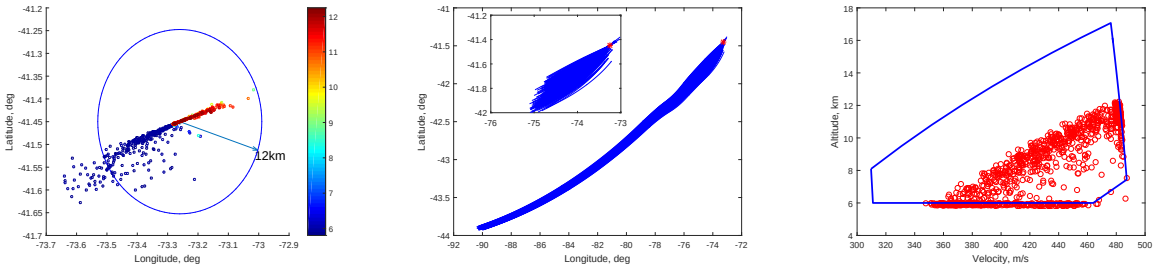
(c) Final altitudes with parachute deployment box.

Figure 7.2: RT performance under disturbances.

7.1.2 Reference Tracking with Deployment Position Alignment Guidance (RTA)

Performance of the guidance with both the RT-mode and the DPA-mode is shown in Fig. 7.3. Fig. 7.3(a) and Fig. 7.3(b) shows that the crossrange errors are greatly

reduced during the final phase of the entry due to lateral control. However, Fig. 7.3(a) still shows significant downrange errors. Furthermore, a number of trajectories end with undershooting downrange errors, which results in trajectories with final altitude around 6 km, where the parachute is forced to deploy. Due to the large disturbances and control saturations, vehicles with low lift-to-drag ratio do not have sufficient control authority to adjust the trajectory to follow a fixed pre-designed reference trajectory. This implies that with only the RT-mode or the combination of the RT-mode with DPA-mode, the required landing accuracy cannot be guaranteed. Hence, the next investigation focuses on the impact of updating the reference trajectory during entry.



(a) Footprint in longitude-latitude plane.

(b) Ground tracks of trajectories.

(c) Final altitudes with parachute deployment box.

Figure 7.3: RTA performance under disturbances.

7.1.3 Reference Tracking, Predictor-Corrector and Deployment Position Alignment Guidance (RUTA)

The complete guidance strategy with the RT-mode, the PC-mode and the DPA-mode is tested in this section. The results of updating one parameter using the golden section method and 3 parameters using the Nelder-Mead method in the PC-mode are compared in Figs. 7.5, 7.4 and Table 7.1. From Figs. 7.5 and 7.4, ground track of trajectories and the footprint show the downrange and crossrange are greatly reduced compared with the RT and RTA guidance algorithms, and most of the trajectories reach a high

elevation at the parachute deployment level around 8 km-12 km.

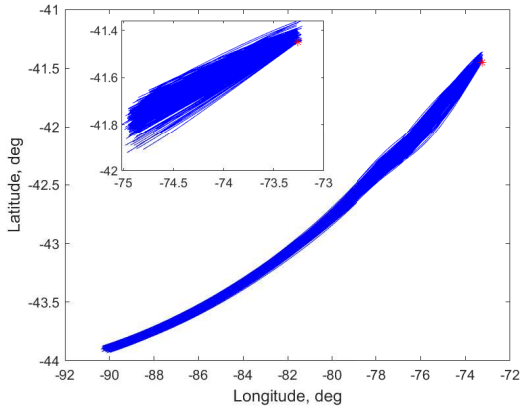
Table 7.1 shows that 99% of the entry trajectories of the RUTA of different updating strategies. By comparing the sub-figures of Figs. 7.5 and 7.4 and the statistical results in Table 7.1, it concludes that updating one parameter is sufficient to achieve the desired performance with high final altitudes and accurate horizontal footprint; 99% of the entry trajectories are guaranteed to have a downrange error within ± 2.0 km, crossrange error within ± 0.3 km and final altitude above 6.9 km. In particular, the crossrange error is rather smaller than that of RUTA with updating 3 parameters. As stated in Section 4.1.2, the Nelder-Mead optimization method requires well-chosen initial values. During each execution of the PC-mode, a poorly chosen set of initial values might not converge to a local minimum. The large crossrange error shown in Table 7.1 has verified this statement in that bank reversal times mainly control the lateral performance. Therefore, one-parameter updating is selected as the strategy used in the PC-mode, and is implemented in the following simulations. The performance of the proposed guidance strategy can be adjusted by choosing different range criterion to trigger the parachute deploy to trade off between deploy elevation and horizontal accuracy, to meet the objectives a particular mission.

Table 7.1: Statistical results of RUTA from the simulation testing

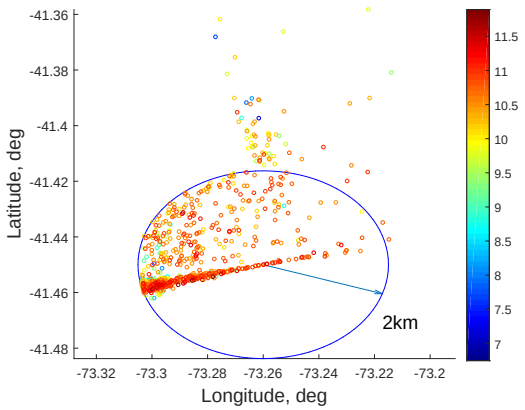
99% Trajectories	Downrange Error(km)	Crossrange Error(km)	Final Altitude(km)
updating 3 parameters	[-1.98, 2.63]	[-3.39, 0.12]	[7.1, 11.3]
updating 1 parameter	[-1.99, 1.76]	[-0.27, 0.16]	[6.9, 12.1]

7.1.4 Comparison of the Different Guidance Strategies

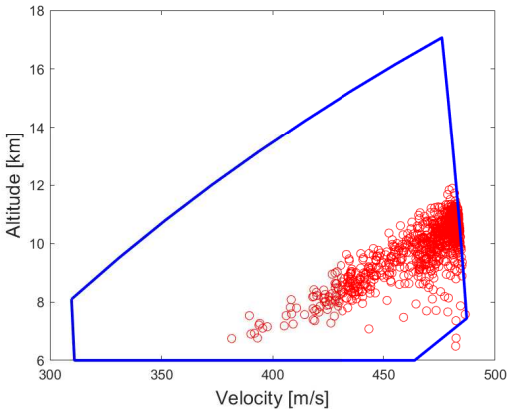
The comparison of the three guidance strategies (RT, RTA and RUTA) in both the horizontal plane and vertical plane are shown in Figs. 7.6 and 7.7. This comparison shows that the complete guidance algorithm with the RT-mode, the PC-mode and the



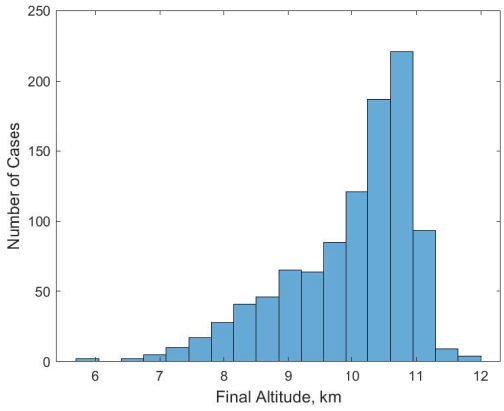
(a) Ground tracks of trajectories.



(b) Footprint in longitude-latitude plane.



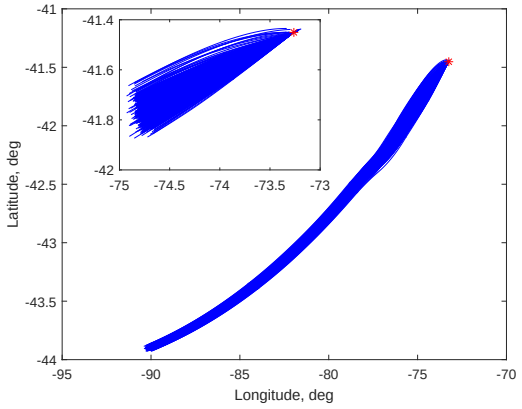
(c) Final altitudes with parachute deployment box.



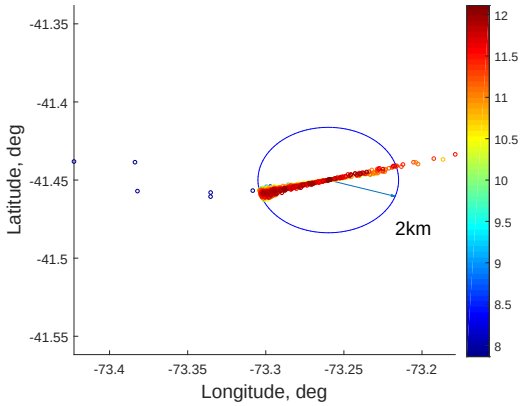
(d) Distribution of final altitude for 1000 trajectories.

Figure 7.4: RUTA performance with optimizing three parameters under disturbances.

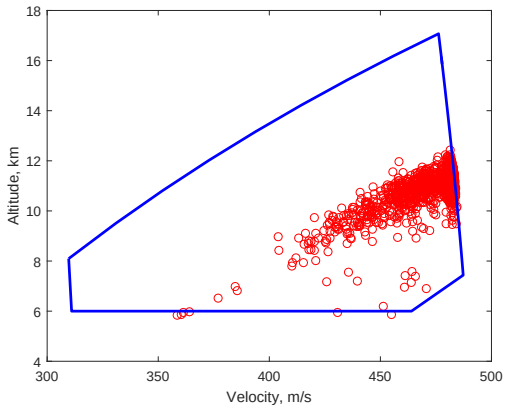
DPA-mode significantly improves the final performance by reducing the radius of the horizontal circle to 2 km and increasing the deploy elevations of 99% trajectories to be greater than 8 km. This performance implies that, under the assumption of perfect knowledge of onboard state variables, the landing site elevations up to 2 km can be achieved by the multi-mode guidance.



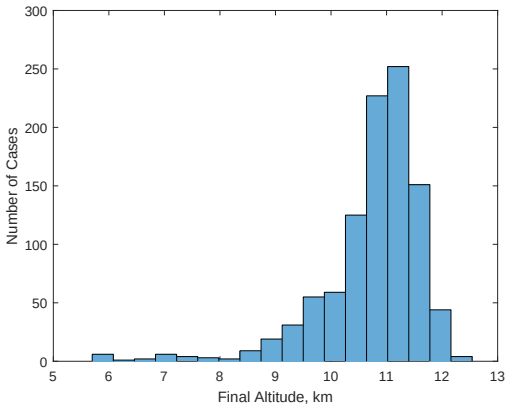
(a) Ground tracks of trajectories.



(b) Footprint in longitude-latitude plane.



(c) Final altitudes with parachute deployment box.



(d) Distribution of final altitude for 1000 trajectories.

Figure 7.5: RUTA performance with optimizing one parameter under disturbances.

7.2 Delivery Performance with Navigation Errors

In this section, navigation errors including initial knowledge errors, attitude initialization errors and measurement noise are considered in the entry guidance assessment. The overall simulation environment includes all the disturbances mentioned in Section 6.3. Two navigation schemes, the conventional inertial navigation and UKF-enhanced navigation are compared regarding the parachute deployment performance of the entry vehicle. Based on the results of the test of different navigation frequencies, the final

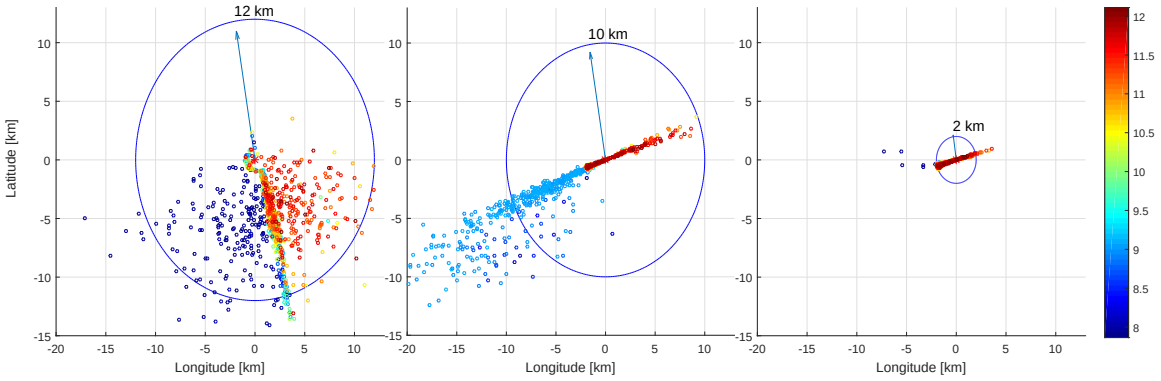


Figure 7.6: Comparison of horizontal errors of RT, RTA and RUTA under disturbances.

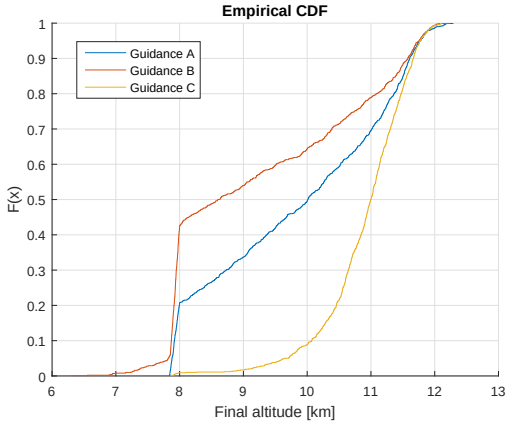
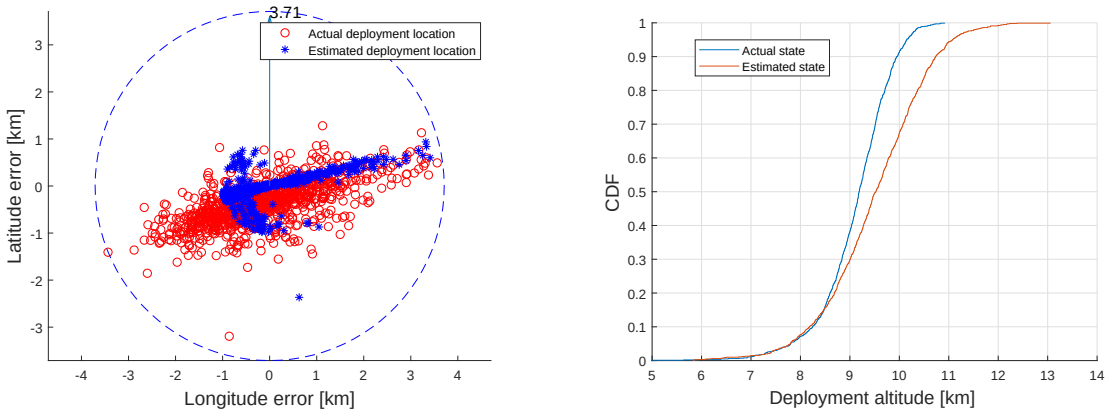


Figure 7.7: Comparison of final altitudes distribution of RT, RTA and RUTA under disturbances.

navigation frequency is set at 10 Hz to save the computational time, given that a larger frequency does not improve the guidance performance.

7.2.1 Guidance with Conventional Inertial Navigation (RUTA-CIN)

The RUTA approach is first assessed with conventional inertial navigation for the four entry flight path angles mentioned in Section 6.1, considering all the disturbances described in Section 6.3.

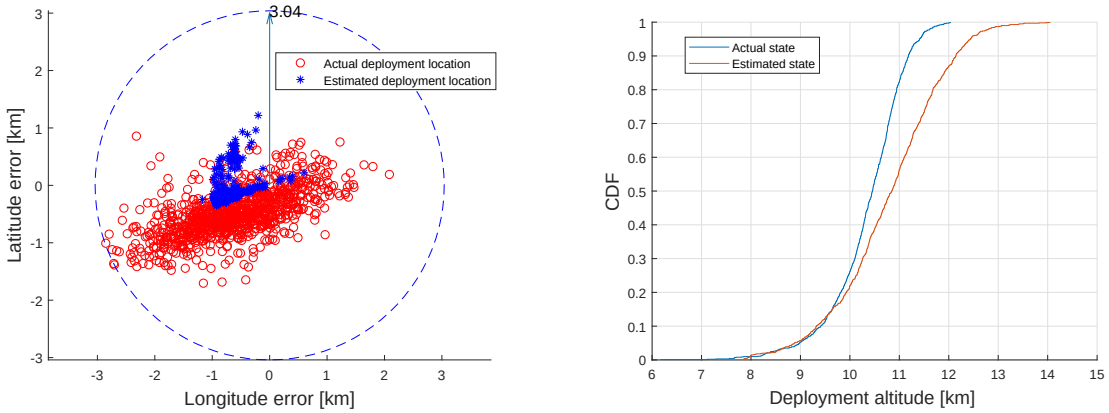


(a) Deployment locations: estimated and actual.

(b) Cumulative density function (CDF) of deployment altitude.

Figure 7.8: RUTA-CIN performance for $\gamma_0 = -15^o$

Figures. 7.8 and 7.9 present that the total delivery errors for steepest entries with $\gamma_0 = -15^o$ and $\gamma_0 = -14^o$ are almost within 1 km. For these steeper entries, the entry trajectories are shorter in duration and there is less time for the navigation errors to grow. Thus the vehicle final performance continues to meet the delivery requirements for the entire range of disturbances considered. The RUTA-CIN strategy is compensating effectively for the disturbances, and the chute deploy trigger effectively terminates most of the estimated states with small horizontal errors. The horizontal errors for the



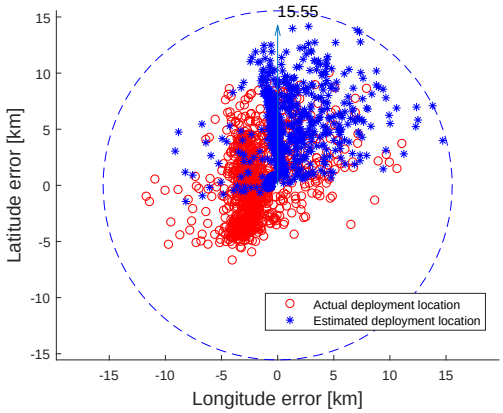
(a) Deployment locations: estimated and actual.

(b) Cumulative density function (CDF) of deployment altitude.

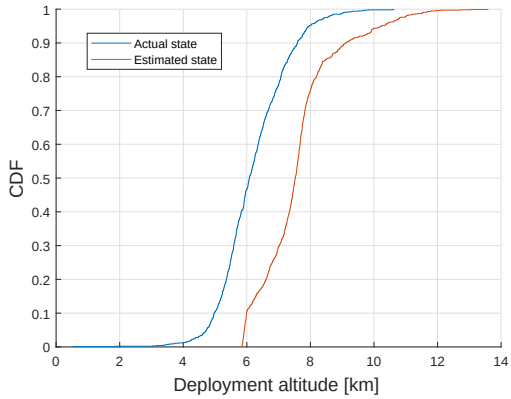
Figure 7.9: RUTA-CIN performance for $\gamma_0 = -14^o$

actual states are larger than those of the estimated states due to the navigation errors. However, the deviations between the actual states and the estimated states are less than 3 km due to the short trajectories of steep entries. Figs. 7.8(a) and 7.9(a) show that the radii of both circles of deployment locations are smaller than 4km. Figs. 7.8(b) and 7.9(b) show the cumulative distributions of the deployment altitudes, from which we can see the final altitudes of all the actual states and the estimated states are above the minimum deploy boundary of 6 km. This deploy altitudes performance is desired for the landing sites with elevations higher than 0 km MOLA, which satisfies the objective of high elevation landing missions. Furthermore, for $\gamma_0 = -15^o$ and $\gamma_0 = -14^o$, the 99% trajectories are terminated at elevations above 7 km and 7.5 km correspondingly, and these two minimum deploy altitudes would guarantee the landing elevations above 1 km and 1.5 km MOLA, which are the desired performance for high elevation landings.

Figs. 7.10 and 7.11 show the RUTA-CIN performance of two shallow entries with $\gamma_0 = -13^o$ and $\gamma_0 = -12^o$. From Figs. 7.10(a) and 7.11(a), we can see that the circles of deployment locations are larger than those for the steeper entries. The horizontal errors can be as large as almost 30 km for the shallowest entry with $\gamma_0 = -12^o$. Furthermore,

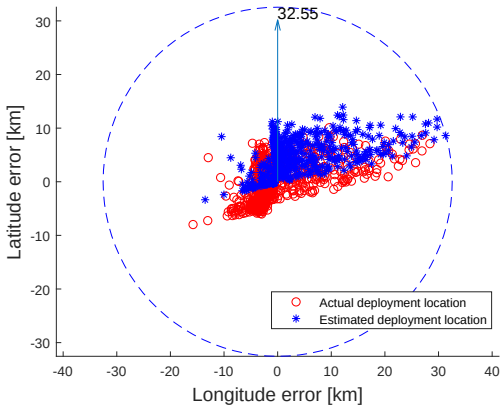


(a) Deployment locations: estimated and actual.

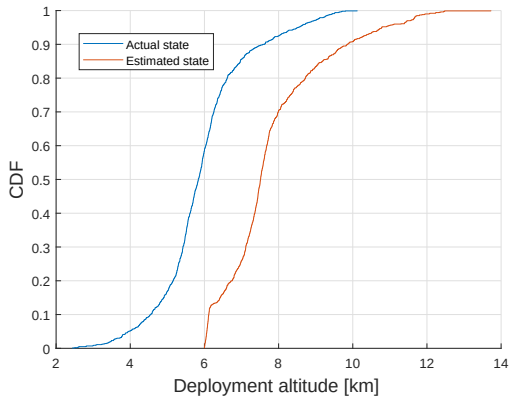


(b) Cumulative density function (CDF) of deployment altitude.

Figure 7.10: RUTA-CIN performance for $\gamma_0 = -13^o$



(a) Deployment locations: estimated and actual.



(b) Cumulative density function (CDF) of deployment altitude.

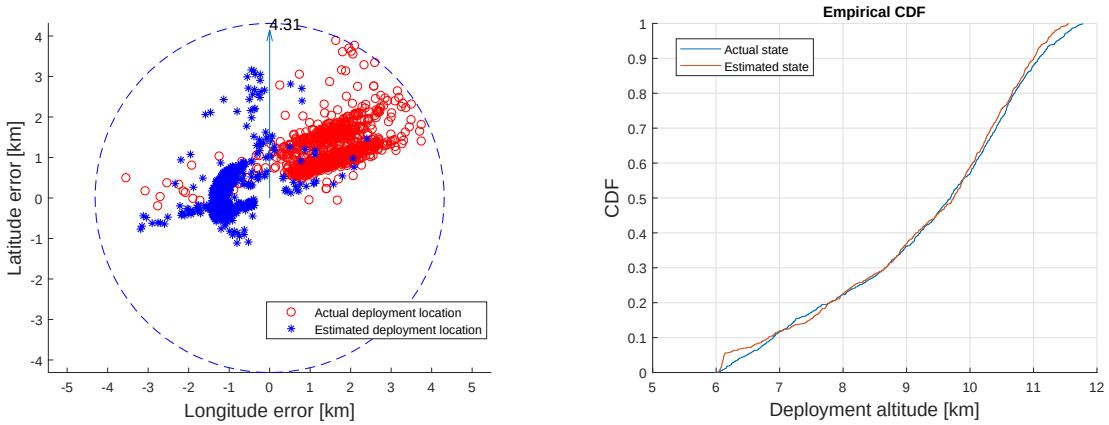
Figure 7.11: RUTA-CIN performance for $\gamma_0 = -12^o$

significant deviations are shown between the actual states and the estimated states. The deviations are particularly large in the altitude. Though the estimated states are terminated by the chute deploy trigger to be above the 6 km, the actual deployment altitude could be as low as 2 km due to the deviations between the actual and the estimated states, as shown in Fig. 7.10. These low deployment altitudes will not leave enough timeline for the subsequent powered descent and landing phases, and lead to

mission failure.

7.2.2 Guidance with UKF-enhanced Navigation (RUTA-UKF)

The RUTA guidance with UKF-enhanced navigation, henceforth referred to as RUTA-UKF, for onboard state estimation during the simulations for shallow entries with $\gamma_0 = -13^o$ and $\gamma_0 = -12^o$. The following figures show the overall performance of these two scenarios.



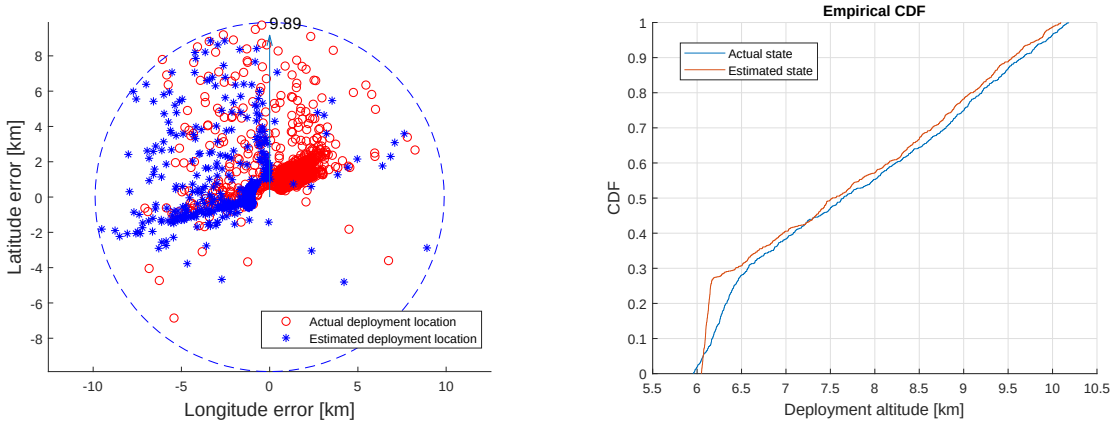
(a) Deployment locations: estimated and actual.

(b) Cumulative density function (CDF) of deployment altitude.

Figure 7.12: RUTA-UKF performance for $\gamma_0 = -13^o$

7.2.3 Comparison of the CIN and UKF-enhanced Navigation

Figs. 7.12 and 7.13 show the RUTA-UKF performance for the two shallow entries with $\gamma_0 = -13^o$ and $\gamma_0 = -12^o$. Compared with Figs. 7.10 and 7.11, the horizontal errors for both cases are significantly reduced. Fig. 7.14 presents the comparisons between the CIN and the UKF-enhanced inertial navigation, showing the mean deployment point and the 99% confidence covariance ellipses for the actual and the estimated states for these two entry scenarios. These two figures indicate the improvement of the UKF-



(a) Deployment locations: estimated and actual.

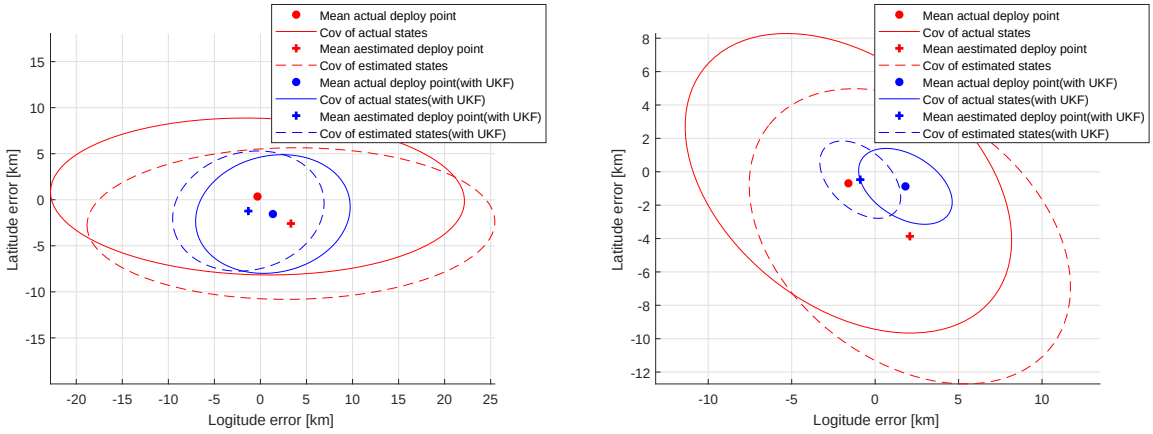
(b) Cumulative density function (CDF) of deployment altitude.

Figure 7.13: RUTA-UKF performance for $\gamma_0 = -12^o$

enhanced navigation in two aspects. One aspect is that the deviations between the actual states and the estimated states have been reduced due to the UKF. The other aspect is that the horizontal errors are reduced, compared with those of the CIN. Since the state estimation has reduced the navigation errors, the total delivery errors are closer to the guidance errors. The improvement of the RUTA-UKF approach is particularly significant in the deployment altitude distributions. From Figs. 7.12(b) and 7.13(b) we can see that the actual states and the estimated states have similar cumulative distributions. For $\gamma_0 = -13^o$, all the trajectories are terminated by the parachute deployment above 6 km. For $\gamma_0 = -12^o$, the trajectories are terminated at altitude above 6 km for 99% tested cases.

The simulation results of the multi-mode guidance and navigation system for the 4 entry flight path angles are summarized in Table 7.2. This table concludes that using RUTA-CIN, landing elevation up to 1.5 km MOLA is achievable for the MSL nominal entry at $\gamma_0 = -14^o$; for $\gamma_0 = -15^o$, 1.0 km elevation is achievable. Using RUTA-UKF, landing elevation up to 0 km MOLA is achievable for the shallower entry with. $\gamma_0 = -13^o$ and $\gamma_0 = -12^o$. With RUTA-CIN, horizontal accuracy within 10 km can be

met for $\gamma_0 = -15^o$ and $\gamma_0 = -14^o$, but not for the shallower entries, while RUTA-UKF meets the horizontal accuracy within 10 km for $\gamma_0 = -13^o$ and $\gamma_0 = -12^o$. For 1-2 km elevation landing, as required to access more of the southern highlands, further improvement in the RUTA-UKF algorithms, or the addition of navigation sensors, would be required.



(a) Covariance ellipses of RUTA-CIN and RUTA-UKF for $\gamma_0 = -12^o$

(b) Covariance ellipses of RUTA-CIN and RUTA-UKF for $\gamma_0 = -13^o$

Figure 7.14: Performance comparison of RUTA-CIN and RUTA-UKF for $\gamma_0 = -12^o$ and $\gamma_0 = -13^o$

The shallowest entry with $\gamma_0 = -12^o$ results in a deployment circle with radius around 10 km. This performance, though satisfies the horizontal accuracy requirement mentioned in Chapter 3, may not be accurate enough for future missions with higher requirements. For the future Mars landing missions such as crewed missions, a smaller parachute deployment error ellipse would be required. Some missions also require pin-point landings with landing accuracy of sub-kilometers [50]. Furthermore, the guidance proposed in this dissertation is designed to achieve a landing elevation above 0 km. Future missions may require landing elevations above 1 km or 2 km. These requirements, though not addressed in this dissertation, are the objectives for the future research.

Table 7.2: Performance comparison of RUTA-CIN and RUTA-UKF

	Max Horizontal Delivery Error [km]	Max Deploy Altitude [km]	Max Site Elevation [km]
$\gamma_0 = -15^o$			
RUTA-CIN	3.7	7.0	1.0
$\gamma_0 = -14^o$			
RUTA-CIN	3.0	7.5	1.5
$\gamma_0 = -13^o$			
RUTA-CIN	15.6	3.3	-3.3
RUTA-UKF	4.3	6.1	0.1
$\gamma_0 = -12^o$			
RUTA-CIN	32.6	2.6	-4.6
RUTA-UKF	9.9	6.0	0.0

In general, two approaches can be explored to further improve the current guidance performance. First, different configurations of entry vehicles from MSL could be designed which allow for higher limits of peak g-load and heat rate, such that a steep entry is a feasible entry condition. In this case, the proposed guidance algorithm requires further verification in order to generate accurate landing performance for each new configuration. Second, if a shallow entry is indeed required due to trajectory constraints, the guidance and navigation strategy would need to be further improved to achieve better performance. Some recent research work has been focusing on adding extra speed reduction capabilities to the entry vehicle such as the Supersonic Inflatable Aerodynamics Decelerator (SIAD) [48] or the Hypersonic Inflatable Aerodynamics Decelerator (HIAD) [49]. With additional means of reducing the speed of the entry

vehicle, the guidance algorithm would have more control over the horizontal accuracy. There would be an additional entry phase requiring an entry guidance component to be developed and integrated into the overall strategy.

Chapter 8

Conclusions

A guidance and navigation approach has been developed and tested to achieve accurate landing at higher elevation sites on Mars. The proposed guidance algorithm which combines the RT-mode, the BR-mode, the PC-mode and DPA-mode is able to deliver the entry vehicle to the maximum deploy altitude of 2 km with horizontal accuracy within 2 km, under the assumption of perfect knowledge of onboard state variables. When navigation errors are considered, in order to achieve the high deploy elevation above 6 km and the total delivery error less than 10 km, the guidance approach was further combined with a navigation scheme. The UKF was selected as the state estimator based on the observability analysis of the entry dynamics. The combined guidance and navigation algorithm has been assessed by simulation testing that accounted for all the significant sources of error. A range of entry flight path angles was selected to include both shallow and steep entries, where shallow entry is of interest for lowering the maximum acceleration and peak heat rate for which the entry vehicle has to be designed.

Results from the simulation testings have shown that for steep entries, the guidance algorithm with the conventional inertial navigation is capable of generating trajectories with the landing elevations up to 1 km for 99% cases and horizontal errors within 4 km. For shallow entries with longer trajectories, the UKF-enhanced navigation is

necessary to obtain estimations of onboard state variables and significantly improves the final performance by achieving the landing elevations up to 0.1 km and horizontal errors within 10 km. In conclusion, the combined multi-mode guidance and navigation approach has achieved the two objectives of high elevation landing up to 0 km MOLA and deploy circle radius smaller than 10 km for an MSL-type of entry vehicle.

Bibliography

[1] Kluever, C. A., "Entry Guidance Performance for Mars Precision Landing", *Journal of Guidance, Control, and Dynamics* November–December 2008, Vol. 31, No. 6, pp. 1537-1544.

[2] Mendeck, G. F. and Craig, L. M., "Entry Guidance Design and Postflight Performance for 2011 Mars Science Laboratory Mission", *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, July-Aug. 2014, pp.

[3] Curiosity's Landing Site: Gale Crater, `"http://mars.nasa.gov/msl/mission/timeline/prelaunch/landingsiteselection/aboutgalecrater/"`.

[4] Braun, R. D. and Manning, R. M., "Mars exploration entry, descent, and landing challenges". *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, March?April 2007, pp.

[5] Li S. and Jiang, X., "Review and prospect of guidance and control for Mars atmospheric entry?, *Progress in Aerospace Sciences*, 2014.

[6] Soler, L., Khatib, A., and Mease, K., "Mars Entry Trajectory Planning for Higher Elevation Landing", *23rd AAS/AIAA Space Flight Mechanics Meeting*, 2013.

[7] Benito, J., Soler, L., Bombelli. A. and Mease K., "Entry Trajectory Planner for Higher Elevation Mars Landing", *unpublished manuscript*

[8] Benito, J. and Mease, K. D., "Entry Trajectory Planning for Higher Elevation Landing", *AAS Astrodynamics Specialist Conference*, Mackinac Island, MI, Aug. 2007, pp. AAS 07-309.

[9] Carman, G. L., "Apollo-Derived Mars Precision Lander Guidance", 1998

[10] Mendeck, G. F., "Guidance Design for Mars Smart Landers Using The Entry Terminal Point Controller", *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 5-8 August 2002, Monterey, CA.

[11] Ping Lu, "Entry Guidance: A Unified Method", *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, May-June 2014

[12] Xue, S., and Lu, P., "Constrained Predictor-Corrector Entry Guidance", *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1273-1280.

[13] Braun, R. D., and Powell, R. W., "Predictor-Corrector Guidance Algorithm for Use in High-Energy Aerobraking System Studies", *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 672?678.

[14] Craig A. Kluever, "Approach and Landing Range Guidance for an Unpowered Reusable Launch Vehicle", *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, November 2015.

[15] Lu, P., "Entry Guidance and Trajectory Control for Reusable Launch Vehicle", *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 143-149.

[16] Benito, J. and Mease, K. D., "Nonlinear Predictive Controller for Drag Tracking in Entry Guidance", *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, Aug. 2008, pp. AIAA 2008-7350.

[17] Benito, J. and Mease, K. D., "Mars Entry Guidance with Improved Altitude Control", *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, Colorado, Aug. 2006, pp. AIAA 2006-6674.

[18] Mendeck, G. and Craig, L., "Entry Guidance for the 2011 Mars Science Laboratory Mission", *AIAA Atmospheric Flight Mechanics Conference*, 08 - 11 August 2011, Portland, Oregon, pp. AIAA 2011-6639.

[19] Duan, G., Rovira Navarro, M. and Mease, K. D., "Trajectory Tracking and Online Replanning for Mars Entry", *AIAA/AAS Astrodynamics Specialist Conference*, Long Beach, CA, Sept. 13, 2016, pp. AIAA 2016-5444.

[20] Duan, G. and Mease, K. D., "Mars Entry Guidance for High Elevation Landing with Uncertainty Quantification", *22nd AIAA International Space Planes and Hypersonics Systems and Technologies Conference*, 17-19 Sept. 2018, Orlando, FL, pp. 2018-5315.

[21] Yu, Z., Cui, P. and Crassidis, J. L., "Design and optimization of navigation and guidance techniques for Mars pinpoint landing: Review and prospect", *Progress in Aerospace Sciences*, vol. 1, No. 13, 2017, pp. 0376-0421.

[22] Li, S. and Peng, Y., "Radio beacons/IMU integrated navigation for Mars entry", *Advances in Space Research*, vol. 47, 2011, pp. 1265–1279.

[23] Yu, Z., Cui, P. and Zhu, S., "Observability-based beacon configuration optimization for Mars entry navigation", *Journal of Guidance, Control, and Dynamics*, vol. 38, No. 4, 2015, pp. 643–650.

[24] Jiang, X., Li, S. and Huang, X., "Radio/FADS/IMU integrated navigation for Mars entry", *Advances in Space Research*, vol. 61, No. 5, 2018, pp. 1342-1358.

[25] M.C. Heyne, "Spacecraft Precision Entry Navigation Using an Adaptive Sigma Point Kalman Filter Bank", The University of Texas at Austin, Austin TX, 2007.

[26] Dutta S., Braun R. D.,and Karlgaard C., "Uncertainty Quantification for Mars Entry, Descent, and Landing Reconstruction Using Adaptive Filtering", *Journal of Spacecraft and Rockets* Vol. 51, No. 3, May–June 2014

[27] Dutta S. and Braun R. D., "Statistical Entry, Descent, and Landing Performance Reconstruction of the Mars Science Laboratory", *Journal of Spacecraft and Rockets* Vol. 51, No. 4, July–August 2014

[28] Lévesque, J. F. and Lafontaine, J., "Innovative Navigation Schemes for State and Parameter Estimation During Mars Entry", *Journal of Guidance, Control, and Dynamics* Vol. 30, No. 1, January–February 2007

[29] Benito, J. and Mease, K. D., "Reachable and Controllable Sets for Planetary Entry and Landing", *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 641-654.

[30] Benito, J., "Advance in Spacecraft Atmospheric Entry Guidance", Ph.D. Dissertation, Mechanical and Aerospace Engineering Dept., University of California Irvine, Irvine, CA, 2010.

[31] M.A. Patterson and A.V. Rao. Gpops-ii: "A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming", ACM Transactions on Mathematical Software (TOMS), 41(1):1, 2014.

[32] Wolf, A. A., *et al.* "Improving the Landing Precision of an MSL-Class Vehicle", *2012 IEEE Aerospace Conference*, Big Sky, MT, 2012, pp. 1-10.

[33] Hunt, L., "Sufficient Conditions for Controllability", *IEEE Transactions on Circuits and Systems*, Vol. CAS-29, NO. 5, May 1982.

[34] Yu, Z., Cui, P. and Zhu, S., "Observability-Based Beacon Configuration Optimization for Mars Entry Navigation", *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, April 2015, pp. 643-650.

[35] Mendeck, G. F. and Craig, L. E., "Spacecraft Entry Navigation using Sigma Point Kalman Filtering", *2006 IEEE/ION Position, Location, And Navigation Symposium*, Coronado, CA, April 2006, pp. 2153-3598.

[36] A. Gelman., et al, "Bayesian Data Analysis", Chapman and Hall, 1995, ISBN 0-412-03991-5.

[37] Ristic, B. and Salmond, D. "A study of a nonlinear filtering problem for tracking an extended target", *Seventh Intl. Conf. on Information Fusion*, June 2004, pp. 503-509.

[38] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. "Estimation with Applications to Tracking and Navigation", John Wiley & Sons, 2001

[39] J. C. Spall, "Adaptive stochastic approximation by the simultaneous perturbation method", *IEEE transactions on automatic control*, vol. 45, pp. 1839-1853, 2000.

[40] Stoica, P. and Moses, R., *Spectral Analysis of Signals*, New Jersey, Pearson Prentice Hall, 2005, Print.

[41] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, VOL. 50, NO. 2, Feb. 2002.

[42] Braun, R. and Manning, R., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft And Rockets*, Vol. 44, No. 2, April 2007, pp. 310-323.

[43] Subrahmanyam, P. and Rasky, D., "Entry, Descent, and Landing technological barriers and crewed Mars vehicle performance analysis," *Progress in Aerospace Sciences*, Vol. 91, May. 2017, pp. 1-26.

[44] Rao, A. V., *et al.* "Algorithm 902: GPOPS, a Matlab Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 39, No. 3, 2010, pp. 1–38.

[45] Mandalia, A. and Braun, R., "High-Altitude Divert Architecture for Future Robotic and Human Mars Missions", *Journal of Spacecraft and Rockets*, Vol. 52, No. 5, Sep. 2015, pp. 1311-1319.

[46] Striepe, S., *et al.* "Mars Science Laboratory Simulations for Entry, Descent, and Landing", *Journal of Spacecraft And Rockets*, Vol. 43, No. 2, April 2006, pp. 311-323.

[47] Timothy, P. C., and Robert, H. B., "Mars Entry Navigation: Atmospheric Interface Through Parachute Deploy," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2002.

[48] Miller, M., Steinfeldt B. A. and Braun, R., "Investigation of Drag-Modulated Supersonic Inflatable Aerodynamic Decelerators for Sounding Rocket Payloads", *Journal of Spacecraft And Rockets*, March–April 2015, Vol. 52, No. 2.

[49] Li, L., *et al.* "Design of a Novel Hypersonic Inflatable Aerodynamic Decelerator for Mars Entry, Descent, and Landing",

[50] M.B. Quadrelli, L.J. Wood, J.E. Riedel, et al., Guidance, navigation, and control technology assessment for future planetary science missions, *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, July 2015, pp.2015-116.